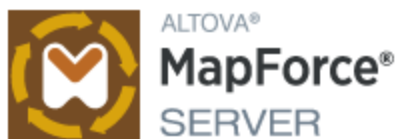


Altova MapForce Server 2024 Advanced Edition



Manuel de l'utilisateur et de référence

Altova MapForce Server 2024 Advanced 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: 2024

© 2018-2024 Altova GmbH

Table des matières

1	Introduction	6
2	Installation et licence	8
2.1	Configuration sur Windows.....	9
2.1.1	Installation sur Windows.....	9
2.1.2	Installer sur Windows Server Core.....	10
2.1.3	Installer LicenseServer (Windows).....	12
2.1.4	Licence MapForce Server (Windows).....	13
2.2	Configuration sur Linux.....	17
2.2.1	Installer sur Linux.....	17
2.2.2	Installer LicenseServer (Linux).....	19
2.2.3	Licence MapForce Server (Linux).....	20
2.3	Configuration sur macOS.....	23
2.3.1	Installer sur macOS.....	23
2.3.2	Installer LicenseServer (macOS).....	24
2.3.3	Licence MapForce Server (macOS).....	25
2.4	Mise à jour MapForce Server.....	28
2.5	Migrer MapForce Server vers un nouvel appareil.....	29
3	Installation	30
3.1	Exigences du système.....	30
3.2	Cœurs de processeur et licences.....	31
3.3	Chemins importants.....	32
3.4	Linux	33
3.4.1	Installer MapForce Server.....	33
3.4.2	Consulter les produits installés actuellement.....	34
3.4.3	Désinstaller MapForce Server.....	34
3.5	macOS.....	35

3.5.1	Installer MapForce Server.....	35
3.5.2	Désinstaller MapForce Server.....	36
3.6	Windows.....	37
3.6.1	Installer MapForce Server.....	37
3.6.2	Désinstaller MapForce Server.....	39
4	Exécuter les mappages	40
4.1	Préparer les mappages pour l'exécution de serveur.....	43
4.2	Ressources Globales.....	49
4.3	Optimisation Join.....	51
4.4	Identifiants.....	54
4.4.1	Exemple: OAuth 2.0 Authorization.....	56
4.5	Authentification dynamique.....	71
5	Ligne de commande MapForce Server	72
5.1	assignlicense.....	75
5.2	exportresourcestrings.....	76
5.3	help	78
5.4	licenseserver.....	79
5.5	run	80
5.6	setdeflang.....	84
5.7	verifylicense.....	85
5.8	version.....	86
6	MapForce Server API	87
6.1	Interface .NET.....	88
6.1.1	Exemple C#.....	89
6.1.2	Exemple Visual Basic .NET	92
6.2	Interface COM.....	96
6.2.1	Exemple C++.....	96
6.2.2	Exemple VBScript.....	99
6.2.3	Exemple VBA.....	102
6.3	Interface Java.....	105

6.3.1	Exemple Java.....	109
6.4	Exemple : Exécuter un mappage avec des paramètres.....	112
6.5	API Reference (COM, .NET).....	118
6.5.1	Interfaces.....	118
6.6	API Reference (Java).....	136
6.6.1	Classes.....	136
7	Gestion du certificat numérique	148
7.1	Certificats de serveur de confiance sur Linux.....	151
7.2	Certificats de serveur de confiance sur macOS.....	152
7.3	Certificats de serveur de confiance sur Windows.....	153
7.4	Accéder aux boutiques de certificat sur Windows.....	154
7.5	Exporter des certificats depuis Windows.....	155
7.6	Certificats client sur Linux.....	161
7.7	Certificats client sur macOS.....	163
7.8	Certificats client sur Windows.....	165
7.9	Gestionnaire de taxonomie.....	167
7.9.1	Exécuter le Gestionnaire de taxonomie.....	171
7.9.2	Catégories de statut.....	173
7.9.3	Retoucher ou Installer une taxonomie.....	174
7.9.4	Désinstaller une taxonomie, Réinitialiser.....	175
7.9.5	Interface de ligne de commande (CLI).....	176
8	Fichiers catalogues	184
	Index	185

1 Introduction

MapForce Server est une solution d'entreprise qui exécute des transformations de mappage de données sur les systèmes d'exploitation Windows, Linux, et macOS. Les mappages de données eux (ou Mapping Design Files, *.mfd) sont conçus visuellement avec Altova MapForce (<http://www.altova.com/fr/mapforce.html>). Vous y définissez les entrées, sorties, et les étapes intermédiaires de traitement qui seront appliquées à vos données. Le rôle de MapForce Server est d'exécuter les fichiers MapForce Server Execution (.mfx) compilés avec MapForce, et de produire les fichiers ou les données de sortie ou bien même de mettre à jour les bases de données ou d'appeler les services Web, conformément au design du mappage sous-jacent.



MapForce Server peut fonctionner seul ou bien sous la gestion d'Altova FlowForce Server (<http://www.altova.com/fr/flowforce.html>). S'il est installé sur le même appareil que MapForce Server, FlowForce Server automatise l'exécution des mappages par le biais de tâches programmées ou basées sur des déclencheurs, qui peuvent être exposés en tant que services web. De plus, FlowForce Server contient une bibliothèque intégrée de fonctions qui permettent de prendre des actions automatisées supplémentaires avant ou après l'exécution du mappage, comme par exemple envoyer des e-mails, copier des fichiers et des répertoires, charger des fichiers sur FTP, exécuter des commandes shell, etc.

Fonctions

- Performance au niveau serveur lors de l'exécution de mappages de données
- Multiplateforme : MapForce Server marche sur les systèmes d'exploitation Windows, Linux ou macOS
- Interface de ligne de commande
- Une API que vous pouvez appeler depuis le code C++, C#, Java, VB.NET, VBScript, ou VBA
- Intégration native avec FlowForce Server
- Prise en charge pour les Ressources globales Altova. Un moyen de rendre des fichiers, des dossiers ou des références de base de données configurables et portables sur des environnements multiples et sur plusieurs applications Altova, voir [Ressources Globales Altova](#)⁴⁹
- Accélère les mappages lorsque l'optimisation Join est possible (voir [À propos de l'Optimisation Join](#)⁵¹)
- Exécute des mappages qui appliquent des fonctions et des défauts à plusieurs items simultanément. Ces mappages permettent, par exemple de remplacer aisément toutes les valeurs nulles rencontrées avec des strings vides ou du texte personnalisé
- Exécute des mappages qui lisent des données depuis et écrire des données sur le format binaire Protocol Buffers
- Exécute des mappages qui effectuent des inserts de base de données bulk

Limitations

- 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. Sur Linux et macOS, la connectivité de base de données native est disponible pour les bases de données SQLite et PostgreSQL. Pour d'autres bases de données exécutées sur Linux ou macOS, JDBC devrait être utilisé.

Dernière mise à jour : 08.04.2024

2 Installation et licence

Cette section décrit l'installation, la gestion de licence et d'autres procédures de configuration. Elle est organisée en sections comme suit :

- [Configuration sur Windows](#) ⁹
- [Configuration sur Linux](#) ¹⁷
- [Configuration sur macOS](#) ²³
- [Mise à niveau de MapForce Server](#) ²⁸
- [Migrer MapForce Server vers un nouvel appareil](#) ²⁹

2.1 Configuration sur Windows

Cette section décrit l'[installation](#)⁹ et la [licence](#)¹³ de MapForce Server sur les systèmes Windows.

Exigences de système (Windows)

Notez les exigences système suivantes :

- Windows 10, Windows 11
- Windows Server 2016 ou plus récent

Prérequis

Notez les exigences préalables suivantes :

- Réaliser une installation en tant qu'utilisateur de privilèges administratifs.
- À partir de la version 2021, une version 32-bit de MapForce Server ne peut pas être installée sur la version 64-bit, ou une version 64-bit sur une version 32-bit. Vous devez soit (i) supprimer la version plus ancienne avant d'installer la nouvelle version ou (ii) mettre à niveau vers la version plus nouvelle qui est la même version bit que votre ancienne installation.

2.1.1 Installation sur Windows

MapForce Server est disponible pour l'installation sur des systèmes Windows. La procédure élargie d'installation et de configuration est décrite ci-dessous. Pour des informations détaillées sur des parties spécifiques de la procédure d'installation, voir les sections respectives.

Installation de MapForce Server

MapForce Server peut être installé sur les systèmes Windows comme suit :

- En tant que produit de serveur autonome séparé. Pour installer MapForce Server, téléchargez et exécutez le programme d'installation MapForce Server. Suivez les instructions sur écran.
- Pour installer MapForce Server comme partie du package [FlowForce Server](#), téléchargez et exécutez le programme d'installation FlowForce Server. Suivez les installations sur écran veillez à cocher l'option pour installer MapForce Server.

Les programmes d'installation pour MapForce Server et [FlowForce Server](#) sont disponibles au Altova Download Center (<http://www.altova.com/download.html>). Vous pouvez sélectionner votre langue d'installation depuis la zone inférieure gauche de l'assistant. Notez que cette sélection définit également la langue par défaut de MapForce Server. Vous pouvez changer la langue plus tard depuis la ligne de commande.

Installer LicenseServer

Pour que MapForce Server fonctionne, il faut qu'il ait une enregistré licence par le biais d'un [Altova LicenseServer](#) sur votre réseau. Lorsque vous installez MapForce Server ou FlowForce Server sur les systèmes Windows, vous pouvez installer LicenseServer ensemble avec MapForce Server ou FlowForce Server. Pour les détails, voir [Installer LicenseServer](#)¹².

Après l'installation, l'exécutible MapForce Server sera situé par défaut sous le chemin suivant :

```
<ProgramFilesFolder>\Altova\MapForceServer2024\bin\MapForceServer.exe
```

Tous les enregistrements nécessaires pour utiliser MapForce Server via une interface COM, comme une interface Java, et dans l'environnement .NET seront effectués par le programme d'installation.

Installer sur Windows Server Core

Windows Server Core n'a pas de GUI et doit être installé via la ligne de commande. Voir la section [Installer sur Windows Server Core](#) ¹⁰ pour l'installation.

Désinstaller MapForce Server

Désinstaller MapForce Server comme suit :

1. Cliquez de la touche droite sur le bouton Windows **Démarrer** et sélectionnez **Paramètres**.
2. Ouvrez le panneau de configuration (commencez à écrire « Panneau de configuration » et cliquez sur l'entrée suggérée).
3. Sous *Programmes*, cliquez **Désinstaller un programme**.
4. Dans le panneau de configuration, sélectionnez MapForce Server et cliquez sur **Désinstaller**.

Licence d'évaluation

Pendant le processus d'installation, vous recevrez l'option de demander une licence d'évaluation de 30 jours pour MapForce Server. Après avoir soumis la demande, une licence d'évaluation sera envoyée à l'adresse e-mail que vous avez enregistré.

2.1.2 Installer sur Windows Server Core

Windows Server Core est une installation Windows minimale qui n'utilise qu'un certain nombre de fonctions de la GUI. Vous pouvez installer MapForce Server sur un appareil Windows Server Core comme suit :

1. Téléchargez le programme d'installation MapForce Server exécutable depuis le site web d'Altova. Ce fichier est nommé **MapForceServerAdv.exe**. Assurez-vous que l'exécutable corresponde à votre plateforme de serveur (32-bit ou 64-bit).
2. Sur un appareil standard Windows (pas l'appareil Windows Server Core), exécutez la commande **MapForceServerAdv.exe /u**. Ceci déballe le fichier **.msi** vers le même dossier que celui du programme d'installation.
3. Copiez le fichier décompressé **.msi** vers l'appareil Windows Server Core.
4. Si vous mettez à jour une version antérieure de MapForce Server, fermez MapForce Server avant de réaliser la prochaine étape.
5. Utilisez le fichier **.msi** pour l'installation en exécutant la commande **msiexec /i MapForceServerAvancé.msi**. Ceci lance l'installation de Windows Server Core.

Important : Garder le fichier MSI !

Veuillez noter les points suivants :

- Gardez le fichier 1' `extraction.msi` à un endroit sûr. Vous en aurez besoin plus tard pour désinstaller, réparer ou modifier votre installation.
- Si vous voulez renommer le fichier MSI, faites-le avant d'installer MapForce Server.
- Le nom du fichier MSI est stocké dans le registre. Vous pouvez mettre à jour son nom ici si le nom du fichier a changé.

Inscrire MapForce Server avec LicenseServer

Si vous installez MapForce Server pour la première fois ou si vous mettez à jour vers une **version majeure**, vous allez devoir enregistrer MapForce Server avec Altova LicenseServer sur votre réseau. Si vous êtes en train de mettre à jour vers une version non majeure de MapForce Server, alors l'enregistrement précédent de LicenseServer sera connu par l'installation et vous n'aurez pas besoin d'enregistrer MapForce Server avec LicenseServer. Toutefois, si vous voulez changer le LicenseServer qui est utilisé par MapForce Server à tout moment, vous allez devoir enregistrer MapForce Server avec le nouveau LicenseServer.

Pour enregistrer MapForce Server avec Altova LicenseServer pendant l'installation, exécutez la commande d'installation avec la propriété `REGISTER_WITH_LICENSE_SERVER`, telle que recensée ci-dessous, fournissant le nom ou l'adresse de la machine de LicenseServer en tant que valeur de la propriété, par exemple :

```
msiexec /i MapForceServerAdvanced.msi REGISTER_WITH_LICENSE_SERVER="localhost"
```

Pour enregistrer MapForce Server avec un Altova LicenseServer après l'installation, exécutez la commande suivante :

```
msiexec /r MapForceServerAdvanced.msi REGISTER_WITH_LICENSE_SERVER="<MyLicenseServer-Machine-Address>"
```

Commandes utiles

Vous trouverez ci-dessous un ensemble de commandes utiles dans le contexte de l'installation.

Pour tester la valeur retour de votre installation, exécutez un script semblable à celui ci-dessous. Le code retour sera dans la variable d'environnement `%errorlevel%` . Un code retour 0 indique un succès.

```
start /wait msiexec /i MapForceServerAdvanced.msi /q
echo %errorlevel%
```

Pour une installation silencieuse avec un code retour et un log de la procédure d'installation :

```
start /wait msiexec /i MapForceServerAdvanced.msi /q /L*v! <pathToInstallLogFile>
```

Pour modifier l'installation :

```
msiexec /m MapForceServerAdvanced.msi
```

Pour réparer l'installation :

```
msiexec /r MapForceServerAdvanced.msi
```

Pour désinstaller MapForce Server :

```
msiexec /x MapForceServerAdvanced.msi
```

Pour la désinstallation de MapForce Server en silence et faire rapport du résultat détaillé dans un fichier log :

```
start /wait msiexec /x MapForceServerAdvanced.msi /q /L*v! <pathToUninstallLogFile>
```

Pour installer MapForce Server en utilisant un autre langage (les codes de langage disponibles sont : allemand=`de`; espagnol=`es`; français=`fr`) :

```
msiexec /i MapForceServerAdvanced.msi INSTALLER_LANGUAGE=<languageCode>
```

Note : sur Windows Server Core, la fonctionnalité des graphiques de MapForce Server ne sera pas disponible.

Note : pour installer les taxonomies, utilisez le Taxonomy Package Manager par le biais de la ligne de commande. Consultez le manuel MapForce Server pour des informations sur comment faire.

2.1.3 Installer LicenseServer (Windows)

Pour que MapForce Server fonctionne, il faut qu'il ait une licence par le biais d'un [Altova LicenseServer](#) sur votre réseau. Lorsque vous installez MapForce Server ou FlowForce Server sur les systèmes Windows, vous pouvez installer LicenseServer ensemble avec MapForce Server ou FlowForce Server. Si un LicenseServer est déjà installé sur votre réseau, vous ne devez pas installer un autre — sauf si une plus nouvelle version de LicenseServer est requise. (*Voir le point suivant, [versions de LicenseServer](#).*)

Pendant la procédure d'installation de MapForce Server ou de FlowForce Server, activez ou désactivez l'option pour installer LicenseServer, si besoin. Veuillez noter les points suivants :

- Si vous n'avez pas encore installé LicenseServer, laissez les paramètres par défaut tels quels. L'assistant installera la dernière version sur l'ordinateur sur lequel vous exécutez l'assistant.
- Si vous n'avez pas encore installé LicenseServer et souhaitez installer Altova LicenseServer sur un autre ordinateur, décochez la case *Installer Altova LicenseServer sur l'appareil* et choisissez **Plus tard**. Dans ce cas, vous devez installer LicenseServer séparément et inscrire MapForce Server après.
- Si LicenseServer a déjà été installé sur votre ordinateur mais qu'il s'agit d'une version inférieure à celle indiquée par l'assistant d'installation, laissez les paramètres par défaut (pour mettre à jour à la version plus récente) tels quels. Dans ce cas, l'assistant d'installation mettra automatiquement votre version de LicenseServer à jour. L'information d'inscription et de licence existante sera reportée à la nouvelle version de LicenseServer.
- Si LicenseServer a été installé sur votre ordinateur ou sur votre réseau et a la même version que celle indiquée par l'assistant, suivez les étapes suivantes :
 - Décochez la case *Installer Altova LicenseServer sur l'appareil*.
 - Sous *Inscrire ce produit avec*, choisissez le serveur de licence avec lequel vous voulez vous inscrire MapForce Server. De manière alternative, choisissez **Plus tard**. Veuillez noter que vous pouvez toujours sélectionner **Plus tard** si vous voulez ignorer les associations de LicenseServer et poursuivre l'installation de MapForce Server.

Pour plus d'information sur l'inscription et la licence MapForce Server avec [Altova LicenseServer](#), voir la section [Licence MapForce Server](#)¹³.

Versions de LicenseServer

- Les produits de Altova doivent être dotés d'une licence soit () avec une version de LicenseServer qui correspond à la version installée MapForce Server, soit pour une version ultérieure de LicenseServer.
- La version LicenseServer qui correspond à la version actuelle de MapForce Server est **3.14**.
- Sur Windows, vous pouvez installer la version correspondante de LicenseServer comme faisant partie de l'installation de MapForce Server ou installez LicenseServer séparément. Sur Linux et macOS, vous devez installer LicenseServer séparément.
- Avant que la nouvelle version de LicenseServer ne soit installée, chaque version plus ancienne doit être désinstallée.

- Lors de la désinstallation de LicenseServer, toute l'information liée à l'inscription et à la licence contenue dans la version plus ancienne de LicenseServer sera enregistrée dans une base de données sur votre appareil de serveur. Ces données seront importées automatiquement dans la version plus nouvelle lorsque celle-ci est installée.
- Les versions LicenseServer sont rétro-compatibles. Elles fonctionneront avec des versions plus anciennes de MapForce Server.
- La dernière version de LicenseServer disponible sur le site web d'Altova. Cette version fonctionnera avec toute version actuelle ou ancienne de MapForce Server.
- Le numéro de version de la licence LicenseServer actuellement installée est indiqué au bas de la [page de configuration LicenseServer](#) (tous les onglets).

2.1.4 Licence MapForce Server (Windows)

Afin d'utiliser MapForce Server, il doit être doté d'une licence de Altova LicenseServer. La gestion des licences est une procédure à deux étapes :

1. **Inscrire MapForce Server** avec LicenseServer. L'inscription est réalisée depuis MapForce Server.
2. **Attribuer une licence** à MapForce Server depuis LicenseServer. Télécharger la dernière version de LicenseServer depuis le [site web d'Altova](#) et installez-la sur votre appareil local ou un appareil sur votre réseau.

Ces deux étapes sont décrites dans cette section. Pour toute information détaillée, voir le [manuel utilisateur LicenseServer](#) sur le [site web d'Altova](#).

2.1.4.1 Démarrer LicenseServer, MapForce Server

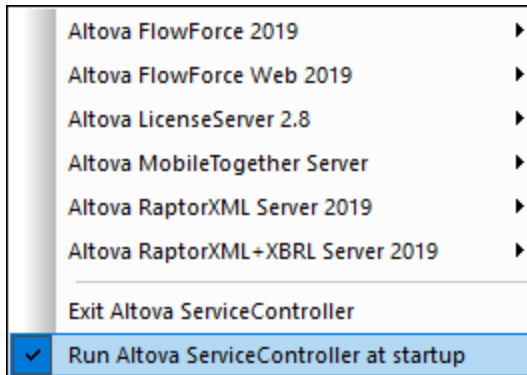
Altova LicenseServer (désigné LicenseServer) et MapForce Server sont tous les deux démarrés depuis Altova ServiceController.

Altova ServiceController

Altova ServiceController (ServiceController en abrégé) est une application permettant de lancer, d'arrêter et de configurer confortablement les services Altova **ms**. ServiceController est installé avec l'Altova LicenseServer et avec les produits de serveur Altova installés en tant que services (DiffDog Server, FlowForce Server, Mobile Together Server et RaptorXML(+XBRL) Server). ServiceController peut être accédé par le biais de la barre d'état système (*voir la capture d'écran ci-dessous*).

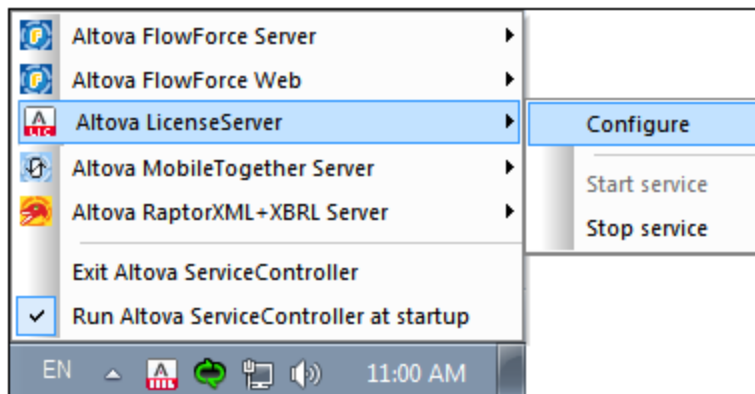


Pour spécifier que ServiceController démarre automatiquement lors de la connexion au système, cliquer sur l'icône **ServiceController** dans la zone de notification pour afficher le menu **ServiceController** (*voir la capture d'écran ci-dessous*) et basculez sur la commande **Run Altova ServiceController at Startup**. (Cette commande est activée par défaut.) Pour quitter ServiceController, cliquer sur l'icône **ServiceController** dans la zone de notification et, dans le menu qui apparaît (*voir la capture d'écran ci-dessous*), cliquez sur **Exit Altova ServiceController**.



Démarrer LicenseServer

Pour démarrer LicenseServer, cliquez sur l'icône **ServiceController** dans la zone de modification, passez au-dessus de **Altova LicenseServer** dans le menu qui s'ouvre (voir la capture d'écran ci-dessous), puis sélectionnez **Start Service** depuis le sous-menu de LicenseServer. Si LicenseServer est déjà en cours d'exécution, l'option *Démarrer le service* sera désactivée. Vous pouvez également arrêter le service par le biais de ServiceController.



2.1.4.2 Enregistrer MapForce Server

Afin de pouvoir détenir une licence MapForce Server depuis Altova LicenseServer, MapForce Server doit être enregistré avec LicenseServer

Pour enregistrer MapForce Server depuis l'interface de la ligne de commande, utilisez la commande `licenseserveret` et donnez l'adresse de l'appareil LicenseServer (voir ci-dessous).

```
MapForceServer licenseserver [options] ServerName-Or-IP-Address
```

Par exemple, si `localhost` est le nom du serveur sur lequel LicenseServer est installé, utilisez la commande suivante :

```
MapForceServer licenseserver localhost
```

Si MapForce Server a été installée faisant partie d'une installation de [FlowForce Server](#), enregistrer FlowForce Server avec LicenseServer enregistrera aussi automatiquement MapForce Server. Procédez comme suit : (i) démarrez Altova FlowForce Web par le biais de ServiceController (*voir les points précédents*); (ii) saisissez votre mot de passe pour accéder à la page Setup; (iii) sélectionnez le nom ou l'adresse du LicenseServer et cliquez sur **Enregistrer avec LicenseServer**. Pour plus d'informations, voir Enregistrer FlowForce Server.

Après avoir réussi l'enregistrement, allez à l'[onglet de gestion des clients de la page de configuration de LicenseServer](#) pour attribuer une licence à MapForce Server.

Pour plus d'informations sur l'enregistrement des produits Altova avec LicenseServer, voir le [manuel utilisateur de LicenseServer](#).

2.1.4.3 Attribuer des licences MapForce Server

Après avoir réussi l'inscription de MapForce Server, elle sera recensée dans l'onglet de gestion des clients de la page de configuration de LicenseServer. Allez-y et [attribuez une licence](#) à MapForce Server.

L'obtention de la licence des produits de serveur Altova, est basée sur le nombre de cœurs de processeurs disponibles sur la machine du produit. Par exemple, un processeur double cœur a deux cœurs, un processeur quadricœur a quatre cœurs, un processeur hexacœur a six cœurs, etc. Le nombre de cœurs pour lesquels une licence a été délivrée pour un produit doit être supérieur ou égal au nombre de cœurs disponibles sur cette machine de serveur, que ce serveur soit une machine physique ou virtuelle. Par exemple, si un serveur a huit cœurs (un processeur octacœur), vous devrez acheter au moins une licence octacœur. Vous pouvez aussi additionner les licences pour obtenir le nombre souhaité de cœurs. Ainsi, deux licences de quadricœurs peuvent être utilisées pour un serveur octacœur au lieu d'acheter une licence octacœur.

Si vous utilisez un serveur d'ordinateur avec un grand nombre de cœurs CPU, mais ne disposez que d'un faible volume à traiter, vous pouvez aussi créer une machine virtuelle qui disposera d'un plus petit nombre de cœurs et acheter une licence pour ce nombre de cœurs. Il va de soi que la vitesse de traitement d'un tel déploiement sera moins rapide que si tous les cœurs disponibles sur le serveur étaient utilisés.

Note : chaque licence de produit de serveur Altova peut être utilisée pour une seule machine client à la fois, même si la licence a une capacité de licence qui n'est pas utilisée (l'appareil client est l'appareil sur lequel le produit de serveur Altova est installé). Par exemple, si une licence de 10-cœurs est utilisée pour une machine client qui détient 6 cœurs CPU, les 4 cœurs restants de la capacité de licence ne pourront pas être utilisés simultanément pour une autre machine client.

Licences FlowForceServer et MapForceServer licensing

FlowForce Server Advanced Edition et MapForce Server Advanced Edition sont uniquement exécutés sur des appareils à deux ou plus de cœurs.

Lors de l'évaluation du nombre de cœurs que vous souhaitez mettre sous licence, cette décision doit inclure les volumes de données que vous souhaitez traiter et le temps de traitement escompté de votre environnement commercial. Dans la plupart des cas, un grand nombre de cœurs signifie plus de volumes de données traités en moins de temps. Ci-dessous, vous trouverez quelques astuces :

- FlowForce Server est exécuté en tant qu'application multi-thread. Si le nombre de requêtes récurrentes vers le serveur est important, un nombre insuffisant de cœurs entraînera des latences (temps

d'attente). Par exemple, si vous exposez des tâches en tant que services Web, il peut y avoir des centaines de requêtes concurrentes de la part des clients. Dans ce cas, FlowForce Server profitera grandement d'un nombre important de cœurs.

- MapForce Server utilisera un seul cœur à la fois, par mappage. Donc si vous devez exécuter plusieurs mappages simultanément, il est recommandé de disposer d'un grand nombre de cœurs. Par exemple, lorsque MapForce Server est exécuté sous une gestion FlowForce Server, plusieurs tâches de mappage peuvent se superposer et être exécutées en même temps, selon la configuration. Veuillez noter que si les volumes traités par vos mappages sont extrêmement gros, une latence peut tout de même se produire.

Outre les points mentionnés ci-dessus, veuillez noter qu'il y a plusieurs facteurs externes qui influent sur les volumes et les délais de traitement que votre serveur est en mesure de gérer (par exemple, le matériel, la charge actuelle du CPU, le volume de mémoire d'autres applications en cours sur le serveur). Afin d'obtenir les mesures de performance les plus précises, il est généralement conseillé tout d'abord d'exécuter les outils dans votre environnement puis de les exposer aux facteurs réels et aux données spécifiques à votre activité.

Exécution thread unique

Si un produit de serveur Altova permet une exécution single-thread, une option pour une *exécution single-thread* sera disponible. Dans ces cas, la licence produit du serveur Altova pour uniquement un cœur est disponible dans le pool des licences, un appareil avec des cœurs multiples peut être assigné à cette licence à one-core. Dans un tel cas, l'appareil exécutera ce produit en single-core. Le traitement sera donc plus lent car le multi-threading (qui est possible sur de multiples cœurs) ne sera pas disponible. Le produit sera exécuté en mode single thread sur cet appareil.

Pour assigner une licence single-core à un appareil multiple-core dans LicenseServer, sélectionnez la case à cocher *Limit to single thread execution* pour ce produit.

Estimation des exigences core

Il existe de nombreux facteurs externes divers qui influent sur les volumes de données et les temps de traitement que votre serveur arrive à gérer (par exemple : le matériel, la charge actuelle sur le CPU, et l'attribution de la mémoire d'autres applications exécutées sur le serveur). Afin de mesurer la performance aussi précisément que possible, testez les applications dans votre environnement avec les volumes de données et les conditions qui établissent aussi fidèlement que possible des situations professionnelles réelles.

2.2 Configuration sur Linux

Cette section décrit l'[installation](#)¹⁷ et la [licence](#)²⁰ de MapForce Server sur les systèmes Linux (Debian, Ubuntu, CentOS, RedHat).

Exigences de système (Linux)

- <% REDHAT%>
- <% CENTOS%>
- <% DEBIAN%>
- <% UBUNTU%>
- AlmaLinux 9.0
- Rocky Linux 9.0

Prérequis

- Réaliser l'installation soit en tant qu'utilisateur **root**, soit en tant qu'utilisateur avec des privilèges **sudo**.
- La version précédente de MapForce Server doit être désinstallée avant d'installer une nouvelle.
- Les bibliothèques suivantes sont requises comme prérequis pour installer et exécuter l'application. Si les packages ci-dessous ne sont pas déjà disponibles sur l'appareil Linux, exécutez la commande `yum` (ou `apt-get`, si applicable) pour les installer.

CentOS, RedHat	Debian	Ubuntu
krb5-libs	libgssapi-krb5-2	libgssapi-krb5-2

2.2.1 Installer sur Linux

MapForce Server est disponible pour l'installation sur les systèmes Linux. Réaliser installation soit en tant qu'utilisateur `root`, soit en tant qu'utilisateur avec des privilèges `sudo`.

Intégration de FlowForce Server et d'autres produits de serveur d'Altova

Si vous installez MapForce Server ensemble avec FlowForce Server, il est recommandé d'installer d'abord FlowForce Server. Si vous installez MapForce Server avant FlowForce Server, il faut, après avoir installé MapForce Server et FlowForce Server, exécuter la commande suivante :

```
cp /opt/Altova/MapForceServer2024/etc/*.tool /opt/Altova/FlowForceServer2024/tools
```

Cette commande copie le fichier `.tool` depuis le répertoire `/etc` de MapForce Server vers le répertoire FlowForce Server `/tools`. Le fichier `.tool` est requis par FlowForce Server. Il contient le chemin vers l'MapForce Server exécutable. Vous n'avez pas besoin d'exécuter cette commande si vous installez FlowForce Server avant d'installer MapForce Server.

Désinstaller MapForce Server

Avant d'installer MapForce Server, vous devez désinstaller toute version plus ancienne.

Pour vérifier quels produits de serveur Altova sont installés :

```
[Debian, Ubuntu] : dpkg --list | grep Altova
[CentOS, RedHat] : rpm -qa | grep server
```

Pour désinstaller une version plus ancienne de MapForce Server:

```
[Debian, Ubuntu] : sudo dpkg --remove mapforceserveradv
[CentOS, RedHat] : sudo rpm -e mapforceserveradv
```

Sur les systèmes Debian et Ubuntu, il se pourrait que MapForce Server apparaisse encore dans la liste des produits installés après son installation. Dans ce cas, exécutez la commande `purge` pour effacer MapForce Server de la liste. Vous pouvez aussi utiliser la commande `purge` au lieu de la commande `remove` recensée ci-dessus.

```
[Debian, Ubuntu] : sudo dpkg --purge mapforceserveradv
```

Télécharger le package Linux MapForce Server

Les packages d'installation pour MapForce Server pour les systèmes Linux suivants sont disponibles sur le [site web d'Altova](#).

Distribution	Extension de package
Debian	.deb
Ubuntu	.deb
CentOS	.rpm
RedHat	.rpm

Après avoir téléchargé le package Linux, copiez-le dans n'importe quel répertoire sur le système Linux. Puisque vous allez devoir mettre sous licence MapForce Server with an [Altova LicenseServer](#), vous téléchargerez LicenseServer depuis le [site web d'Altova](#) en même temps que vous téléchargez MapForce Server, plutôt que de le télécharger plus tard.

Installer MapForce Server

Dans une fenêtre de terminal, basculez vers le répertoire où vous copiez le package Linux. Par exemple, si vous le copiez dans un répertoire d'utilisateur appelé `MyAltova` qui est situé dans le répertoire `/home/User`, basculez vers ce répertoire comme suit :

```
cd /home/User/MyAltova
```

Installer MapForce Server utilisant la commande pertinente :

```
[Debian] : sudo dpkg --install mapforceserveradv-2024-debian.deb
[Ubuntu] : sudo dpkg --install mapforceserveradv-2024-ubuntu.deb
[CentOS] : sudo rpm -ivh mapforceserveradv-2024-1.x86_64.rpm
[RedHat] : sudo rpm -ivh mapforceserveradv-2024-1.x86_64.rpm
```

Vous devrez éventuellement ajuster le nom du package ci-dessus pour qu'il corresponde au release ou à la version service pack actuels.

Le package MapForce Server sera installé dans le dossier suivant :

```
/opt/Altova/MapForceServer2024
```

2.2.2 Installer LicenseServer (Linux)

Pour que MapForce Server fonctionne, il faut qu'il ait une licence par le biais d'un [Altova LicenseServer](#) sur votre réseau. Téléchargez LicenseServer depuis le [site web d'Altova](#) et copiez le package dans n'importe quel répertoire. Installez-le comme vous avez installé MapForce Server (voir [page précédente](#)¹⁷).

```
[Debian] : sudo dpkg --install licenseserver-3.14-debian.deb
[Ubuntu] : sudo dpkg --install licenseserver-3.14-ubuntu.deb
[CentOS] : sudo rpm -ivh licenseserver-3.14-1.x86_64.rpm
[RedHat] : sudo rpm -ivh licenseserver-3.14-1.x86_64.rpm
```

Le pack de LicenseServer sera installé dans le chemin suivant :

```
/opt/Altova/LicenseServer
```

Pour plus d'information sur l'inscription et la licence MapForce Server avec [Altova LicenseServer](#), voir la section [Licence MapForce Server](#)²⁰. Voir également la [documentation LicenseServer](#) pour des informations plus détaillées.

Versions de LicenseServer

- Les produits de Altova doivent être dotés d'une licence soit () avec une version de LicenseServer qui correspond à la version installée MapForce Server, soit pour une version ultérieure de LicenseServer.
- La version LicenseServer qui correspond à la version actuelle de MapForce Server est **3.14**.
- Sur Windows, vous pouvez installer la version correspondante de LicenseServer comme faisant partie de l'installation de MapForce Server ou installez LicenseServer séparément. Sur Linux et macOS, vous devez installer LicenseServer séparément.
- Avant que la nouvelle version de LicenseServer ne soit installée, chaque version plus ancienne doit être désinstallée.
- Lors de la désinstallation de LicenseServer, toute l'information liée à l'inscription et à la licence contenue dans la version plus ancienne de LicenseServer sera enregistrée dans une base de données sur votre appareil de serveur. Ces données seront importées automatiquement dans la version plus nouvelle lorsque celle-ci est installée.
- Les versions LicenseServer sont rétro-compatibles. Elles fonctionneront avec des versions plus anciennes de MapForce Server.
- La dernière version de LicenseServer disponible sur le site web d'Altova. Cette version fonctionnera avec toute version actuelle ou ancienne de MapForce Server.
- Le numéro de version de la licence LicenseServer actuellement installée est indiqué au bas de la [page de configuration LicenseServer](#) (tous les onglets).

2.2.3 Licence MapForce Server (Linux)

Afin d'utiliser MapForce Server, il doit être doté d'une licence de Altova LicenseServer. La gestion des licences est une procédure à deux étapes :

1. **Inscrire MapForce Server** avec LicenseServer. L'inscription est réalisée depuis MapForce Server.
2. **Attribuer une licence** à MapForce Server depuis LicenseServer. Télécharger la dernière version de LicenseServer depuis le [site web d'Altova](#) et installez-la sur votre appareil local ou un appareil sur votre réseau.

Ces deux étapes sont décrites dans cette section. Pour toute information détaillée, voir le [manuel utilisateur LicenseServer](#) sur le [site web d'Altova](#).

2.2.3.1 Démarrer LicenseServer, MapForce Server

Démarrer Altova LicenseServer et MapForce Server soit comme utilisateur `root` ou comme utilisateur avec des privilèges `sudo`.

Démarrer LicenseServer

Pour s'enregistrer et gérer la licence MapForce Server avec LicenseServer, celui-ci doit être exécuté en tant que daemon sur le réseau. Démarrez LicenseServer en tant que daemon avec la commande suivante :

```
sudo systemctl start licenseserver
```

Si à un moment ou un autre, vous êtes amenés à devoir arrêter LicenseServer, remplacez `start` par `arrêter` dans les commandes ci-dessus. Par exemple :

```
sudo systemctl stop licenseserver
```

2.2.3.2 Enregistrer MapForce Server

Afin de pouvoir détenir une licence MapForce Server depuis Altova LicenseServer, MapForce Server doit être enregistré avec LicenseServer

Pour enregistrer MapForce Server, utilisez la commande `licenseserver` :

```
sudo /opt/Altova/MapForceServer2024/bin/mapforceserver licenseserver [options]  
ServerName-Or-IP-Address
```

Par exemple, si `localhost` est le nom du serveur sur lequel LicenseServer est installé :

```
sudo /opt/Altova/MapForceServer2024/bin/mapforceserver licenseserver localhost
```

Dans la commande ci-dessus, `localhost` est le nom du serveur sur lequel LicenseServer est installé. Notez également que l'emplacement de l'MapForce Server exécutable est :

```
/opt/Altova/MapForceServer2024/bin/
```

Après avoir réussi l'enregistrement, allez à l'[onglet de gestion des clients de la page de configuration de LicenseServer](#) pour attribuer une licence à MapForce Server.

Pour plus d'informations sur l'enregistrement des produits Altova avec LicenseServer, voir le [manuel utilisateur de LicenseServer](#).

2.2.3.3 Licence MapForce Server

Après avoir réussi l'inscription de MapForce Server, elle sera recensée dans l'onglet de gestion des clients de la page de configuration de LicenseServer. Allez-y et [attribuez une licence](#) à MapForce Server.

L'obtention de la licence des produits de serveur Altova, est basée sur le nombre de cœurs de processeurs disponibles sur la machine du produit. Par exemple, un processeur double cœur a deux cœurs, un processeur quadricœur a quatre cœurs, un processeur hexacœur a six cœurs, etc. Le nombre de cœurs pour lesquels une licence a été délivrée pour un produit doit être supérieur ou égal au nombre de cœurs disponibles sur cette machine de serveur, que ce serveur soit une machine physique ou virtuelle. Par exemple, si un serveur a huit cœurs (un processeur octacœur), vous devrez acheter au moins une licence octacœur. Vous pouvez aussi additionner les licences pour obtenir le nombre souhaité de cœurs. Ainsi, deux licences de quadricœurs peuvent être utilisées pour un serveur octacœur au lieu d'acheter une licence octacœur.

Si vous utilisez un serveur d'ordinateur avec un grand nombre de cœurs CPU, mais ne disposez que d'un faible volume à traiter, vous pouvez aussi créer une machine virtuelle qui disposera d'un plus petit nombre de cœurs et acheter une licence pour ce nombre de cœurs. Il va de soi que la vitesse de traitement d'un tel déploiement sera moins rapide que si tous les cœurs disponibles sur le serveur étaient utilisés.

Note : chaque licence de produit de serveur Altova peut être utilisée pour une seule machine client à la fois, même si la licence a une capacité de licence qui n'est pas utilisée (l'appareil client est l'appareil sur lequel le produit de serveur Altova est installé). Par exemple, si une licence de 10-cœurs est utilisée pour une machine client qui détient 6 cœurs CPU, les 4 cœurs restants de la capacité de licence ne pourront pas être utilisés simultanément pour une autre machine client.

Licences FlowForceServer et MapForceServer licensing

FlowForce Server Advanced Edition et MapForce Server Advanced Edition sont uniquement exécutés sur des appareils à deux ou plus de cœurs.

Lors de l'évaluation du nombre de cœurs que vous souhaitez mettre sous licence, cette décision doit inclure les volumes de données que vous souhaitez traiter et le temps de traitement escompté de votre environnement commercial. Dans la plupart des cas, un grand nombre de cœurs signifie plus de volumes de données traités en moins de temps. Ci-dessous, vous trouverez quelques astuces :

- FlowForce Server est exécuté en tant qu'application multi-thread. Si le nombre de requêtes récurrentes vers le serveur est important, un nombre insuffisant de cœurs entraînera des latences (temps d'attente). Par exemple, si vous exposez des tâches en tant que services Web, il peut y avoir des centaines de requêtes concurrentes de la part des clients. Dans ce cas, FlowForce Server profitera grandement d'un nombre important de cœurs.
- MapForce Server utilisera un seul cœur à la fois, par mappage. Donc si vous devez exécuter plusieurs mappages simultanément, il est recommandé de disposer d'un grand nombre de cœurs. Par exemple, lorsque MapForce Server est exécuté sous une gestion FlowForce Server, plusieurs tâches de

mappage peuvent se superposer et être exécutées en même temps, selon la configuration. Veuillez noter que si les volumes traités par vos mappages sont extrêmement gros, une latence peut tout de même se produire.

Outre les points mentionnés ci-dessus, veuillez noter qu'il y a plusieurs facteurs externes qui influent sur les volumes et les délais de traitement que votre serveur est en mesure de gérer (par exemple, le matériel, la charge actuelle du CPU, le volume de mémoire d'autres applications en cours sur le serveur). Afin d'obtenir les mesures de performance les plus précises, il est généralement conseillé tout d'abord d'exécuter les outils dans votre environnement puis de les exposer aux facteurs réels et aux données spécifiques à votre activité.

Exécution thread unique

Si un produit de serveur Altova permet une exécution single-thread, une option pour une *exécution single-thread* sera disponible. Dans ces cas, la licence produit du serveur Altova pour uniquement un cœur est disponible dans le pool des licences, un appareil avec des cœurs multiples peut être assigné à cette licence à one-core. Dans un tel cas, l'appareil exécutera ce produit en single-core. Le traitement sera donc plus lent car le multi-threading (qui est possible sur de multiples cœurs) ne sera pas disponible. Le produit sera exécuté en mode single thread sur cet appareil.

Pour assigner une licence single-core à un appareil multiple-core dans LicenseServer, sélectionnez la case à cocher *Limit to single thread execution* pour ce produit.

Estimation des exigences core

Il existe de nombreux facteurs externes divers qui influent sur les volumes de données et les temps de traitement que votre serveur arrive à gérer (par exemple : le matériel, la charge actuelle sur le CPU, et l'attribution de la mémoire d'autres applications exécutées sur le serveur). Afin de mesurer la performance aussi précisément que possible, testez les applications dans votre environnement avec les volumes de données et les conditions qui établissent aussi fidèlement que possible des situations professionnelles réelles.

2.3 Configuration sur macOS

Cette section décrit l'[installation](#)²³ et la [licence](#)²⁵ de MapForce Server sur les systèmes macOS.

Exigences de système (macOS)

Notez les exigences système suivantes :

- macOS 12 ou plus récent

Prérequis

Notez les exigences préalables suivantes :

- Assurez-vous d'avoir installé Altova LicenseServer et qu'il est exécuté.
- Réaliser l'installation soit en tant qu'utilisateur `root`, soit en tant qu'utilisateur avec des privilèges `sudo`.
- La version précédente de MapForce Server doit être désinstallée avant d'installer une nouvelle.
- L'appareil macOS doit être configuré de telle façon que son nom se résout en adresse IP. Ceci signifie que vous devez être en mesure d'effectuer un ping avec succès du nom de l'hôte depuis le terminal en utilisant la commande `ping <hostname>`.

2.3.1 Installer sur macOS

Ce chapitre décrit l'installation et la configuration de MapForce Server sur les systèmes macOS.

Intégration dans FlowForce

Si vous installez MapForce Server ensemble avec FlowForce Server, il est recommandé d'installer d'abord FlowForce Server. Si vous installez MapForce Server avant FlowForce Server, il faut, après avoir installé, exécuter la commande suivante :

```
cp /usr/local/Altova/MapForceServer2024/etc/*.tool /usr/local/Altova/FlowForceServer2024/tools
```

Cette commande copie le fichier `.tool` depuis le répertoire `/etc` de MapForce Server vers le répertoire FlowForce Server `/tools`. Le fichier `.tool` est requis par FlowForce Server. Il contient le chemin vers l'MapForce Server exécutable. Vous n'avez pas besoin d'exécuter cette commande si vous installez FlowForce Server avant d'installer MapForce Server.

Désinstaller MapForce Server

Dans le dossier Applications, cliquez avec la touche de droite sur l'icône MapForce Server et choisissez **Déplacer vers la Corbeille**. L'application sera déplacée dans la corbeille. Vous allez toutefois devoir déplacer l'application du dossier `usr`. Pour ce faire, utilisez la commande suivante :

```
sudo rm -rf /usr/local/Altova/MapForceServer2024/
```

Si vous devez désinstaller une ancienne version de Altova LicenseServer, vous devez d'abord arrêter son exécution en tant que service. Pour ce faire, utilisez la commande suivante :

```
sudo launchctl unload /Library/LaunchDaemons/com.altova.LicenseServer.plist
```

Pour vérifier si le service a été arrêté, ouvrez le moniteur d'activités et assurez-vous que LicenseServer n'est pas dans la liste. Puis, procédez à la désinstallation de la même manière que décrit ci-dessus pour MapForce Server.

Installer MapForce Server

Pour installer MapForce Server, suivez les étapes suivantes :

1. Téléchargez le fichier de l'image de disque (.dmg) de MapForce Server depuis le site Altova website (<http://www.altova.com/download.html>).
2. Cliquez pour ouvrir l'image de disque téléchargée (.dmg). Ceci fait que le programme d'installation MapForce Server apparaît en tant que nouveau lecteur virtuel sur votre ordinateur.
3. Sur le nouveau lecteur virtuel, double-cliquez sur le package d'installation (.pkg).
4. Suivez les étapes successives de l'assistant d'installation. Celles-ci sont explicites et incluent un étape lors de laquelle vous devez accepter le contrat de licence avant de pouvoir continuer. *Voir aussi [Licensing MapForce Server](#)*²⁵.
5. Pour éjecter le lecteur après l'installation, cliquez de la touche droite sur le lecteur et sélectionnez **Éjecter**.

Le package MapForce Server sera installé dans le dossier :

```
/usr/local/Altova/MapForceServer2024 (application binaries)  
/var/Altova/MapForceServer (data files : database and logs)
```

Le serveur daemonMapForce Server démarre automatiquement après l'installation et redémarre l'appareil. Vous pouvez toujours démarrer MapForce Server en tant que daemon avec les commandes suivantes :

```
sudo launchctl load /Library/LaunchDaemons/com.altova.MapForceServer2024.plist
```

2.3.2 Installer LicenseServer (macOS)

Altova LicenseServer peut être téléchargé depuis le site web d'Altova (<http://www.altova.com/download.html>). Exécutez l'installation tel que décrit [ici](#)²³.

Le package de LicenseServer sera installé dans le suivantes dossier :

```
/usr/local/Altova/LicenseServer
```

Pour toute information relative à l'enregistrement de MapForce Server avec [Altova LicenseServer](#) et à la licence, voir [Licence sur macOS](#)²⁵.

Versions de LicenseServer

- Les produits de Altova doivent être dotés d'une licence soit () avec une version de LicenseServer qui correspond à la version installée MapForce Server, soit pour une version ultérieure de LicenseServer.

- La version LicenseServer qui correspond à la version actuelle de MapForce Server est **3.14**.
- Sur Windows, vous pouvez installer la version correspondante de LicenseServer comme faisant partie de l'installation de MapForce Server ou installez LicenseServer séparément. Sur Linux et macOS, vous devez installer LicenseServer séparément.
- Avant que la nouvelle version de LicenseServer ne soit installée, chaque version plus ancienne doit être désinstallée.
- Lors de la désinstallation de LicenseServer, toute l'information liée à l'inscription et à la licence contenue dans la version plus ancienne de LicenseServer sera enregistrée dans une base de données sur votre appareil de serveur. Ces données seront importées automatiquement dans la version plus nouvelle lorsque celle-ci est installée.
- Les versions LicenseServer sont rétro-compatibles. Elles fonctionneront avec des versions plus anciennes de MapForce Server.
- La dernière version de LicenseServer disponible sur le site web d'Altova. Cette version fonctionnera avec toute version actuelle ou ancienne de MapForce Server.
- Le numéro de version de la licence LicenseServer actuellement installée est indiqué au bas de la [page de configuration LicenseServer](#) (tous les onglets).

2.3.3 Licence MapForce Server (macOS)

Afin d'utiliser MapForce Server, il doit être doté d'une licence de Altova LicenseServer. La gestion des licences est une procédure à deux étapes :

1. **Inscrire MapForce Server** avec LicenseServer. L'inscription est réalisée depuis MapForce Server.
2. **Attribuer une licence** à MapForce Server depuis LicenseServer. Télécharger la dernière version de LicenseServer depuis le [site web d'Altova](#) et installez-la sur votre appareil local ou un appareil sur votre réseau.

Ces deux étapes sont décrites dans cette section. Pour toute information détaillée, voir le [manuel utilisateur LicenseServer](#) sur le [site web d'Altova](#).

2.3.3.1 Démarrer LicenseServer, MapForce Server

Démarrer Altova LicenseServer et MapForce Server soit comme utilisateur `root` ou comme utilisateur avec des privilèges `sudo`.

Démarrer LicenseServer

Pour s'enregistrer et gérer la licence MapForce Server correctement avec LicenseServer, celui-ci doit être exécuté en tant que daemon. Lancez LicenseServer en tant que daemon avec la commande suivante :

```
sudo launchctl load /Library/LaunchDaemons/com.altova.LicenseServer.plist
```

Si à un moment ou un autre, vous êtes amenés à devoir arrêter LicenseServer, remplacez `charger` par `unload` dans les commandes ci-dessus.

2.3.3.2 Enregistrer MapForce Server

Afin de pouvoir détenir une licence MapForce Server depuis Altova LicenseServer, MapForce Server doit être enregistré avec LicenseServer

Pour enregistrer MapForce Server depuis l'interface de ligne de commande, utilisez la commande `licenseserver` :

```
sudo /usr/local/Altova/MapForceServer2024/bin/MapForceServer licenseserver [options]
ServerName-Or-IP-Address
```

Par exemple, si `localhost` est le nom du serveur sur lequel LicenseServer est installé :

```
sudo /usr/local/Altova/MapForceServer2024/bin/MapForceServer licenseserver localhost
```

Dans la commande ci-dessus, `localhost` est le nom du serveur sur lequel LicenseServer est installé. Notez également que l'emplacement de l'MapForce Server exécutable est :

```
/usr/local/Altova/MapForceServer2024/bin/
```

Après avoir réussi l'enregistrement, allez à l'[onglet de gestion des clients de la page de configuration de LicenseServer](#) pour attribuer une licence à MapForce Server.

Pour plus d'informations sur l'enregistrement des produits Altova avec LicenseServer, voir le [manuel utilisateur de LicenseServer](#).

2.3.3.3 License MapForce Server

Après avoir réussi l'inscription de MapForce Server, elle sera recensée dans l'onglet de gestion des clients de la page de configuration de LicenseServer. Allez-y et [attribuez une licence](#) à MapForce Server.

L'obtention de la licence des produits de serveur Altova, est basée sur le nombre de cœurs de processeurs disponibles sur la machine du produit. Par exemple, un processeur double cœur a deux cœurs, un processeur quadricœur a quatre cœurs, un processeur hexacœur a six cœurs, etc. Le nombre de cœurs pour lesquels une licence a été délivrée pour un produit doit être supérieur ou égal au nombre de cœurs disponibles sur cette machine de serveur, que ce serveur soit une machine physique ou virtuelle. Par exemple, si un serveur a huit cœurs (un processeur octacœur), vous devrez acheter au moins une licence octacœur. Vous pouvez aussi additionner les licences pour obtenir le nombre souhaité de cœurs. Ainsi, deux licences de quadricœurs peuvent être utilisées pour un serveur octacœur au lieu d'acheter une licence octacœur.

Si vous utilisez un serveur d'ordinateur avec un grand nombre de cœurs CPU, mais ne disposez que d'un faible volume à traiter, vous pouvez aussi créer une machine virtuelle qui disposera d'un plus petit nombre de cœurs et acheter une licence pour ce nombre de cœurs. Il va de soi que la vitesse de traitement d'un tel déploiement sera moins rapide que si tous les cœurs disponibles sur le serveur étaient utilisés.

Note : chaque licence de produit de serveur Altova peut être utilisée pour une seule machine client à la fois, même si la licence a une capacité de licence qui n'est pas utilisée (l'appareil client est l'appareil sur lequel le produit de serveur Altova est installé). Par exemple, si une licence de 10-cœurs est utilisée pour une machine client qui détient 6 cœurs CPU, les 4 cœurs restants de la capacité de licence ne pourront pas être utilisés simultanément pour une autre machine client.

Licences FlowForceServer et MapForceServer licensing

FlowForce Server Advanced Edition et MapForce Server Advanced Edition sont uniquement exécutés sur des appareils à deux ou plus de cœurs.

Lors de l'évaluation du nombre de cœurs que vous souhaitez mettre sous licence, cette décision doit inclure les volumes de données que vous souhaitez traiter et le temps de traitement escompté de votre environnement commercial. Dans la plupart des cas, un grand nombre de cœurs signifie plus de volumes de données traités en moins de temps. Ci-dessous, vous trouverez quelques astuces :

- FlowForce Server est exécuté en tant qu'application multi-thread. Si le nombre de requêtes récurrentes vers le serveur est important, un nombre insuffisant de cœurs entraînera des latences (temps d'attente). Par exemple, si vous exposez des tâches en tant que services Web, il peut y avoir des centaines de requêtes concurrentes de la part des clients. Dans ce cas, FlowForce Server profitera grandement d'un nombre important de cœurs.
- MapForce Server utilisera un seul cœur à la fois, par mappage. Donc si vous devez exécuter plusieurs mappages simultanément, il est recommandé de disposer d'un grand nombre de cœurs. Par exemple, lorsque MapForce Server est exécuté sous une gestion FlowForce Server, plusieurs tâches de mappage peuvent se superposer et être exécutées en même temps, selon la configuration. Veuillez noter que si les volumes traités par vos mappages sont extrêmement gros, une latence peut tout de même se produire.

Outre les points mentionnés ci-dessus, veuillez noter qu'il y a plusieurs facteurs externes qui influent sur les volumes et les délais de traitement que votre serveur est en mesure de gérer (par exemple, le matériel, la charge actuelle du CPU, le volume de mémoire d'autres applications en cours sur le serveur). Afin d'obtenir les mesures de performance les plus précises, il est généralement conseillé tout d'abord d'exécuter les outils dans votre environnement puis de les exposer aux facteurs réels et aux données spécifiques à votre activité.

Exécution thread unique

Si un produit de serveur Altova permet une exécution single-thread, une option pour une *exécution single-thread* sera disponible. Dans ces cas, la licence produit du serveur Altova pour uniquement un cœur est disponible dans le pool des licences, un appareil avec des cœurs multiples peut être assigné à cette licence à one-core. Dans un tel cas, l'appareil exécutera ce produit en single-core. Le traitement sera donc plus lent car le multi-threading (qui est possible sur de multiples cœurs) ne sera pas disponible. Le produit sera exécuté en mode single thread sur cet appareil.

Pour assigner une licence single-core à un appareil multiple-core dans LicenseServer, sélectionnez la case à cocher *Limit to single thread execution* pour ce produit.

Estimation des exigences core

Il existe de nombreux facteurs externes divers qui influent sur les volumes de données et les temps de traitement que votre serveur arrive à gérer (par exemple : le matériel, la charge actuelle sur le CPU, et l'attribution de la mémoire d'autres applications exécutées sur le serveur). Afin de mesurer la performance aussi précisément que possible, testez les applications dans votre environnement avec les volumes de données et les conditions qui établissent aussi fidèlement que possible des situations professionnelles réelles.

2.4 Mise à jour MapForce Server

La manière la plus simple de reporter une licence depuis la version précédente de MapForce Server vers la version plus récente est par le biais du processus d'installation. Les étapes clés au cours de l'installation sont :

1. Enregistrer la nouvelle version de MapForce Server avec le serveur de licence qui détient la licence utilisée par l'ancienne version de MapForce Server.
2. Acceptez le contrat de licence de MapForce Server. (Si vous n'acceptez pas le contrat, la nouvelle version ne sera pas installée.)

Note : Si vous n'inscrivez pas MapForce Server avec LicenseServer au cours du processus d'installation, vous pouvez faire ceci plus tard et ensuite compléter le processus de licence.

2.5 Migrer MapForce Server vers un nouvel appareil

Si vous voulez migrer MapForce Server depuis un appareil vers un autre (y compris sur des plateformes prises en charge), suivez les directives ci-dessous.

Migrer MapForce Server vers un nouvel appareil consiste à réattribuer la licence depuis l'ancien appareil vers le nouveau. Pour ce faire, procédez comme suit :

1. Installez MapForce Server sur votre nouvel appareil. Si l'installation a déjà été réalisée en tant que partie de l'installation de FlowForce Server, ignorez cette étape.
2. Sur le nouvel appareil, enregistrez MapForce Server avec Altova LicenseServer.
3. Sur l'ancien appareil, assurez-vous qu'aucun client n'utilise le serveur (par exemple, qu'aucun mappage ne soit exécuté).
4. Ouvrir la page d'administration de Altova LicenseServer. Désactivez la licence de l'ancien appareil MapForce Server et réattribuez-la au nouvel appareil.

3 Installation

3.1 Exigences du système

▼ Windows

Windows 10, Windows 11

▼ Windows Server

Windows Server 2016 ou plus récent

Sur Windows, MapForce Server est disponible aussi bien en tant que paquet 32-bit que 64-bit.

Exigences de système (Linux)

- <% REDHAT%>
- <% CENTOS%>
- <% DEBIAN%>
- <% UBUNTU%>
- AlmaLinux 9.0
- Rocky Linux 9.0

Prérequis

- Réaliser l'installation soit en tant qu'utilisateur **root**, soit en tant qu'utilisateur avec des privilèges **sudo**.
- La version précédente de MapForce Server doit être désinstallée avant d'installer une nouvelle.
- Les bibliothèques suivantes sont requises comme prérequis pour installer et exécuter l'application. Si les packages ci-dessous ne sont pas déjà disponibles sur l'appareil Linux, exécutez la commande `yum` (ou `apt-get`, si applicable) pour les installer.

CentOS, RedHat	Debian	Ubuntu
krb5-libs	libgssapi-krb5-2	libgssapi-krb5-2

▼ macOS

macOS 10.15 ou plus récent

3.2 Cœurs de processeur et licences

La mise sous licence des produits de serveur Altova se base sur le nombre de **cœurs de processeur physiques** disponibles sur l'appareil du produit (contrairement au nombre de cœurs logiques). Par exemple, un processus dual a deux cœurs, un processeur quadri-cœurs en a quatre, un processeur hexa-cœur en a six, etc. Le nombre de cœurs mis sous licence pour un produit doit être supérieur ou égal au nombre des cœurs disponibles dans ce serveur, que le serveur soit un appareil physique ou virtuel. Par exemple, si un serveur dispose de huit cœurs, vous devez acheter une licence à 8-cœurs. Vous pouvez aussi combiner les licences pour obtenir le décompte des cœurs. Donc des licence à 4-cœurs peuvent être utilisées pour un serveur à 8-cœurs au lieu d'une licence à 8-cœurs.

Si vous utilisez un serveur d'ordinateur avec un grand nombre de cœurs CPU mais que vous n'avez qu'un faible volume à traiter, vous pouvez aussi créer une machine virtuelle à laquelle un petit nombre de cœurs est attribué, et acheter une licence pour ce nombre. Il va sans dire qu'un tel déploiement présentera une vitesse de traitement plus faible que si tous les cœurs disponibles de l'ordinateur sont utilisés.

Note: Chaque licence de produit de serveur Altova peut être utilisé pour un seul appareil client (l'appareil sur lequel le produit de serveur Altova est installé) à la fois, même si la licence dispose de capacités de mise sous licence non-utilisées. Par exemple, si une licence à 10-cœurs est utilisée pour un appareil client qui a 6 cœurs CPU, les 4 cœurs restant de capacité de licence ne peuvent pas être utilisés simultanément pour un autre appareil client.

Estimation des exigences de cœur

Il existe plusieurs facteurs externes qui influent sur les volumes de données et sur les délais de traitement que votre serveur est en mesure de gérer (par exemple : le matériel, la charge actuelle du CPU et l'attribution de mémoire d'autres applications en cours sur le serveur). Pour pouvoir mesurer la performance aussi précisément que possible, tester les applications dans votre environnement avec des volumes de données, et dans des conditions qui se rapprochent au maximum des véritables situations de votre entreprise.

3.3 Chemins importants

Répertoire d'installation

L'application est installée dans le répertoire suivant :

Linux /opt/Altova/MapForceServer2024/bin
Mac /usr/local/Altova/MapForceServer2024/bin
Windows <ProgramFilesFolder>\Altova\MapForceServer2024\bin

Programme d'exécution

Après l'installation, vous pourrez accéder au programme d'exécution MapForce Server sous le chemin suivant :

Linux /opt/Altova/MapForceServer2024/bin/**mapforceserver**
Mac /usr/local/Altova/MapForceServer2024/bin/**mapforceserver**
Windows <ProgramFilesFolder>\Altova\MapForceServer2024\bin**MapForceServer.exe**

Données de configuration de l'application

Les données de configuration de l'application sont stockées dans le répertoire suivant :

Linux /var/opt/Altova/MapForceServer
Mac /var/Altova/MapForceServer
Windows C:\ProgramData\Altova\MapForceServer

3.4 Linux

3.4.1 Installer MapForce Server

Exigences du système

Exigences de système (Linux)

- <% REDHAT%>
- <% CENTOS%>
- <% DEBIAN%>
- <% UBUNTU%>
- AlmaLinux 9.0
- Rocky Linux 9.0

Prérequis

- Réaliser l'installation soit en tant qu'utilisateur **root**, soit en tant qu'utilisateur avec des privilèges **sudo**.
- La version précédente de MapForce Server doit être désinstallée avant d'installer une nouvelle.
- Les bibliothèques suivantes sont requises comme prérequis pour installer et exécuter l'application. Si les packages ci-dessous ne sont pas déjà disponibles sur l'appareil Linux, exécutez la commande `yum` (ou `apt-get`, si applicable) pour les installer.

CentOS, RedHat	Debian	Ubuntu
krb5-libs	libgssapi-krb5-2	libgssapi-krb5-2

Conditions préalables

- Effectuer l'installation soit en tant qu'utilisateur **root** soit en tant qu'utilisateur disposant de privilèges **sudo**.
- Assurez-vous que vous avez désinstallé MapForce Server avant d'installer la nouvelle version.

Installer MapForce Server

1. Télécharger le package d'installation depuis <https://www.altova.com/download#server> sur un répertoire local.
2. Puis se rendre dans le répertoire sur lequel vous avez téléchargé le pack d'installation FlowForce Server, par exemple :

```
cd /home/User/Downloads
```

3. Installer le package MapForce Server.

```
[Debian, Ubuntu] sudo dpkg --install mapforceserver-2024-debian.deb
[CentOS, RedHat] sudo rpm -ivh mapforceserver-2024-1.x86_64.rpm
```

Note: Vous devrez éventuellement ajuster le nom du package ci-dessus pour qu'il corresponde à la

publication actuelle ou à la version du service pack.

Mettre MapForce Server sous licence

1. S'assurer que la dernière version d'Altova LicenseServer (<https://www.altova.com/licenseserver>) est installée et en cours d'exécution soit sur l'appareil local ou le réseau.
2. S'enregistrer sur MapForce Server avec Altova LicenseServer. Pour ce faire, vous devez connaître soit le nom d'hôte ou l'adresse IP de l'ordinateur sur lequel LicenseServer est exécuté. Par exemple, si LicenseServer est exécuté sur 127.0.0.1, vous pouvez l'enregistrer comme suit :

```
sudo ./mapforceserver licenseserver 127.0.0.1
```

Note : La commande ci-dessus part du principe que vous appelez le programme d'exécution depuis le répertoire d'installation du programme, voir aussi [Chemins importants](#) ³²

3. Se connecter à l'interface d'administration sur LicenseServer et attribuer la licence sur l'appareil sur laquelle MapForce Server est exécutée.

Pour plus d'informations à propos des licences, se référer à la documentation Altova LicenseServer (<https://www.altova.com/fr/documentation>).

3.4.2 Consulter les produits installés actuellement

To check if an Altova server product is installed, run the following command:

```
[Debian, Ubuntu] dpkg --get-selections | grep Altova  
[CentOS, RedHat] rpm -qa | grep server
```

3.4.3 Désinstaller MapForce Server

To uninstall MapForce Server, run the following command:

```
[Debian, Ubuntu] sudo dpkg --purge mapforceserveradv  
[CentOS, RedHat] sudo rpm -e mapforceserver
```

Note that the uninstallation of Debian and Ubuntu packages should be done with "--purge" instead of "--remove".

3.5 macOS

3.5.1 Installer MapForce Server

Exigences du système

▼ macOS

macOS 10.15 ou plus récent

Conditions préalables

- Effectuer l'installation en tant qu'utilisateur avec des privilèges d'administrateur (root).

Intégration FlowForce Server

Vous pouvez installer MapForce Server soit en version autonome soit en tant que partie de l'installation FlowForce Server. Pour installer MapForce Server sous la gestion FlowForce Server, choisir l'option respective sur l'assistant d'installation FlowForce. Le programme d'installation FlowForce Server contient le programme d'installation MapForce Server, et le programme d'installation FlowForce Server Advanced Edition contient le programme d'installation MapForce Server Advanced Edition. Les programmes d'installation des deux produits FlowForce Server et MapForce Server sont disponibles pour téléchargement dans l'Altova Download Center (<https://www.altova.com/download>).

Si vous installez MapForce Server avec FlowForce Server, il est recommandé d'installer tout d'abord FlowForce Server. Sinon, voir les "Tâches post-installation" ci-dessous.

Installer MapForce Server

1. Télécharger le fichier de l'image du disque (.dmg) depuis le centre de téléchargement Altova (<https://www.altova.com/fr/download>) et cliquer pour l'ouvrir. Une fois cette étape effectuée, l'installateur apparaît sous la forme d'un nouveau lecteur sur votre ordinateur.
2. Double-cliquer sur le fichier de package (.pkg), et passer par les étapes successives de l'assistant d'installation. Celles-ci sont parfaitement intuitives et contiennent une étape de confirmation de l'accord de licence avant de pouvoir poursuivre. Une fois avoir terminé les étapes de l'assistant, l'icône MapForce Server devient disponible dans **Applications**. Le paquet est installé dans le répertoire **/usr/local/Altova/MapForceServer2024/**.
3. Si vous n'avez pas encore installé Altova License Server, ou si vous souhaitez le mettre à niveau de la dernière version, double-cliquer sur le fichier de paquet correspondant (.pkg) et suivez les instructions sur l'écran.

Mettre MapForce Server sous licence

1. S'assurer que la dernière version d'Altova LicenseServer (<https://www.altova.com/licenseserver>) est installée et en cours d'exécution soit sur l'appareil local ou le réseau.
2. S'enregistrer sur MapForce Server avec Altova LicenseServer. Pour ce faire, vous devez connaître soit le nom d'hôte ou l'adresse IP de l'ordinateur sur lequel LicenseServer est exécuté. Par exemple, si LicenseServer est exécuté sur 127.0.0.1, vous pouvez l'enregistrer comme suit :

```
sudo mapforceserver licenseserver 127.0.0.1
```

3. Se connecter à l'interface d'administration sur LicenseServer et attribuer la licence sur l'appareil sur laquelle MapForce Server est exécutée.

Pour plus d'informations, se référer à la documentation Altova LicenseServer (<https://www.altova.com/fr/documentation>).

Tâches post-installation

Si vous avez installé MapForce Server avant FlowForce Server, alors, après avoir installé MapForce Server et FlowForce Server, exécutez la commande suivante :

```
cp /usr/local/Altova/MapForceServer2019/etc/*.tool /usr/local/Altova/FlowForceServer2019/tools
```

Cette commande copie le fichier **.tool** depuis le répertoire **/etc** de MapForce Server vers le répertoire **/tools** de FlowForce Server. Le fichier **.tool** est exigé par FlowForce Server ; il contient le chemin vers le programme d'exécution MapForce Server. Vous n'avez pas besoin d'exécuter cette commande si vous installez FlowForce Server avant d'installer MapForce Server.

3.5.2 Désinstaller MapForce Server

Pour désinstaller **MapForce Server**:

1. Dans **Applications**, cliquer avec la touche de droite sur l'icône MapForce Server et choisir **Déplacer vers la corbeille**. Veuillez noter que votre répertoire d'installation MapForce Server et vos données de configuration ne seront pas supprimés par cette étape.
2. En option, si vous souhaitez supprimer également le répertoire d'installation de MapForce Server, exécutez la commande suivante :

```
sudo rm -rf /usr/local/Altova/MapForceServer2024/
```

Note : Les étapes ci-dessus ne suppriment pas les données de configuration de **/var/Altova/MapForceServer2024**. Notez que lorsque les données de configuration ont été supprimées il n'est pas possible de les récupérer ultérieurement.

3.6 Windows

3.6.1 Installer MapForce Server

Exigences du système

- Windows 10, Windows 11
- Windows Server 2016 ou plus récent

Conditions préalables

- Effectuer l'installation en tant qu'utilisateur avec des privilèges d'administrateur.
- À partir de la version 2021, une version 32-bit de MapForce Server ne peut pas être installée par-dessus une version 64-bit, ou une version 64-bit sur une version 32-bit. Vous devez soit (i) supprimer l'ancienne version avant d'installer la nouvelle, ou (ii) mettre à niveau une version plus récente qui est la même version de bit que votre installation plus ancienne.

Installer MapForce Server

Vous pouvez installer MapForce Server soit en version autonome soit en tant que partie de l'installation FlowForce Server. Pour installer MapForce Server sous la gestion FlowForce Server, choisir l'option respective sur l'assistant d'installation FlowForce. Le programme d'installation FlowForce Server contient le programme d'installation MapForce Server, et le programme d'installation FlowForce Server Advanced Edition contient le programme d'installation MapForce Server Advanced Edition. Les programmes d'installation des deux produits FlowForce Server et MapForce Server sont disponibles pour téléchargement dans l'Altova Download Center (<https://www.altova.com/download>).

Pour installer MapForce Server en version autonome :

1. Télécharger le fichier d'installation depuis l'Altova Download Center (<https://www.altova.com/download>) et l'exécuter.
2. En option, choisir le langage d'installation depuis la fenêtre dans la partie inférieure gauche de l'assistant et cliquer sur **Suivant**. Vous pouvez changer le langage ultérieurement depuis la ligne de commande.
3. Procéder comme suit :
 - a. Si vous n'avez pas encore installé Altova LicenseServer, ne touchez pas aux paramètres par défaut. L'Assistant installera la dernière version d'Altova LicenseServer sur l'ordinateur sur lequel vous exécutez l'assistant.
 - b. Si vous n'avez pas encore installé Altova LicenseServer et que vous souhaitez installer Altova LicenseServer sur un autre ordinateur, cliquer pour supprimer la case à cocher **Installer Altova LicenseServer sur cet appareil**, puis choisir **S'enregistrer plus tard**. Dans ce cas, vous devrez installer Altova LicenseServer et enregistrer MapForce Server séparément.
 - c. Si Altova LicenseServer a déjà été installé sur votre ordinateur mais que sa version est moins élevée que celle prescrite par l'assistant à l'installation, ne touchez pas aux paramètres par défaut. Dans ce cas, l'assistant à l'installation mettra automatiquement à niveau votre version LicenseServer avec celle indiquée dans le dialogue. Veuillez noter que l'enregistrement existant et les informations de licence seront préservées après la mise à niveau.
 - d. Si Altova LicenseServer a déjà été installé sur votre ordinateur ou sur votre réseau, et s'il a la même version que celle indiquée par l'assistant, procédez comme suit :
 - i. cliquer pour supprimer la case à cocher **Installer Altova LicenseServer sur cet appareil**.

- ii. sous **Enregistrer ce produit avec**, choisir l'instance d'Altova LicenseServer sur laquelle vous souhaitez enregistrer MapForce Server, ou choisir **S'enregistrer plus tard**. Notez que vous pouvez toujours sélectionner **S'enregistrer plus tard** si vous voulez ignorer les associations de LicenseServer et poursuivre l'installation de MapForce Server.
4. Cliquer sur **Suivant**.
 5. Suivre les instructions de l'assistant pour terminer l'installation.

Installation sur Windows Server Core

Windows Server Core est une installation Windows minimale qui n'utilise qu'un certain nombre de fonctions de la GUI. Vous pouvez installer MapForce Server sur un appareil Windows Server Core comme suit :

1. Téléchargez l'exécutable MapForce Server depuis le centre Altova Download Center (<https://www.altova.com/download>). Assurez-vous que l'exécutable corresponde à votre plate-forme de serveur (32-bit ou 64-bit).
2. Sur un ordinateur Windows avec une interface utilisateur graphique, ouvrez une fenêtre d'invite de commande en tant qu'Administrateur et exécutez la commande suivante sur l'installation exécutable :

```
MapForceServerAdv<version>.exe /u
```

3. Ceci décompresse le(s) fichier(s) .msi file(s) dont vous avez besoin pour votre prochaine étape. Copiez le fichier .msi sur l'ordinateur qui exécute Windows Server Core.
4. Si vous mettez à jour une version antérieure de MapForce Server, fermez MapForce Server avant de réaliser la prochaine étape.
5. Installez MapForce Server via le fichier .msi en utilisant la commande suivante :

```
msiexec /i MapForceServerAdvanced.msi
```

Pour tester la valeur retour de votre installation, vous voulez éventuellement exécuter un script semblable au suivant :

```
start /wait msiexec /i MapForceServerAdvanced.msi /q  
echo %errorlevel%
```

Le code retour de l'opération d'installation sera disponible dans la variable d'environnement %errorlevel% . Le code retour **0** indique un succès.

Pour une installation silencieuse avec un code retour et un log de la procédure d'installation, exécutez :

```
start /wait msiexec /i MapForceServerAdvanced.msi /q /L*v! <pathToInstallLogFile>
```

Note : gardez les fichiers .msi à un endroit sûr. Vous en aurez besoin plus tard pour désinstaller, réparer ou modifier l'installation.

Pour modifier l'installation, exécutez :

```
msiexec /m MapForceServerAdvanced.msi
```

Pour réparer l'installation, exécutez :

```
msiexec /r MapForceServerAdvanced.msi
```

Pour désinstaller MapForce Server, exécutez :

```
msiexec /x MapForceServerAdvanced.msi
```

En alternative, la commande suivante désinstallera MapForce Server en toute discrétion et fera rapport du résultat détaillé dans un fichier log :

```
start /wait msiexec /x MapForceServerAdvanced.msi /q /L*v! <pathToUninstallLogFile>
```

Pour installer des taxonomies, utilisez le Taxonomy Package Manager via la ligne de commande. Consultez le manuel MapForce Server pour des informations sur comment faire.

Mettre MapForce Server sous licence

1. Assurez-vous que la dernière version d'Altova LicenseServer (<https://www.altova.com/fr/licenseserver>) est installée et fonctionne correctement soit sur l'appareil local soit sur le réseau.
2. Si vous ne l'avez pas encore fait pendant l'installation, enregistrer MapForce Server auprès d'Altova LicenseServer. Pour ce faire, vous devez connaître soit le nom d'hôte ou l'adresse IP de l'ordinateur sur lequel LicenseServer est exécuté. Par exemple, si LicenseServer est exécuté sur 127.0.0.1, vous pouvez l'enregistrer comme suit :

```
MapForceServer.exe licenseserver 127.0.0.1
```

3. Connectez-vous sur l'interface d'administration de LicenseServer et attribuez-y la licence de l'appareil sur lequel MapForce Server est exécuté.

Pour plus d'informations, veuillez vous référer à la documentation Altova LicenseServer (<https://www.altova.com/fr/documentation>).

3.6.2 Désinstaller MapForce Server

Désinstaller MapForce Server

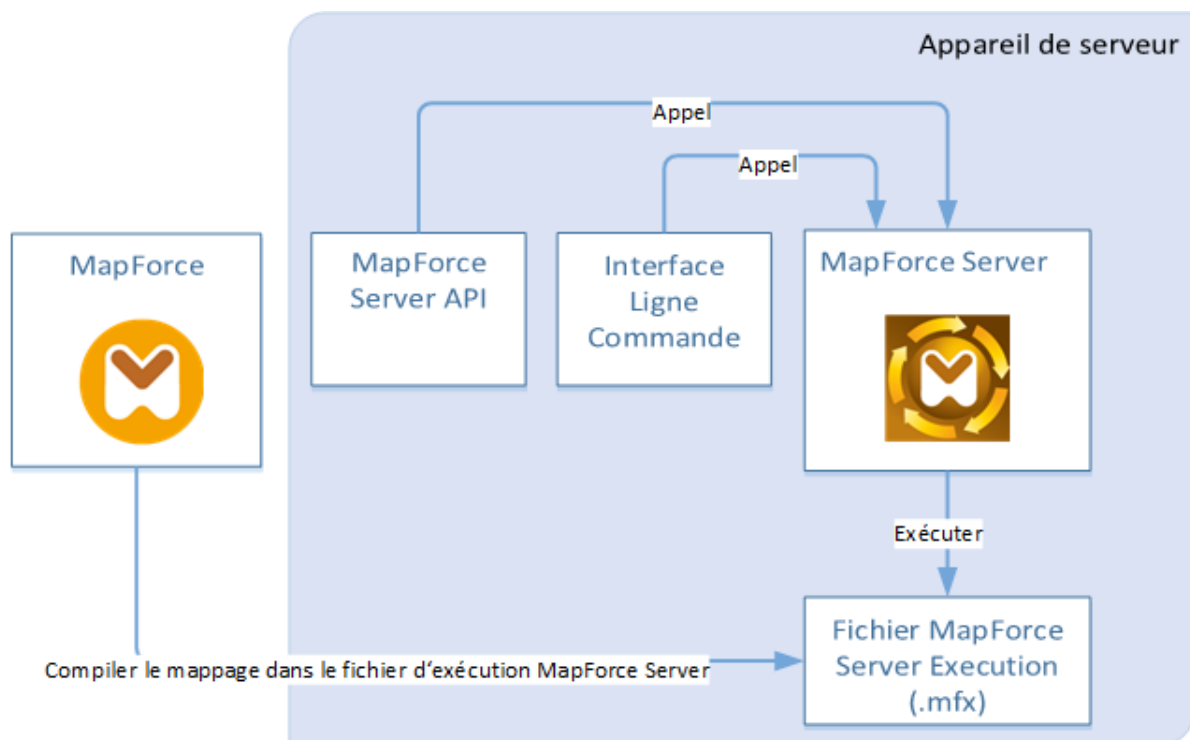
1. Cliquer avec la touche de droite sur la touche "Start" de Windows et choisir **Paramètres**.
2. Ouvrir le Panneau de configuration (commencer à saisir "Panneau de configuration " et cliquer sur l'entrée suggérée).
3. Sous **Programmes**, cliquer sur **Désinstaller un programme**.
4. Dans le Panneau de configuration, choisir "Altova MapForce Server" puis cliquer sur **Désinstaller**.

4 Exécuter les mappages

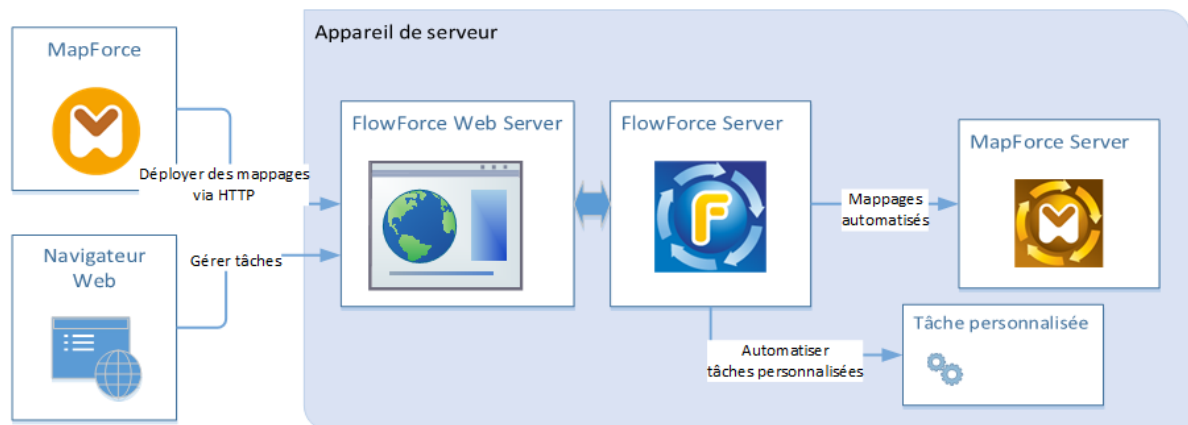
Le rôle de MapForce Server est d'exécuter les mappages de données créés avec Altova MapForce (<http://www.altova.com/mapforce.html>).

Tout d'abord, vous créez visuellement les mappages de données (ou les Mapping Design Files, *.mfd) dans MapForce, où vous définissez les entrées, les sorties et toute étapes intermédiaire de traitement qui doivent être appliquées à vos données (y compris trier, filtrer, personnaliser, etc.). Une fois que votre mappage est prêt, vous pouvez l'exécuter avec MapForce Server d'une des manières suivantes :

- Sur l'appareil Windows sur lequel MapForce est exécuté, compiler le mappage à un MapForce Server Execution File (.mfx). Les fichiers .mfx sont en fait des mappages de données emballés pour une exécution dans un environnement de serveur. Vous pouvez copier ces fichiers sur un des systèmes d'exploitation pris en charge où MapForce Server est exécuté (y compris sur les différentes plateformes, voir [Exigences du système](#)³⁰). Sur l'appareil de serveur, vous pouvez exécuter le fichier .mfx à l'aide de l'interface de ligne de commande proposé par MapForce Server, ou en utilisant l'API MapForce Server.



- Sur l'appareil Windows sur lequel MapForce est exécuté, déployer le mappage vers un appareil à serveur où MapForce Server et FlowForce Server sont installés. L'appareil du serveur peut être un système d'exploitation différent (voir [Exigences du système](#)³⁰). Les mappages déployés de cette manière deviennent des fonctions FlowForce Server et ils vous permettent de créer des tâches programmées ou basées sur des déclencheurs. Lorsque des mappages sont exécutés en tant que tâches FlowForce Server, ils peuvent aussi être exposés en tant que services Web, enchaînés en tant que sous-étapes d'autres tâches, ou deviennent une partie des flux de travail, ce qui inclut l'envoi d'e-mails, la vérification de codes de sortie, l'exécution de commandes shell, etc.




Pour plus d'informations concernant ce scénario, voir la documentation FlowForce Server (<http://manual.altova.com/fr/FlowForceServer>).

Comment exécuter des mappages compilés en tant que fichiers MapForce Server Execution

1. Lancer MapForce Enterprise ou Professional Edition.
2. Ouvrir le mappage à compiler.
3. Dans le menu **Fichier**, cliquer sur **Compiler sur fichier d'exécution MapForce**, et choisir un répertoire de destination.
4. Copier le fichier .mfx sur le répertoire ou le serveur de destination, avec des fichiers d'entrée ou de dépendances. Pour plus d'information, voir [Préparer des mappages pour l'exécution de serveur](#) ⁴³
5. Appeler la commande "run" dans l'interface de ligne de commande (voir Utilisation de la ligne de commande), ou la méthode équivalente de l'API MapForce Server (voir [MapForce Server API](#) ⁸⁷).

Comment exécuter des mappages déployés sur FlowForce Server

1. Ouvrir dans MapForce Enterprise ou Professional le mappage que vous souhaitez déployer.
2. S'assurer que le langage de transformation (moteur d'exécution) du mappage est défini sur Intégré. Pour faire passer le moteur d'exécution à Intégré, choisir la commande de menu **Sortie | Moteur d'exécution intégré**, ou cliquer sur la touche de barre d'outils **Sélectionner le moteur d'exécution intégré** ().
3. Dans le menu **Fichier**, cliquer sur **Déployer sur FlowForce Server**.
4. Saisir les détails de connexion de serveur (hôte, port), les identifiants FlowForce, et le conteneur de destination FlowForce. Pour continuer avec la création de la tâche FlowForce immédiatement dans le navigateur, choisir l'option **Ouvrir navigateur web pour créer une nouvelle tâche**. Vous pouvez aussi créer la tâche FlowForce ultérieurement (voir étape suivante).
5. Ouvrir un navigateur, s'inscrire sur l'interface d'administration Web de FlowForce Server, et naviguer sur le conteneur où vous avez déployé le mappage (voir étape précédente). Cette étape n'est pas nécessaire si vous avez choisi l'option **Ouvrir navigateur web pour créer une nouvelle tâche** dans l'étape précédente.
6. Définir la tâche FlowForce Server, y compris ses déclencheurs, paramètres ou étapes d'exécution supplémentaires (pour des exemples, consulter la documentation FlowForce Server <https://www.altova.com/fr/documentation>). Une fois que la tâche est configurée pour être exécutée, la transformation de mappage sous-jacente sera exécutée et la sortie de mappage sera produite.

Note : Si MapForce Server est exécuté sur un autre appareil que celui sur lequel le mappage a été conçu, veuillez vous assurer que les chemins sont ajustés aux fichiers d'entrée ou aux détails de connexion de base de données de manière à ce qu'ils soient pertinents dans le nouvel environnement d'exécution cible. Par exemple, si un mappage appelle une base de données et nécessite un pilote de base de données, le pilote doit aussi être installé dans l'environnement cible pour que le mappage puisse être exécuté avec succès. Pour consulter ou ajuster les détails de connexion à la base de données, cliquer avec la touche de droite sur le composant de base de données dans MapForce et sélectionner **Propriétés**. Une fois avoir effectué des changements au design de mappage dans MapForce, pensez à le recompiler dans un fichier d'exécution MapForce Server (.mfx) ou, selon les cas, le redéployer sur FlowForce Server. Pour plus d'informations, voir [Préparer les mappages pour l'exécution de serveur](#)⁴³.

4.1 Préparer les 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 le mappage a été conçu. C'est pour cette raison que les mappages 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.

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).
- 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 :

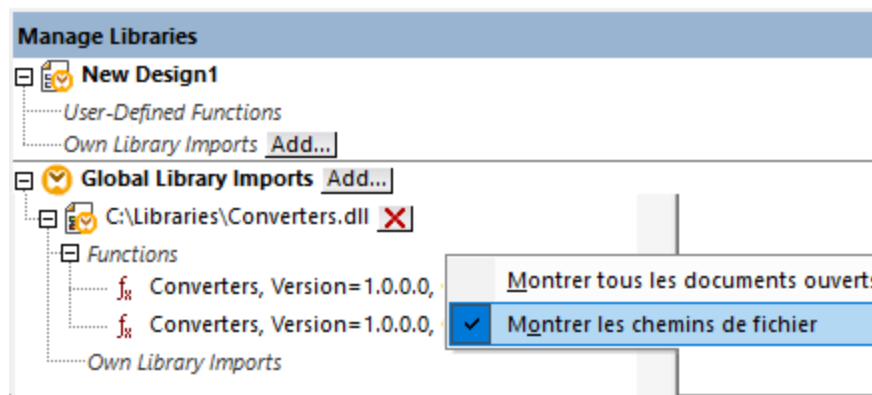
- Sur l'appareil source, l'édition MapForce Enterprise ou Professional est exigée pour concevoir le mappage et le déployer sur un appareil cible.
- MapForce Server et FlowForce Server doivent tous deux être sous licence sur l'appareil cible. Le rôle de MapForce Server est d'exécuter le mappage ; le rôle 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.
- 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 (<https://www.microsoft.com/en-us/download/details.aspx?id=50040>) 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.



- Certains mappages lisent plusieurs fichiers d'entrée en utilisant un chemin de caractère générique. 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 .
- 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 [Gestion de certificat numérique](#) ¹⁴⁸ .
- 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, veuillez consulter la documentation MapForce .

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. 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 .
- 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 .

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. 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.

2. Définir les variables d'environnement à l'emplacement sur lequel le pilote JDBC est installé. Généralement, vous devez 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 devez 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/instantclient_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	\$PATH:/opt/oracle/instantclient_11_2

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**.
2. 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 . Un moyen simple d'éviter les problèmes 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).
3. 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. Pour consulter un exemple, voir le sujet de documentation FlowForce "Exposer une tâche en tant que service Web" (https://www.altova.com/manual/fr/flowforceserveradvanced/2024.2/index.html?fs_example_web_service.htm).
- 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.
2. 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.

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 [Ressources globales](#) ⁴⁹.

Packs de taxonomie XBRL

Lorsque vous déployez un mappage qui référence des Packs de taxonomie XBRL sur FlowForce Server, MapForce collectionne toutes les références externes depuis le mappage et puis les résout en utilisant la configuration actuelle et les packs de taxonomie installés actuellement. S'il y a des références externes résolues qui pointent vers un pack de taxonomie, le pack de taxonomie est déployé avec le mappage. FlowForce Server utilisera ce pack (tel qu'il était pendant le déploiement) pour exécuter le mappage. Afin de réinitialiser le pack de taxonomie utilisé par FlowForce Server, vous devrez le modifier dans MapForce et de redéployer le mappage.

Veillez noter que le catalogue root de MapForce Server influe sur la manière dont les taxonomies sont résolues sur l'appareil cible. Le catalogue root est trouvé sous le chemin relatif suivant dans le répertoire d'installation MapForce Server : **etc/RootCatalog.xml**.

Les packs de taxonomie qui ont été déployés avec un mappage sera utilisé si le catalogue root de MapForce Server ne contient pas déjà un tel pack qui est défini pour le même préfixe URL. Le catalogue root de MapForce Server a une priorité sur la taxonomie déployée.

Si MapForce Server est exécuté en autonomie (sans FlowForce Server), il est possible de spécifier le catalogue root qui devrait être utilisé par le mappage comme suit :

- Au niveau de la ligne de commande, il est possible en ajoutant l'option `-catalog` à la commande `run`.
- Dans l'API MapForce Server, appeler la méthode `SetOption`, et fournir la chaîne "catalog" en tant que premier argument, et le chemin vers le catalogue root en tant que second argument.

Si un mappage utilise des composants XBRL avec des bases de lien de table, le pack de taxonomie ou le fichier de configuration du pack de taxonomie doit être fourni au mappage lors de l'exécution comme suit :

- Au niveau de la ligne de commande MapForce Server, ajouter l'option `--taxonomy-package` ou `--taxonomy-packages-config-file` à la commande `run`.
- Dans l'API MapForce Server, appeler la méthode `SetOption`. Le premier argument doit être soit "taxonomy-package" ou "taxonomy-packages-config-file". Le second argument doit être le chemin actuel vers le pack de taxonomie (ou le fichier de configuration de taxonomie).

4.2 Ressources Globales

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 Server, vous pouvez ensuite aisément extraire des données soit de la base de données de `développement` ou de `production`, en utilisant la configuration désirée en tant que paramètre de ligne de commande lors de l'exécution du mappage.

Les Ressources globales peuvent être utilisées dans toutes les différentes applications d'Altova (*voir la sous-section 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](#)
- [MapForce](#)
- [StyleVision](#)
- [XMLSpy](#)
- [FlowForce Server](#)
- [MapForce Server](#)
- [RaptorXML Server et RaptorXML+XBRL Server.](#)

Pour plus d'informations concernant la création de Ressources globales, se référer au chapitre "Ressources Globales Altova" de la documentation MapForce.

Ressources 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 :

1. *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.
2. *Le nom de configuration de la Ressource globale.* Chaque Ressource globale a une configuration par défaut. Vous pouvez aussi créer des configurations supplémentaires.

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\MapForceServer2024\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 l'[API de MapForce Server](#) ⁸⁷.

4.3 Optimisation Join

L'optimisation Join accélère l'exécution des mappages de données dans lesquels de grands ensembles de données sont filtrés ou joints.

L'optimisation Join fonctionne en éliminant les boucles imbriquées qui se produisent de manière interne lors de l'exécution d'un mappage. Une boucle imbriquée se produit lorsque le mappage itère chaque item d'un ensemble autant de fois qu'il y a des items dans un second ensemble. Veuillez noter qu'il est normal pour le moteur d'exécution* de mappage d'effectuer des boucles (itérations) sur plusieurs séquences d'items, du fait de sa conception. Lorsque des boucles indépendantes imbriquées se produisent (c'est à dire, des boucles qui itèrent sur d'autres boucles), le mappage peut bénéficier de l'optimisation join, qui pourrait réduire considérablement le temps requis pour exécuter le mappage. Les boucles imbriquées sont à peine visibles lorsque vous exécutez les mappages où les données d'entrée ne sont pas significativement plus grandes ; néanmoins, cela peut devenir un défi en cas de mappages qui traitent les fichiers ou les bases de données qui consistent en un très grand nombre d'enregistrements.

* Le moteur d'exécution d'un mappage peut être MapForce, MapForce Server, ou un programme C#, C++, ou Java généré par MapForce. L'optimisation Join est disponible exclusivement dans MapForce Server Advanced Edition.

Pour concevoir MapForce Server en tant qu'un moteur d'exécution cible, cliquer sur la touche BUILT-IN () dans la barre d'outils de MapForce. Cela permettra de vous assurer que votre mappage bénéficie de la plupart des fonctions disponibles. Si vous choisissez un autre langage de transformation, certaines fonctions MapForce peuvent ne pas être prises en charge dans ce langage.

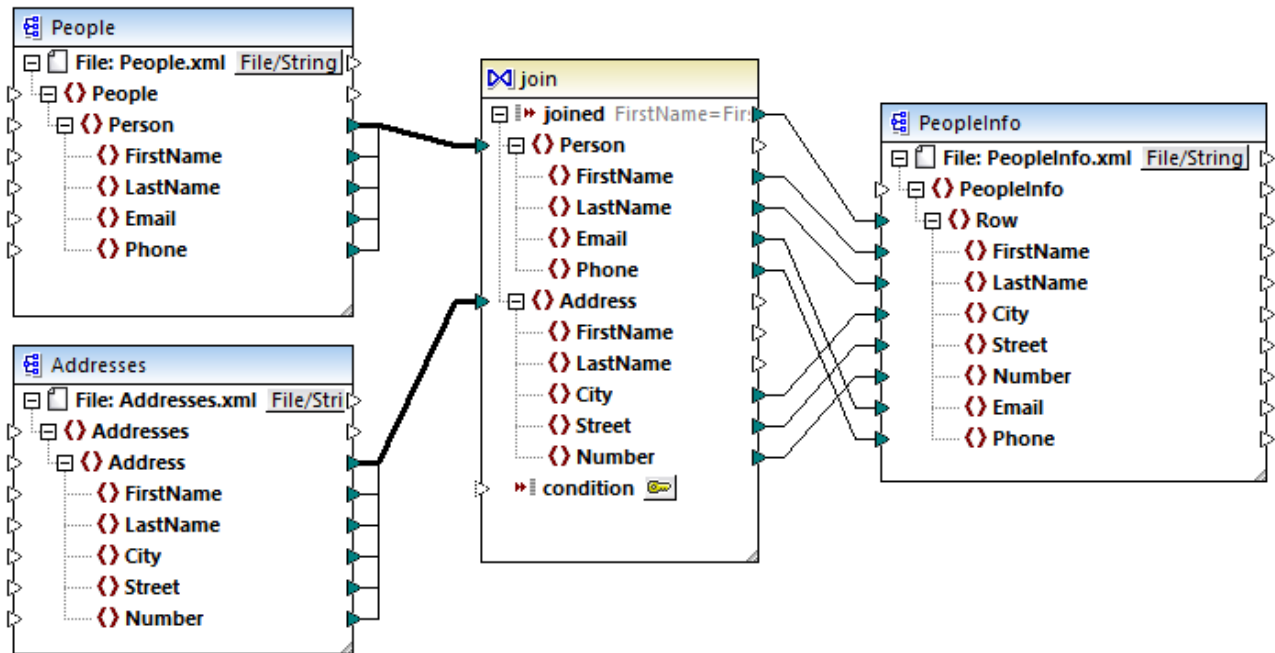
Comme indiqué ci-dessus, le but principal de l'optimisation join est d'adresser des boucles imbriquées de manière efficace. À présent, observons comment les boucles imbriquées se produisent.

Habituellement, des boucles imbriquées se produisent lorsque le mappage contient au moins un composant Join, et que le mode SQL JOIN** n'est pas possible.

** Lorsque certaines conditions sont remplies dans MapForce, les mappages peuvent permettre un mode d'exécution spécial appelé "SQL Join mode" (cela est uniquement applicable si le mappage lit les données provenant d'une base de données). Lorsque les données sont jointes de cette manière, l'opération join est effectuée par la base de données (c'est à dire qu'un SQL JOIN se produit), et cela élimine le besoin des boucles imbriquées dans le moteur d'exécution du mappage. Pour plus d'informations concernant le SQL Join, veuillez vous référer à la documentation MapForce (<https://www.altova.com/fr/documentation.html>).

Par exemple, l'image ci-dessous montre un mappage (conçu avec Altova MapForce) qui combine des données provenant de deux fichiers XML utilisant un composant Join. Sur l'ordinateur sur lequel MapForce est installé, ce mappage est disponible sous le chemin suivant : ..

`\\Documents\Altova\MapForce2024\MapForceExamples\Tutorial\JoinPeopleInfo.mfd`. Certaines données concernant les personnes sont uniquement disponibles dans le premier fichier XML (**Email**, **Phone**), alors que d'autres données sont disponibles uniquement dans le second fichier XML (**City**, **Street**, **Number**). Le but du mappage est d'écrire dans le fichier XML cible les données fusionnées de toutes les personnes à savoir que **FirstName** et **LastName** correspondent dans les deux structures de source.



JoinPeopleInfo.mfd

Dans MapForce, un composant join jumelle des items dans deux ensembles conformément à des conditions personnalisées, qui implique la comparaison de chaque item dans l'ensemble 1 avec chaque item dans l'ensemble 2. Le nombre total des comparaisons représente la jointure croisée (produit cartésien) des deux ensembles. Par exemple, si le premier ensemble contient 50 items, et si le second ensemble contient 100 items, un total de 5000 (50 x 100) comparaisons se produiront. Dans le mappage ci-dessus, les ensembles qui sont comparés correspondent à tous les items d'instance des deux structures XML connectées au composant Join.

Note : Attention, ne pas confondre l'optimisation join (une fonction de MapForce Server Advanced Edition) avec les composants Join (une fonction de MapForce). Pour plus d'information concernant les composants Join, veuillez vous référer à la documentation MapForce (<https://www.altova.com/fr/documentation.html>).

Comme on peut s'y attendre, d'un point de vue de la performance, les mappages qui contiennent des boucles imbriquées nécessitent plus de temps pour être exécutés. Imaginez une situation où les deux ensembles joins contiennent des millions d'enregistrements. Cela peut facilement affecter la performance, et c'est là que l'optimisation join est utile. Très généralement, l'optimisation join se comporte comme un moteur de base de données qui est optimisé à consulter (index) de très grands ensembles de données. À part cela, comme illustré par le mappage ci-dessus, l'optimisation join ne traite pas uniquement avec des données provenant des bases de données. L'optimisation join élimine les boucles imbriquées quel que soit le type de données, en générant, lorsque cela est possible, des tables de consultation internes qui sont requises lors de l'exécution du mappage. Cela améliore de manière considérable la performance de mappage et finalement de réduire le temps requis pour exécuter le mappage.

Note : Lorsque l'optimisation join se produit, l'exécution du mappage prend moins de temps mais nécessite généralement plus de mémoire. Sachez que les profils d'utilisation de la mémoire dépendent de plusieurs facteurs complexes ; c'est pourquoi le comportement observé peut différer selon le cas.

L'optimisation join peut accélérer non seulement des mappages avec des joins, mais aussi ceux qui utilisent des composants de filtre. Dans MapForce, un filtre traite une séquence d'items (c'est à dire, il contrôle une condition booléenne donnée pour chaque instance de l'item connecté à l'entrée **nœud/ligne**). Si la condition booléenne est connectée à une fonction qui, à son tour, doit itérer sur une autre séquence d'items et si le contexte de mappage l'exige, une situation similaire à un join se produit. Si le filtre doit effectuer une comparaison croisée de chaque item dans deux ensembles, il se qualifie pour l'optimisation join.

Pour que le mappage puisse bénéficier d'une optimisation join, il doit être exécuté par MapForce Server Advanced Edition. Pour exécuter un mappage avec MapForce Server Advanced Edition, l'ouvrir dans MapForce, et le compiler dans un fichier d'exécution de mappage (.mfx) en utilisant la commande de menu **Fichier | Compiler vers Fichier d'exécution MapForce Server**. Ensuite exécuter le fichier .mfx en utilisant une méthode API dans le langage de votre choix ou bien la commande `run` de l'interface de ligne de commande (voir aussi [Le principe de base](#)⁴⁰).

4.4 Identifiants

Les objets d'identifiant permettent de rendre les données d'authentification (comme des noms d'utilisateur, des mots de passe et des détails d'authentification OAuth) portables dans plusieurs environnements d'exécution de mappage, d'une manière sûre. Les identifiants sont utiles dans des mappages qui exigent une authentification HTTP de base ou une autorisation OAuth 2.0. Vous pouvez définir des identifiants dans MapForce et aussi dans FlowForce Server. Si des identifiants ont été définis dans MapForce, vous pouvez les déployer en option dans FlowForce Server, comme pour le déploiement des mappages.

Une fois avoir compilé le mappage dans un fichier d'exécution MapForce Server (.mfx), MapForce Server exécutera le fichier .mfx selon vos choix effectués au moment de la conception du mappage.

Si vous avez coché la case **Inclure dans le fichier d'exécution MapForce Server et Déploiement du mappage** lors de la création de l'identifiant dans MapForce, MapForce Server utilisera le moment d'exécution tout identifiant qui a été stocké dans le fichier .mfx. Cela signifie que vous pouvez exécuter le mappage avec une commande comme :

```
<exec> run mapping.mfx
```

Où `<exec>` est le chemin de l'exécutable MapForce Server. Ce chemin peut être soit absolu soit, si le répertoire actuel est le même que l'exécutable, vous pouvez saisir uniquement le nom de l'exécutable.

Si vous avez saisi uniquement le nom de l'identifiant (sans le nom de l'utilisateur et le mot de passe) dans MapForce, vous devez fournir explicitement ces détails au moment de l'exécution du mappage, avec l'aide de l'option de ligne de commande `--credential` disponible pour la commande `run`. Ainsi, vous pouvez utiliser, par exemple, un ensemble d'identifiants différent dans la production, contrairement à ceux utilisés lors de la conception du mappage. L'option `--credential` a la forme `--credential=KEY:VALUE.`, où

- `KEY` est le nom de l'identifiant tel qu'il a été défini dans MapForce.
- `VALUE` est une propriété d'identifiant, ou une liste de propriétés séparées par une esperluette (&). En ce qui concerne les identifiants de type "mot de passe", les propriétés possibles sont `username` et `password`. En ce qui concerne les identifiants de type OAuth 2.0, la seule propriété prise en charge est `oauth:token`.
- Les valeurs de propriété réelles sont fournies comme des paramètres de requête dans une URL, en utilisant le signe "=".

Par exemple :

```
<exec> run mapping.mfx --credential="mycredential:username=admin&password=4xJ38dnx7"
```

Dans l'extrait de code ci-dessus, la valeur de l'option `--credential` est entourée de guillemets pour traiter la valeur littéralement, puisque le nom de l'utilisateur et le mot de passe sont séparés par une esperluette.

Si votre mappage nécessite plusieurs ensembles d'identifiants nommés pour pouvoir être exécutés, vous pouvez spécifier l'option `--credential` plusieurs fois.

Les identifiants fournis en tant qu'options de ligne de commande prennent précedence sur les identifiants stockés.

Si vous n'avez pas coché la case **Inclure dans le fichier d'exécution MapForce Server et Déploiement du mappage**, les champs sensibles manquent. Cela signifie que vous devez fournir le mot de passe dans la ligne de commande tout en référant encore l'identifiant par son nom, par exemple :

```
<exec> run mapping.mfx --credential=mycredential:password=4xJ38dnx7
```

Les champs suivants sont considérés être des données sensibles :

- **Mot de passe** (pour les identifiants de type "Mot de passe")
- **Client Secret, Jeton d'accès, et Réinitialiser jeton** (pour des identifiants de type "OAuth 2.0")

En ce qui concerne des mappages qui nécessitent une autorisation OAuth 2.0, la ligne de commande MapForce Server accepte un jeton d'accès OAuth 2.0 en tant qu'entrée au moment de l'exécution du mappage. Veuillez noter que la ligne de commande de MapForce Server ne fournit pas une GUI interactive par design, vous devrez donc obtenir le jeton d'accès OAuth 2.0 par des moyens externes (par exemple, en le demandant avec MapForce) lorsque vous utilisez la ligne de commande spécifiquement. Néanmoins, cela n'est pas nécessaire si MapForce Server est exécuté sous la gestion FlowForce Server, puisque ce dernier est capable d'acquérir de manière autonome un nouveau jeton d'accès OAuth 2.0 au moment de la marche.

Dans la ligne de commande, l'exécution du mappage avec des identifiants stocké est possible tant que le jeton d'accès OAuth 2.0 stocké n'a pas expiré ou n'a pas été révoqué par le fournisseur du service Web. Ainsi, fournir un nouveau jeton d'accès OAuth 2.0 (obtenu par des moyens externes) en utilisant l'option `--credential`, par exemple :

```
<exec> run mapping.mfx --  
credential=my_oauth_credential:oauth:token=jdsaf1kajlkewsaiurthczv904215-jhd
```

Où :

- `my_oauth_credential` est le nom de l'identifiant OAuth 2.0 créé depuis MapForce.
- `oauth:token` est le moyen d'indiquer à MapForce Server qu'un nouveau jeton d'accès OAuth 2.0 est fourni au moment de la marche.

MapForce Server API

Le MapForce Server API propose des méthodes pour créer des identifiants, ajouter des propriétés dans les identifiants, et fermer les identifiants après avoir fini de les déclarer. L'extrait de code suivant illustre la manière typique de déclarer des identifiants de mot de passe dans un programme C# qui exécute un mappage :

```
//Create a MapForce Server object  
Altova.MapForceServer.Server objMFS = new Altova.MapForceServer.Server();  
// Set the credential name as it was defined in MapForce  
objMFS.BeginCredential("mycredential");  
// Add the credential properties  
objMFS.AddCredentialProperty("username", "altova");  
objMFS.AddCredentialProperty("password", "b45ax78!");  
// Close the credential  
objMFS.EndCredential();
```

Pour effectuer des autorisations OAuth 2.0 depuis un programme qui exécute un mappage, le nom de propriété d'identifiant doit être défini sur `oauth:token`, comme illustré ci-dessous :

```
//Create a MapForce Server object
Altova.MapForceServer.Server objMFS = new Altova.MapForceServer.Server();
// Set the credential name as it was defined in MapForce
objMFS.BeginCredential("my_oauth_credential");
// Add the credential properties
objMFS.AddCredentialProperty("oauth:token", "jdsaf1kajlkewsaiurthczv904215-jhd");
// Close the credential
objMFS.EndCredential();
```

Si le mappage nécessite plusieurs ensembles d'identifiants, utiliser les méthodes ci-dessus pour ajouter autant d'ensembles que d'identifiants sont nécessaires. Une fois que vous avez déclaré tous les identifiants requis, vous pouvez exécuter le fichier d'exécution de mappage d'une manière standard, en appelant la méthode `Run()`. Pour plus d'informations, voir la [Référence API](#) ⁸⁷.

4.4.1 Example: OAuth 2.0 Authorization

Cet exemple vous montre comment appeler un service Web REST-style qui requiert une autorisation OAuth 2.0. L'application client est un fichier d'exécution de MapForce Server (.mfx) qui récupérera des événements de calendrier utilisant l'API Calendrier de Google (<https://developers.google.com/calendar/>). Pour être simple, la tâche liée au du mappage récupérera l'information de calendrier « as is » et produira juste un résultat JSON brut sans autre traitement.

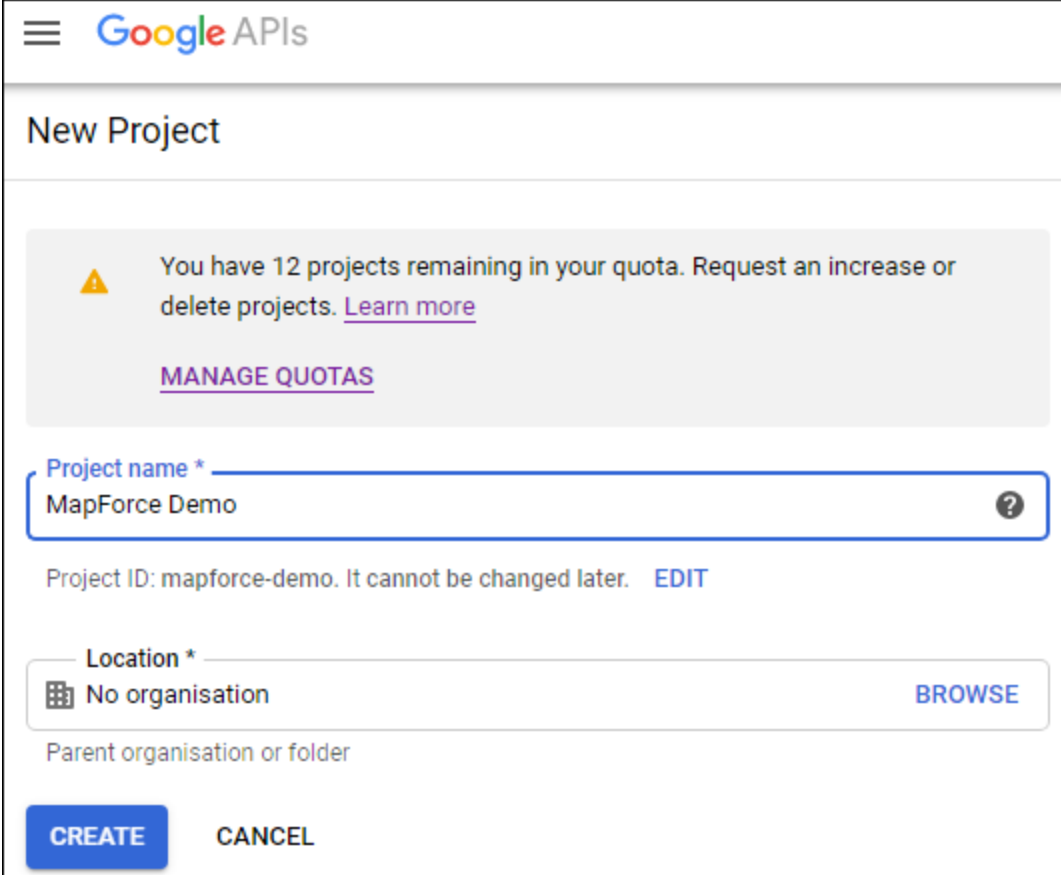
Exigences préalables :

- MapForce Enterprise Edition
- MapForce Server Advanced Edition
- Pour suivre cet exemple étape par étape, vous devez avoir un compte Google. Si vous voulez appeler un autre service Web, demandez des identifiants OAuth 2.0 de votre fournisseur de service Web et utilisez-les à la place dans les instructions ci-dessous.

Obtenir des identifiants OAuth 2.0

Si vous avez déjà des identifiants OAuth 2.0 requis pour accéder au service Web, vous pouvez ignorer cette étape. Autrement, les instructions exactes pour les obtenir dépendent du fournisseur du service Web que votre mappage appellera. Pour appeler l'API Calendrier de Google comme dans l'exemple, suivez les étapes suivantes :

1. Connexion à la Console d'API de Google (<https://console.developers.google.com/>).
2. Créer un nouveau projet.



Google APIs

New Project

⚠ You have 12 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project name * ?

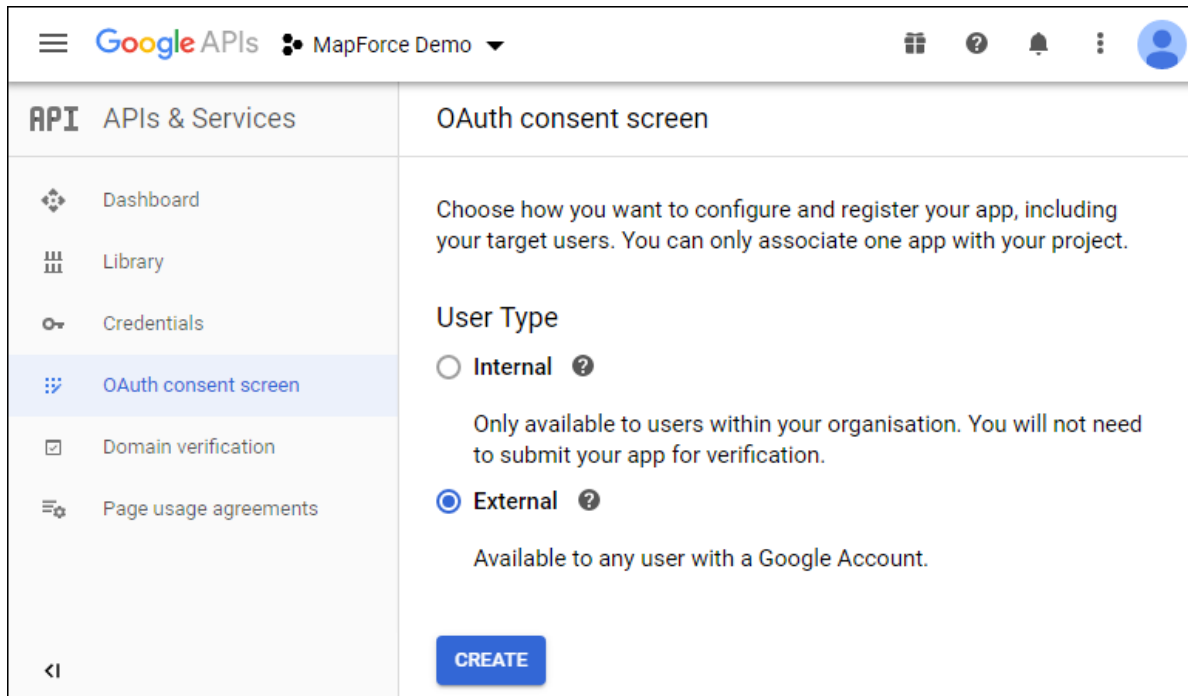
Project ID: mapforce-demo. It cannot be changed later. [EDIT](#)

Location * [BROWSE](#)

Parent organisation or folder

[CREATE](#) [CANCEL](#)

3. Cliquer sur **OAuth consent screen**.
4. Sélectionnez **Externe** en tant que type d'utilisateur, sauf si vous avez un compte G Suite qui vous permet d'allouer un accès API uniquement à des utilisateurs au sein de votre organisation.



5. Saisissez « mapforce-demo » en tant que nom d'application et enregistrez les paramètres.

OAuth consent screen

Before your users authenticate, this consent screen will allow them to choose whether they want to grant access to their private data, as well as give them a link to your terms of service and privacy policy. This page configures the consent screen for all applications in this project.

Verification status
Not published

Application name ?
The name of the app asking for consent

Application logo ?
An image on the consent screen that will help users recognise your app

6. Cliquez sur **Créer identifiants**, puis sélectionnez **OAuth Client ID**.
7. Saisissez **Desktop app** en tant que type d'application et « MapForce Client » en tant que nom client.

← Create OAuth client ID

A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms, each will need its own client ID. See [Setting up OAuth 2.0](#) for more information.

Application type *
Desktop app

[Learn more](#) about OAuth client types

Name *
MapForce Client

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

CREATE CANCEL

8. Cliquez sur **Créer**. L'ID client est créé et devient disponible sur la page **Identifiants**.

Google APIs MapForce Demo

APIs & Services Credentials + CREATE CREDENTIALS DELETE


Dashboard
Library
Credentials
OAuth consent screen
Domain verification
Page usage agreements

OAuth 2.0 Client IDs

Type	Client ID				
Desktop	-s14o...	Copy	Edit	Delete	Download

Service Accounts [Manage service accounts](#)

No service accounts to display

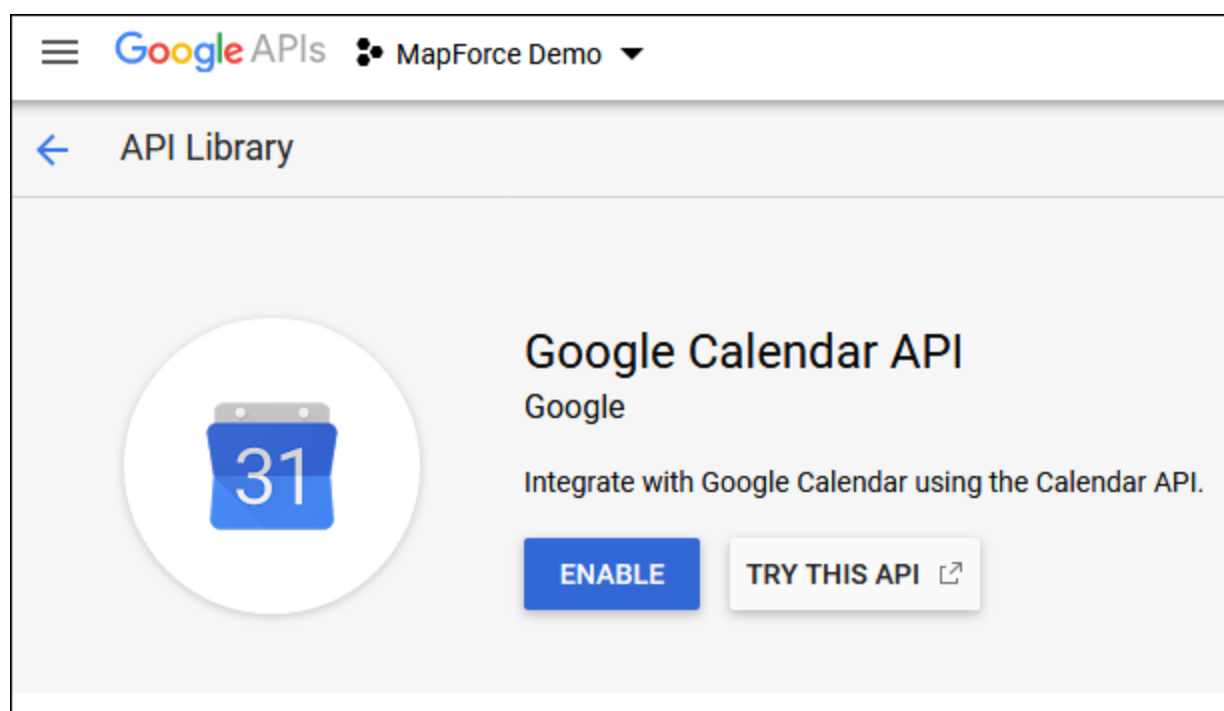
9. Cliquez sur  pour télécharger les détails d'autorisation pour OAuth 2.0 en tant que fichier JSON.

Vous avez désormais obtenu les détails d'autorisation OAuth 2.0 depuis la Console d'API de Google, notamment :

1. Point de terminaison d'autorisation
2. Point de terminaison de jeton
3. ID Client
4. Secret client

Activer l'API Calendrier de Google

Pour accepter des appels de clients, l'API Calendrier de Google utilisée dans cet exemple doit être acceptée. Dans la Console d'API de Google, cliquez sur **Library**, recherchez l'API Calendrier de Google et activez-la :



Dans cet exemple, nous allons appeler la méthode de **liste** de l'entité **Événements**. Vous pouvez trouver une référence détaillée pour cette méthode d'API sur <https://developers.google.com/calendar/v3/reference/events/list>. Pour l'instant, veuillez noter les points importants suivants :

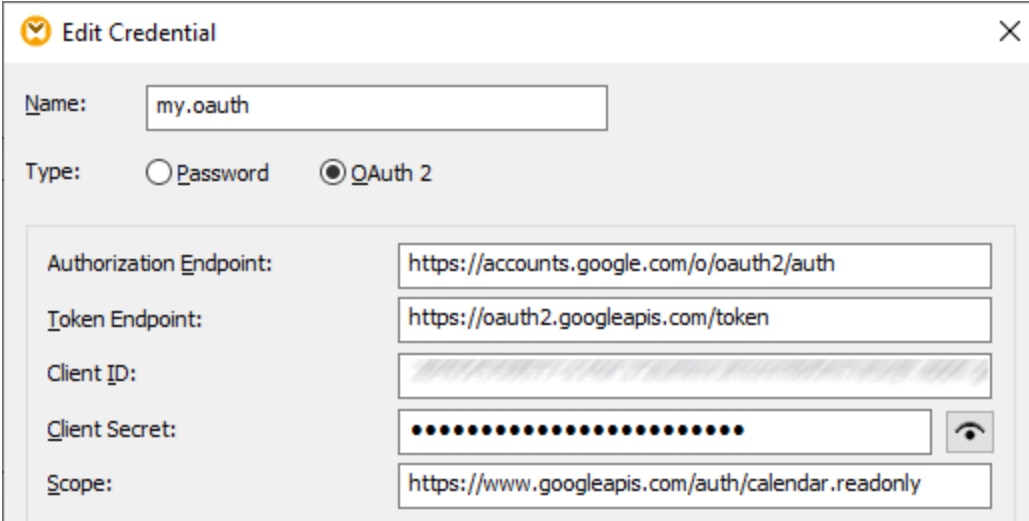
1. Tel que mentionné dans la documentation, la méthode doit être appelée en envoyant une requête GET à `https://www.googleapis.com/calendar/v3/calendars/calendarId/events`, où **calendarId** est l'identifiant pour un calendrier Google. Le paramètre de requête **calendarId** sera configuré depuis MapForce lors d'une étape suivante.
2. Appeler une méthode d'API requiert au moins l'une des deux étendues suivantes :
 - `https://www.googleapis.com/auth/calendar.readonly`
 - `https://www.googleapis.com/auth/calendar`
 - `https://www.googleapis.com/auth/calendar.events.readonly`
 - `https://www.googleapis.com/auth/calendar.events`

Au cours de la procédure d'autorisation pour OAuth 2, votre mappage devra fournir l'une des étendues ci-dessus —ceci sera également configuré à une étape suivante. À des fins d'exemple, la première étendue « read-only » sera suffisante.

Demander un jeton d'autorisation

Pour prévisualiser le mappage dans MapForce, vous devez ajouter les détails d'autorisation pour OAuth 2.0 au mappage et demander un jeton d'autorisation, tel qu'illustré ci-dessous.

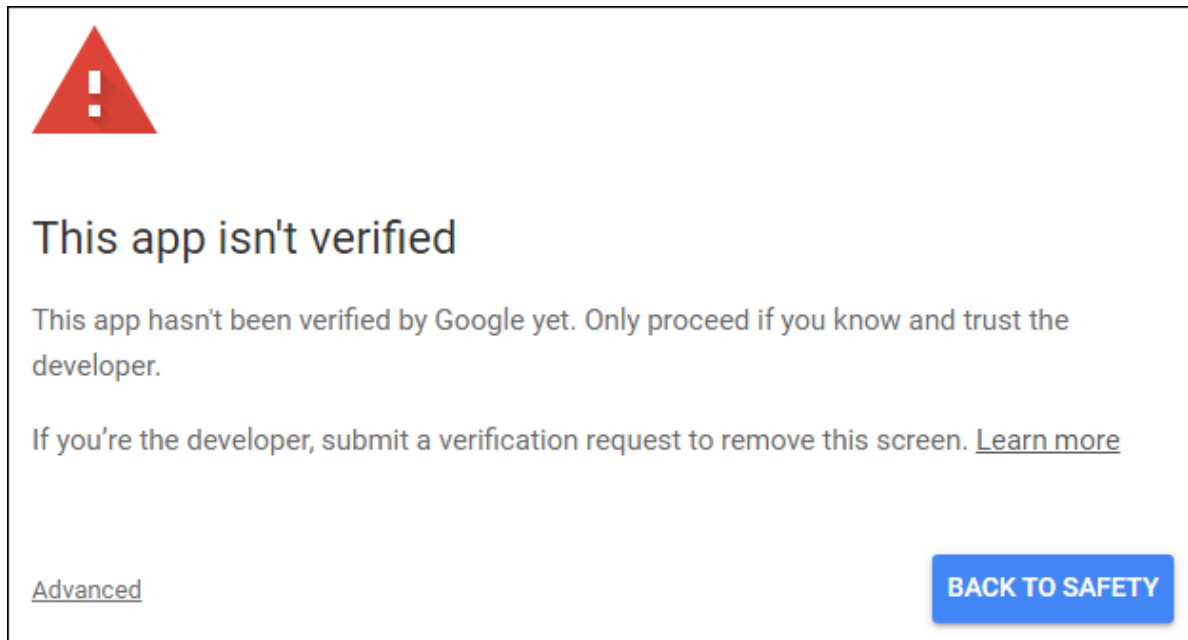
1. Dans MapForce, cliquez avec la touche de droite sur une zone vide du mappage, et sélectionnez **Ouvrir Manager d'identifiants** depuis le menu contextuel.
2. Cliquez sur **+ Ajouter identifiant**.
3. Saisissez un nom (« my.oauth », dans cet exemple), et sélectionnez **OAuth 2** comme type.
4. Remplissez les zones de texte **Point de terminaison d'autorisation**, **Point de terminaison jeton**, **ID client**, **Secret client** avec les valeurs correspondantes depuis le fichier JSON téléchargé précédemment.
5. Saisissez `https://www.googleapis.com/auth/calendar.readonly` dans la zone de texte **étendue**.
6. Laissez tous les autres paramètres tel quel.



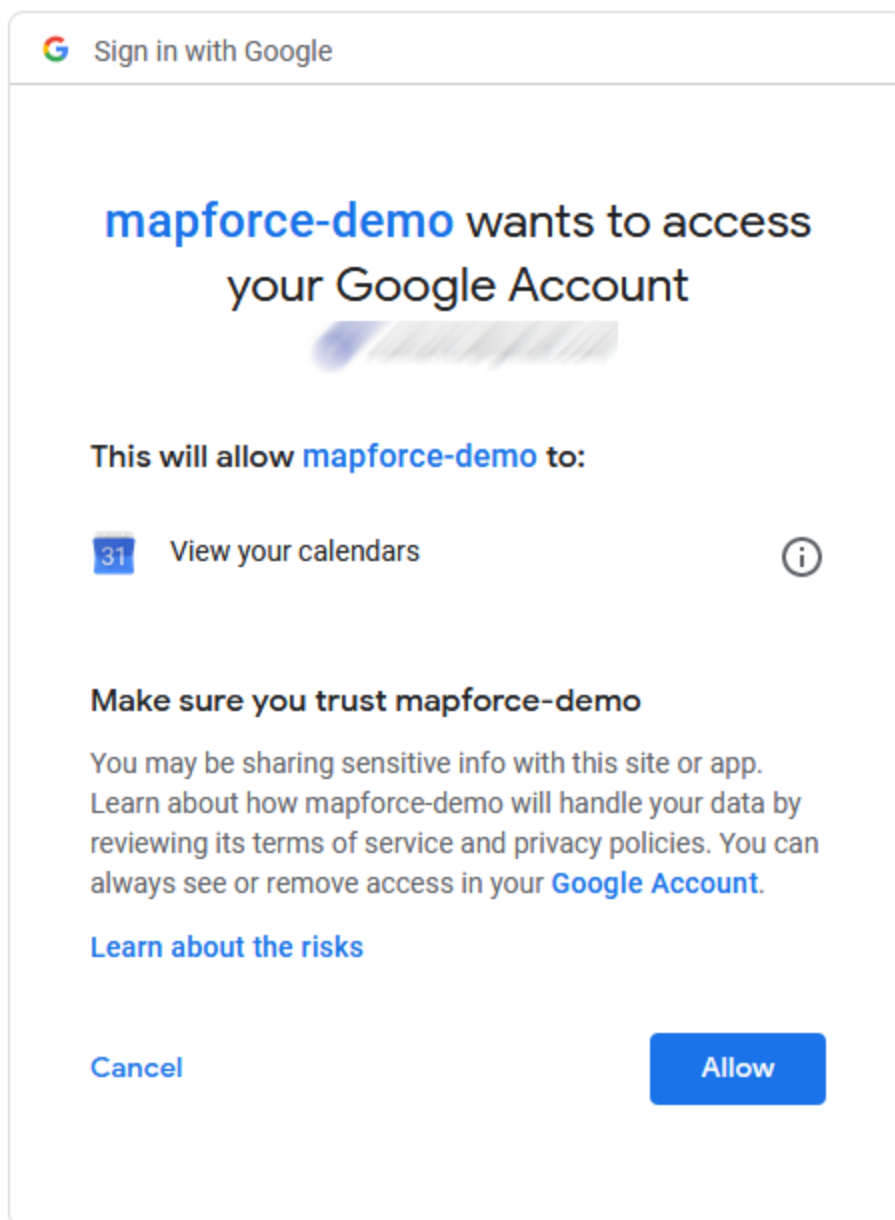
The screenshot shows a dialog box titled "Edit Credential" with a close button (X) in the top right corner. The dialog contains the following fields and options:

- Name:** A text input field containing "my.oauth".
- Type:** Radio buttons for "Password" (unselected) and "OAuth 2" (selected).
- Authorization Endpoint:** A text input field containing "https://accounts.google.com/o/oauth2/auth".
- Token Endpoint:** A text input field containing "https://oauth2.googleapis.com/token".
- Client ID:** A text input field containing a redacted value (represented by diagonal lines).
- Client Secret:** A text input field containing a redacted value (represented by dots) and a visibility toggle icon (an eye with a slash).
- Scope:** A text input field containing "https://www.googleapis.com/auth/calendar.readonly".

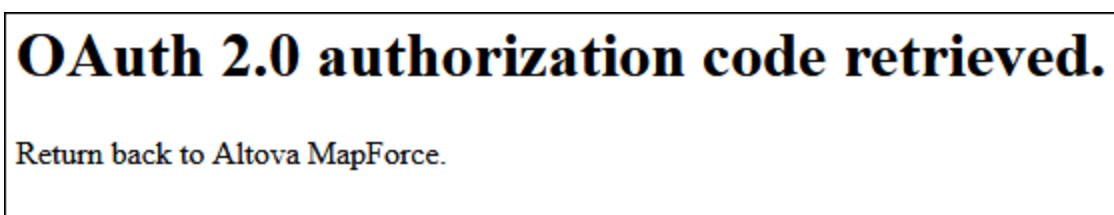
7. Cliquez sur **Demander jeton d'accès** pour obtenir le jeton depuis le serveur d'autorisation (dans cet exemple, Google). Une fenêtre de navigation s'ouvre demandant de vous connecter à votre compte Google.
8. Connexion à votre compte Google. Puisque vous n'avez pas encore soumis de requête de vérification d'appli à Google, la page suivante apparaît.



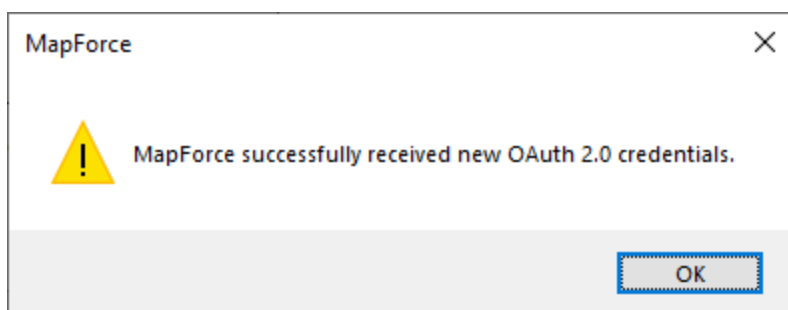
9. Cliquez sur **Avancé**, puis cliquez sur **Aller à mapforce-demo (unsafe)**.



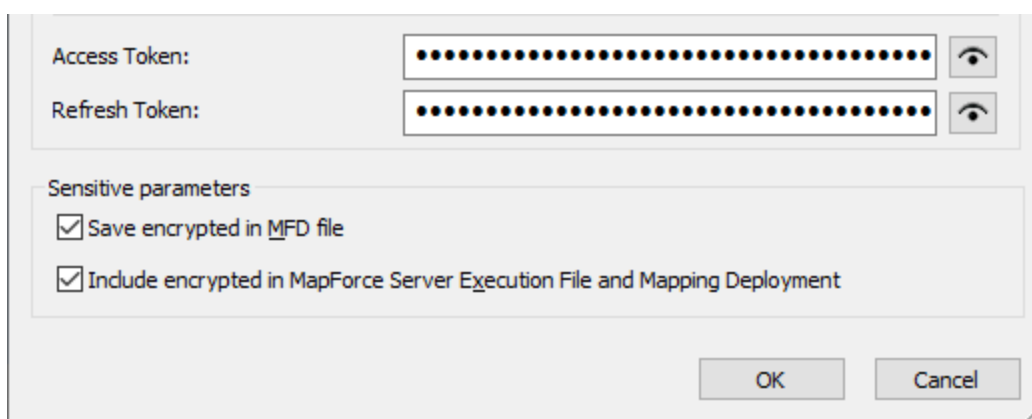
10. Cliquez sur **Autoriser**. Une confirmation est désormais affichée dans le navigateur.



MapForce vous informe également que le code d'autorisation OAuth 2.0 a été récupéré avec succès.



11. Cliquez sur **OK**. Notez que les champs **Jeton d'accès** et **Jeton d'actualisation** ont maintenant été complétés avec des données.



12. Enregistrez le mappage en tant que **GetCalendarEvents.mfd**.

Dans ce tutoriel, la case à cocher **Save encrypted in MFD file** est sélectionnée dans la boîte de dialogue Éditer identifiants. Pour cette raison, les champs sensibles **Secret client**, **Jeton d'autorisation** et **Jeton d'actualisation** seront enregistrés sous forme encryptée dans le fichier de création du mappage (.mfd) quand vous enregistrez le mappage.

Sachez que le jeton d'autorisation expirera après une période. Lorsque cela arrive, vous ne pourrez plus exécuter le mappage (à cette étape, aucun mappage n'a été conçu, mais il sera créé à une étape suivante). À chaque fois que vous devez obtenir un nouveau code d'autorisation manuellement, cliquez sur **Demander jeton d'accès** et suivez les étapes décrites ci-dessus.

Concevoir l'appel de service Web

Le mappage **GetCalendarEvents.mfd** créé ne fait rien pour le moment. La chose qu'il contient sont les identifiants OAuth 2.0 qui permettent l'accès à l'API Calendrier de Google.

Concevons maintenant le service Web dans MapForce, comme suit :

1. Ouvrez le mappage **GetCalendarEvents.mfd**.
2. Dans le menu **Insérer**, cliquez sur **Fonction du service Web**. La boîte de dialogue « Web Service Call Settings » apparaît.
3. Cliquez sur **Manual**.
4. Sélectionnez **GET** comme méthode de requête et entrez l'URL dans le service Web mentionné dans

l'étape précédente : `https://www.googleapis.com/calendar/v3/calendars/calendarId/events`.

5. Puisque **calendarId** est un espace réservé qui doit être alloué en tant que paramètre, encadrez-le à l'intérieur des accolades, tel qu'affiché ci-dessous.

Web Service Call Settings

Service definition

WSDL


Manual

Request Method: GET

Connection Settings

URL: `https://www.googleapis.com/calendar/v3/calendars/{calendarId}/events`

Timeout: 40 seconds Infinite Dynamic URL (supplied by mapping)

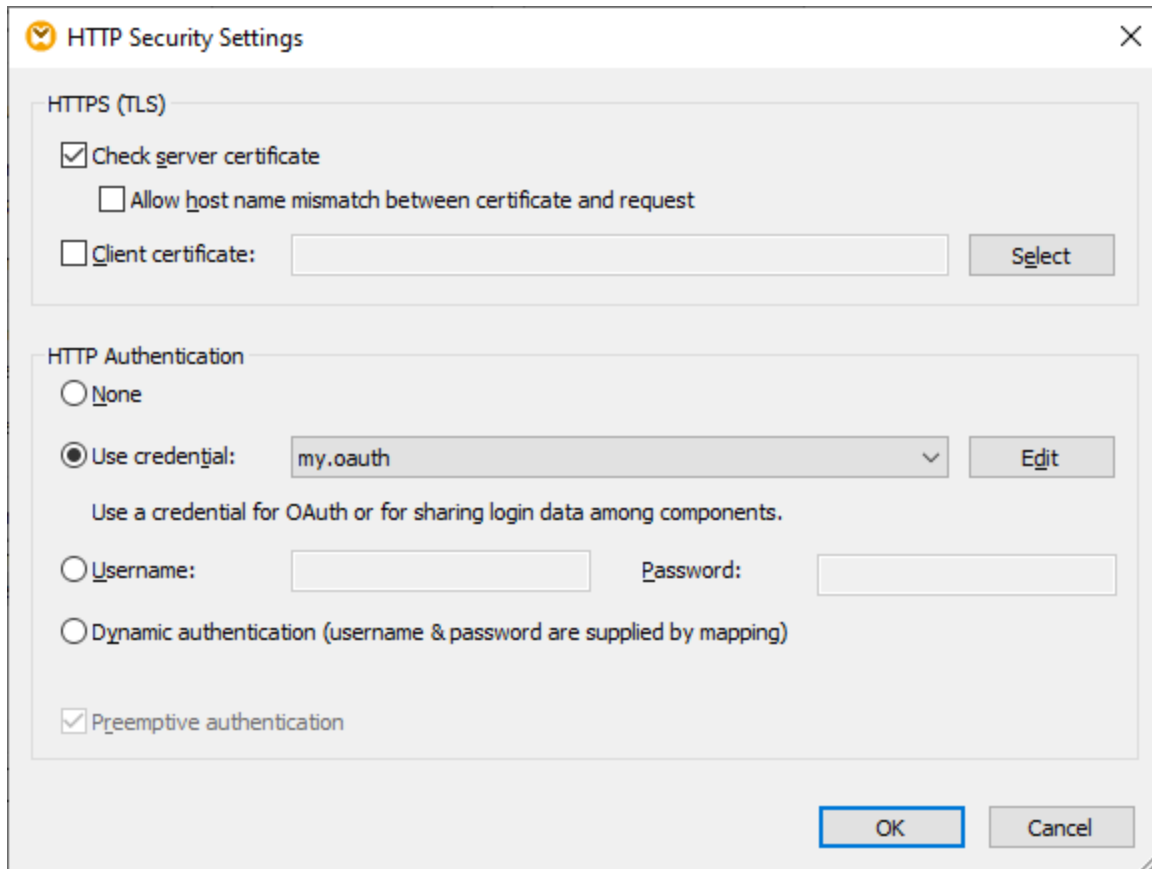
6. Cliquez sur le bouton  **Ajouter paramètre** et définissez les détails de paramètre comme suit :

Parameters:

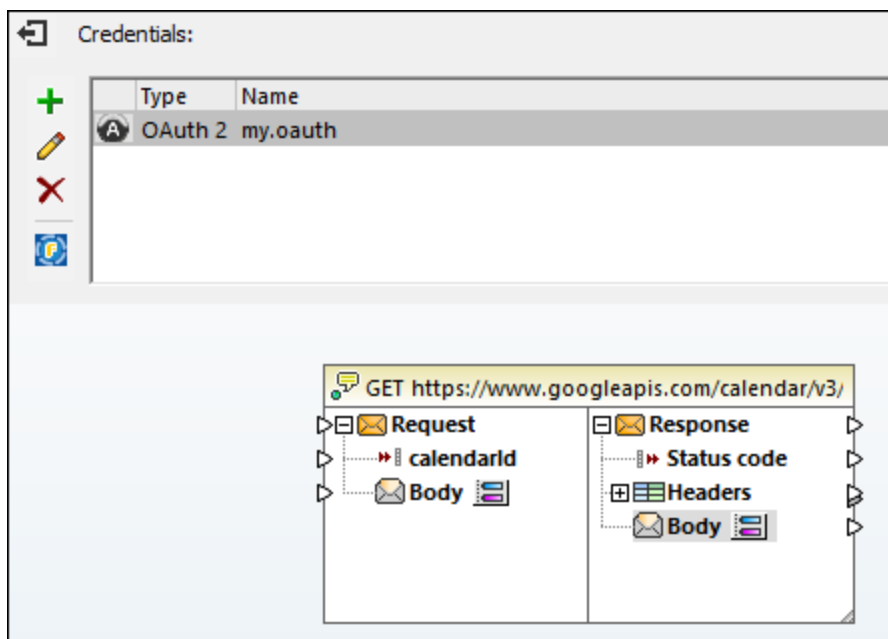
Name	Style	Type	Mappable	Fixed Value	Required	Repeating	Description
calendarId	Template	string	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Dans la configuration ci-dessus, le style du « Template » permet de remplacer la partie URL encadrée à l'intérieur des accolades avec la valeur paramètre au moment de l'exécution. « Mappable » signifie que vous pouvez fournir la valeur depuis le mappage (par exemple, depuis un paramètre d'entrée, ou éventuellement un paramètre de constante). Enfin, le paramètre a été marqué en tant que « requis » parce que l'appel API ne peut pas être réalisé sans.

7. Cliquez sur le bouton **Éditer** adjacent aux **paramètres de sécurité HTTP**.
8. Dans la boîte de dialogue « Paramètres de sécurité HTTP », sélectionnez **Use Credential** et sélectionnez l'enregistrement d'identification « my.oauth » configuré précédemment.

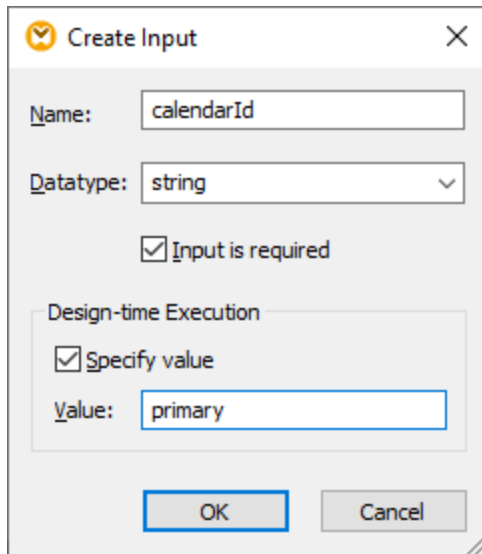


Le service Web configuré jusqu'à cette étape a l'apparence suivante sur le mappage :



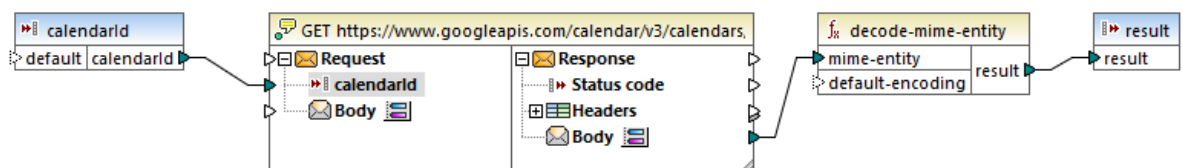
Vous pouvez maintenant terminer le design en effectuant les étapes suivantes :

1. Dans le menu **Insérer**, cliquez sur **Insert Input**, et configurez les composants comme suit :



La composante d'entrée, telle qu'illustrée ci-dessus, est dotée de la valeur « primary » au moment du design. Selon la documentation API, la valeur « primary » signale au serveur API d'accéder au Calendrier principal de Google de l'utilisateur actuellement connecté. Notez que la valeur est une valeur design-time et s'applique uniquement lorsque vous prévisualisez le mappage dans MapForce. Lorsque le mappage est exécuté dans un environnement de serveur, vous allez devoir fournir la valeur désirée au moment de l'exécution.

2. Glissez la fonction **decode-mime-entity** depuis la fenêtre Libraries dans l'espace de mappage. Cette fonction convertit la partie MIME brute reçue du serveur en un string.
3. Dans le menu **Insérer**, cliquez sur **Insérer Résultat**, et ajoutez une composante de résultat simple dont le rôle est de générer le résultat en tant que string normal.
4. Établissez les connexions entre les composantes, tel qu'illustré ci-dessous.



Ceci conclut la partie design dans MapForce.

Tester l'exécution de mappage

Pour tester l'exécution de mappage dans MapForce, cliquez sur l'onglet **Résultats** et notez le résultat affiché dans la fenêtre des Messages.

Si vous recevez une erreur d'autorisation telle que « non autorisé (401) », notez les conseils relatifs à la résolution des problèmes suivants :

1. Assurez-vous que l'API Calendrier de Google est activée, voir [Activer l'API Calendrier de Google](#)⁶⁰.
2. [Demander un nouveau jeton d'autorisation](#)⁶¹, dans le cas où le jeton d'accès précédemment obtenu est venu à expiration.

3. Vérifier que tous les détails OAuth 2.0 ont été saisis correctement dans MapForce.

Si vous avez réussi l'exécution et l'autorisation OAuth 2.0 de MapForce, le résultat de mappage devrait normalement être similaire à celui ci-dessous :

The screenshot shows the MapForce interface. The top pane displays a JSON response for a calendar event. The bottom pane shows the 'Messages' window with two success messages for the 'GetCalendarEvents.mfd' mapping.

```

1  {
2    "kind": "calendar#events",
3    "etag": "\"p32gbjdmvo63ek0g\"",
4    "summary": " ",
5    "updated": "2020-06-16T14:10:43.876Z",
6    "timeZone": "Europe/Vienna",
7    "accessRole": "owner",
8    "defaultReminders": [
9      {
10     "method": "email",
11     "minutes": 10
12    },
13    {
14     "method": "popup",
15     "minutes": 30
16    }
17  ],
18  "nextSyncToken": "CKC5tt_BhuoCEKC5tt_BhuoCGAU=",
19  "items": []
20 }

```

Messages:

- GetCalendarEvents.mfd: Mapping validation successful. - 0 error(s), 0 warning(s)
- GetCalendarEvents.mfd: Execution successful - 0 error(s), 0 warning(s)

Si vous avez utilisé un compte Google qui ne contient pas d'événements de calendrier comme dans cet exemple, l'array « items » sera vide en affichant la réponse. Par contre, si vous ajoutez un événement à votre calendrier Google et exécutez le mappage une nouvelle fois, le résultat reflétera cet ajout. En note rapide, vous pouvez aussi extraire des événements d'un autre calendrier que le calendrier par défaut. Par exemple, vous pouvez extraire des données depuis un calendrier public comme dans celui des « Jours fériés aux États-Unis d'Amérique ». Pour ce faire, définissez la valeur du paramètre de **calendarId** à **en.usa#holiday@group.v.calendar.google.com** au lieu de **primary**.

Pour toute information sur d'autres paramètres que vous pouvez ajouter à l'appel API, consultez la documentation de la méthode API au lien <https://developers.google.com/calendar/v3/reference/events/list>.

Exécuter le mappage avec MapForce Server (autonome)

Cette section traite spécifiquement l'exécution du mappage OAuth 2.0 démo avec MapForce Server installé en tant que produit autonome, et non avec FlowForce Server management. Pour toute information sur l'exécution de tels mappages avec MapForce Server dans FlowForce Server management, consultez la documentation de FlowForce Server, où cet exemple est poursuivi.

Pour exécuter un mappage OAuth 2.0 avec MapForce Server autonome, il y a deux manières de traiter les identifications OAuth 2.0 :

- Inclure un jeton OAuth 2.0 (sous forme encrytée) dans le fichier .mfx compilé. Avec cette approche,

vous devez fournir toute information d'identification OAuth 2.0 dans la ligne de commande (ou dans l'appel API de MapForce Server) parce que l'identification incorporée sera utilisée. Toutefois, ceci signifie que chacun avec un accès au fichier .mfx sera à même de l'exécuter sans fournir le jeton d'autorisation—jusqu'à expiration de celui-ci ou si le serveur d'autorisation le révoque. Plus important encore, vous pouvez toujours écraser le jeton d'autorisation depuis la ligne de commande sans avoir à recompiler le fichier .mfx (voir la prochaine puce).

- Ne pas inclure le jeton OAuth 2.0 dans le fichier .mfx compilé. Avec cette approche, vous (ou tout autre utilisateur qui exécute un fichier .mfx) devez fournir un jeton d'autorisation OAuth 2.0 dans la ligne de commande ou l'appel API de MapForce Server. Le jeton d'autorisation lui-même doit être obtenu en dehors de MapForce Server, par exemple avec MapForce, tel que décrit précédemment.

Dans cet exemple, le jeton d'autorisation ne sera pas inclus dans le fichier .mfx compilé. Mais il sera fourni au moment de l'exécution.

1. Dans MapForce, cliquez avec la touche de droite sur une zone vide du mappage et sélectionnez **Ouvrir Manager d'identifiants**.
2. Double-cliquez sur l'enregistrement de l'identifiant (« my.oauth », dans cet exemple) et désactivez la case à cocher **Inclure dans le fichier d'exécution et le déploiement de mappage dans MapForce Server**.
3. Enregistrez le fichier design de mappage (.mfd).

Compilons maintenant le mappage vers un fichier d'exécution (.mfx) de MapForce Server :

1. Dans le menu **Fichier**, cliquez sur **Compiler vers un fichier d'exécution (.mfx) de MapForce Server**.
2. Sélectionnez un répertoire de destination et enregistrez le fichier en tant que **GetCalendarEvents.mfx**.

Vous pouvez désormais ouvrir une fenêtre d'invite de commandes et exécutez le fichier .mfx file avec une commande telle que :

```
mapforceserver-exec run GetCalendarEvents.mfx --p=calendarId:"primary" --  
credential=my.oauth:oauth:token=mytoken
```

Emplacement :

- **mapforceserver-exec** est le chemin vers le fichier exécutable de MapForce Server, typiquement **C:\Program Files\Altova\MapForceServer2024\bin\MapForceServer.exe**.
- **GetCalendarEvents.mfx** est le chemin vers le fichier .mfx par rapport au répertoire actuel de la ligne de commande. Ajustez le chemin, le cas échéant, ou utilisez un chemin absolu.
- **calendarId** est le nom du paramètre d'entrée tel qu'il a été créé dans MapForce
- **my.oauth** est le nom de l'identification tel qu'elle a été créée dans MapForce lors d'une étape précédente.
- **mytoken** est la valeur du jeton d'autorisation obtenu de manière externe (dans ce cas, avec MapForce).

Lorsque l'exécution et l'autorisation OAuth 2.0 ont réussi, le résultat de la ligne de commande affiche la réponse retournée par l'API Calendrier de Google, par exemple :

```
Command Prompt
C:\OAuth 2.0 Demo>"C:\Program Files\Altova\MapForceServer2020\bin\MapForceServer.exe" run
"C:\OAuth 2.0 Demo\GetCalendarEvents.mfx" --p=calendarId:"primary" --credential=my.oauth:0
auth:token=[REDACTED]
Zy
{
  "kind": "calendar#events",
  "etag": "\"p338brhcio25uk0g\"",
  "summary": [REDACTED],
  "updated": "2020-06-18T13:46:52.898Z",
  "timeZone": "Europe/Vienna",
  "accessRole": "owner",
  "defaultReminders": [
    {
      "method": "email",
      "minutes": 10
    },
    {
      "method": "popup",
      "minutes": 30
    }
  ],
  "nextSyncToken": "CNC9xZLai-oCENC9xZLai-oCGAU=",
  "items": []
}
Execution successful.
U:\OAuth 2.0 Demo>
```

N'oubliez pas que le jeton d'autorisation expire très rapidement (l'intervalle dépend du serveur d'autorisation, qui est Google dans ce cas) et vous devrez éventuellement à en demander un autre si vous recevez des erreurs « non autorisées », voir [Demander un jeton d'autorisation](#) ⁶¹.

4.5 Authentification dynamique

Dans MapForce, il est possible de configurer des mappages qui appellent les services Web pour une authentification HTTP de base. L'authentification dynamique est un des moyens d'y parvenir ; il s'agit d'une alternative à l'utilisation des identifiants. L'authentification dynamique signifie concevoir le mappage de manière à ce qu'il accepte le nom d'utilisateur et le mot de passe en tant que paramètres d'entrée. Pour plus de détails concernant la configuration de l'authentification dynamique, veuillez vous référer à la documentation MapForce (<https://www.altova.com/fr/documentation>).

Si vous avez configuré le mappage pour l'authentification dynamique, alors, le nom d'utilisateur et le mot de passe respectifs doivent être fournis en tant que paramètres au moment de la marche de mappage. Cela n'est pas différent de la fourniture habituelle d'autres types de paramètres au mappage. Par exemple, lors de l'appel de MapForce Server dans la ligne de commande, la syntaxe pour un mappage comme celui au-dessus est :

```
<exec> run mapping.mfx --p=username:admin --p=password:dj9JaVax
```

Où :

- `<exec>` est le chemin vers le programme d'exécution MapForce Server. Ce chemin peut être soit absolu soit, si le répertoire actuel est le même que celui du programme d'exécution, vous pouvez uniquement saisir le nom du programme d'exécution.
- `username` et `password` sont les noms des paramètres d'entrée respectifs dans le mappage MapForce.

Lors de l'appel de l'API MapForce Server, vous pouvez authentifier le mappage en appelant la méthode `AddParameter` avant d'appeler la méthode `Run`. Par exemple, dans C#, le code pour terminer cela ressemblerait à :

```
try
{
    Altova.MapForceServer.Server mfs = new Altova.MapForceServer.Server();
    mfs.AddParameter("username", "admin");
    mfs.AddParameter("password", "dj9JaVax");
    mfs.WorkingDirectory = "C:\\Work";
    if(mfs.Run("C:\\Work\\mapping.mfx"))
    {
        Console.WriteLine("Success");
    }
    else
    {
        Console.WriteLine(mfs.LastExecutionMessage);
    }
    Console.ReadLine();
}
catch(Exception ex)
{
    Console.WriteLine(ex);
}
```

Une fois de plus, le `username` et `password` (premier argument de la méthode `AddParameter`) doivent être les mêmes que les noms des paramètres d'entrée respectifs dans le mappage MapForce.

5 Ligne de commande MapForce Server

MapForce Server fournit une interface de ligne de commande que vous pouvez utiliser pour des tâches administratives telles que la gestion des licences, et pour exécuter les fichiers d'exécution de mappage (.mfx). Les commandes disponibles sont recensées ci-dessous.

- [assignlicense](#)⁷⁵ : (uniquement Windows) télécharge une licence vers le LicenseServer et attribue la licence à MapForce Server.
- [exportresourcestrings](#)⁷⁶ : exporte tous les strings de ressource d'application vers un fichier XML.
- [help](#)⁷⁸ : affiche des informations sur la commande qui est soumise comme argument (ou sur toutes les commandes si aucun argument n'est soumis).
- [licenseserver](#)⁷⁹ : Enregistre MapForce Server avec un LicenseServer sur le réseau local.
- [run](#)⁸⁰ : exécute un fichier d'exécution de mappage (.mfx) compilé avec MapForce.
- [setdeflang](#)⁸⁴ : définit la langue par défaut de MapForce Server.
- [verifylicense](#)⁸⁵ : (uniquement Windows) vérifie si la version actuelle de MapForce Server est sous licence et, en option, si la version est sous licence avec la clé de licence donnée.
- [version](#)⁸⁶ : affiche le numéro de la version de MapForce Server.

Pour appeler MapForce Server dans la ligne de commande, vous devez connaître le chemin du programme d'exécution tel qu'applicable dans votre système d'exploitation. Par défaut, le programme d'exécution MapForce Server est installé sur le chemin suivant :

<i>Linux</i>	/opt/Altova/MapForceServer2024/bin/mapforceserver
<i>macOS</i>	/usr/local/Altova/MapForceServer2024/bin/mapforceserver
<i>Windows</i>	C:\Program Files\Altova\MapForceServer2024\bin\MapForceServer.exe

Note: Si MapForce Server 32-bit est installé sur Windows 64-bit, changer **C:\Program Files** en **C:\Program Files (x86)**.

Par convention, cette documentation omet le chemin complet du programme d'exécution lors de la description d'une commande donnée et utilise **mapforceserver** au lieu du nom du programme d'exécution, par exemple :

```
mapforceserver help
```

mapforceserver est le chemin du programme d'exécution. Veuillez noter que, si vous utilisez un chemin absolu, vous pourrez exécuter des commande quelque soit le répertoire actuel dans lequel votre fenêtre (terminal) d'invite de commande se trouve. Néanmoins, si vous souhaitez appeler le programme d'exécution en ne saisissant que son nom, veuillez vous assurer de suivre l'une des deux étapes auparavant :

- Changer le répertoire actuel du terminal dans le répertoire d'installation de MapForce Server
- Ajouter le répertoire dans lequel le programme d'exécution se trouve à la variable d'environnement PATH.

Ces deux scénarios sont décrits en détail ci-dessous.

Trucs et astuces

Si vous ne maîtrisez pas encore la ligne de commande, nous vous proposons les astuces suivantes.

- Pour connaître le répertoire actuel dans lequel se trouve la ligne de commande, saisir `pwd` sur Linux et macOS. Sur Windows, saisir `echo %CD%`.
- Utilisez la touche **Tab** pour vous déplacer rapidement dans les chemins de dossier ou de répertoire sans devoir les saisir en entier. Par exemple, si vous saisissez `cd c:\prog` dans la ligne de commande, puis que vous appuyez sur **Tab**, `C:\Program Files` sera rempli automatiquement (ou peut-être un autre répertoire sous C:\ dont le nom commence avec "Prog").
- Lorsque vous saisissez des chemins qui contiennent un espace blanc, comme `C:\Program Files` sur Windows, veillez à les contenir dans des guillemets.
- Si vous voyez un message semblable à "Cette commande n'est pas reconnue en tant que commande interne ou externe, programme opérable ou fichier batch", il s'agit probablement d'une erreur d'écriture du chemin ou de la commande.
- Sur Linux, veuillez vous assurer que vous utilisez la casse correcte pour les noms de dossier ou de répertoire. Par exemple, la saisie d'un chemin comme `/home/nikita/downloads` retournera une erreur si le nom du répertoire est en fait `/home/nikita/Downloads`.
- Lorsque vous saisissez un chemin sur Linux ou sur macOS, utiliser la barre oblique vers l'avant, contrairement aux barres obliques vers l'arrière sur Windows.

Comment exécuter une commande

1. Ouvrir une fenêtre d'invite de commande.
 - a. Pour ouvrir une invite de commande sur Windows, appuyer sur la touche **Windows** et commencer à saisir `cmd`. Cliquer sur la suggestion **Command Prompt** qui apparaît.
 - b. Pour ouvrir un terminal sur Mac, cliquer sur l'icône **Finder**, puis sélectionner **Go > Utilities** depuis le menu. Double-cliquer sur l'icône **Terminal** dans la fenêtre Utilities.
 - c. Si vous exécutez Linux depuis une interface d'utilisateur graphique, situer et exécuter la commande **Terminal** telle qu'applicable à votre distribution . Si vous exécutez Linux depuis une interface de ligne de commande, ignorer cette étape.
2. Saisir le chemin complet vers le programme d'exécution, suivi par la commande que vous souhaitez exécuter. Par exemple, la commande ci-dessous propose de l'aide dans la ligne de commande.

<i>Linux</i>	<code>/opt/Altova/MapForceServer2024/bin/mapforceserver help</code>
<i>macOS</i>	<code>/usr/local/Altova/MapForceServer2024/bin/mapforceserver help</code>
<i>Windows</i>	<code>C:\Program Files (x86)\Altova\MapForceServer2024\bin\MapForceServer.exe help</code>

Dans l'exemple ci-dessus, la commande `help` était exécutée sans options ou arguments. D'autres commandes peuvent avoir des arguments et des options, et ces arguments et options peuvent être obligatoires ou optionnelles. Par exemple, la commande `run` a un argument obligatoire qui vous permet de fournir le nom du chemin ou du fichier du fichier .mfx que vous souhaitez exécuter. Consultez la section de référence pour plus de détails concernant chaque commande .

Appeler MapForce Server dans le répertoire d'installation

Pour appeler le programme d'exécution sans devoir saisir le chemin complet, changer le répertoire actuel dans le répertoire sur lequel le programme d'exécution MapForce Server a été installé, par exemple :

<i>Linux</i>	<code>cd /opt/Altova/MapForceServer2024/bin</code>
--------------	--

<i>macOS</i>	<code>cd /usr/local/Altova/MapForceServer2024/bin</code>
<i>Windows</i>	<code>cd "C:\Program Files (x86)\Altova\MapForceServer2024\bin\MapForceServer.exe"</code>

Vous pouvez maintenant exécuter toute commande en saisissant uniquement le nom du programme d'exécution, par exemple :

<i>Linux</i>	<code>./mapforceserver help</code>
<i>macOS</i>	<code>./mapforceserver help</code>
<i>Windows</i>	<code>MapForceServer.exe help</code>

Note : Sur les systèmes Linux et macOS, le préfixe `./` indique que le programme d'exécution se trouve dans le répertoire actuel.

Appeler MapForce Server depuis tout répertoire

Pour appeler le programme d'exécution depuis tout répertoire, référez-vous-y en utilisant le chemin absolu. En alternative, si vous souhaitez appeler le programme en saisissant uniquement le nom du programme d'exécution, vous pouvez éditer la variable d'environnement PATH de votre système d'exploitation de manière à ce qu'il inclut le chemin complet vers le répertoire d'installation MapForce Server. Pour consulter les différents moyens de modifier la variable d'environnement PATH, se référer à la documentation de votre système d'exploitation.

Note: Une fois avoir changé la variable d'environnement PATH, vous devrez éventuellement fermer la fenêtre de terminal et en ouvrir une nouvelle, pour que les modifications prennent effet.

5.1 assignlicense

Syntaxe et description

La commande `assignlicense` télécharge un fichier de licence vers Altova LicenseServer avec lequel MapForce Server est enregistré (voir la commande `licenseserver`), et attribue la licence à MapForce Server. Elle prend le chemin d'un fichier de licence en tant que son argument. La commande vous permet aussi de tester la validité d'une licence.

```
mapforceserver assignlicense [options] FILE
```

- L'argument `FILE` prend le chemin du fichier de licence.
- L'option `--test-only` charge le fichier de licence sur LicenseServer et valide la licence, mais n'attribue pas la licence à MapForce Server.

Pour plus de détails concernant la licence, voir la documentation LicenseServer (<https://www.altova.com/manual/fr/licenseserver/3.14/>).

Exemples

Des exemples de la commande `assignlicense` :

```
mapforceserver assignlicense C:\licensepool\mylicensekey.altova_licenses
mapforceserver assignlicense --test-only=true C:
\licensepool\mylicensekey.altova_licenses
```

- La première commande ci-dessus charge la licence spécifiée sur LicenseServer et l'attribue à MapForce Server.
- La troisième commande charge la licence spécifiée sur LicenseServer et la valide, sans l'attribuer à MapForce Server.

Options

Options are listed in short form (if available) and long form. You can use one or two dashes for both short and long forms. An option may or may not take a value. If it takes a value, it is written like this: `--option=value`. Values can be specified without quotes except in two cases: (i) when the value string contains spaces, or (ii) when explicitly stated in the description of the option that quotes are required. If an option takes a Boolean value and no value is specified, then the option's default value is `TRUE`. Use the `--h, --help` option to display information about the command.

▼ test-only [t]

```
--t, --test-only = true|false
```

Les valeurs sont `true|false`. Si `true`, alors le fichier de licence est chargé sur LicenseServer et validé, mais pas attribué.

5.2 exportresourcestrings

Syntaxe et description

La commande `exportresourcestrings` sort un fichier XML contenant les strings de ressource de l'application MapForce Server dans la langue spécifiée. Les langues d'exportation disponibles sont allemand (de), anglais (en), espagnol (es), français (fr) et japonais (ja).

```
mapforceserver exportresourcestrings [options] LanguageCode XMLOutputFile
```

- L'argument `LanguageCode` donne la langue des strings de ressource dans le fichier XML de sortie ; il s'agit de la langue d'exportation *export language*. Les langues d'exportation autorisées (avec leurs codes de langue respectifs entre parenthèses) sont : allemand (de), anglais (en), espagnol (es), français (fr) et japonais (ja).
- L'argument `XMLOutputFile` spécifie le chemin et le nom du fichier XML de sortie .

Pour créer des localisations, veuillez noter les points suivants.

Exemples

Exemples de la commande `exportresourcestrings` :

```
mapforceserver exportresourcestrings de c:\Strings.xml
```

- La commande ci-dessus crée un fichier appelé `Strings.xml` sous `c:\` qui contient les strings de ressource de MapForce Server en allemand.

Créer des versions localisées de MapForce Server

Vous pouvez créer une version localisée de MapForce Server pour toute langue de votre choix. Cinq versions localisées (allemand, anglais, espagnol, français et japonais) sont déjà disponibles sous le dossier `C:\Program Files (x86)\Altova\MapForceServer2024\bin` , et n'ont donc pas besoin d'être créées.

Créer une version localisée comme suit :

1. Générer un fichier XML contenant des strings de ressource en utilisant la commande `exportresourcestrings` (voir la syntaxe de commande ci-dessus). Les strings de ressource dans ce fichier XML seront ceux d'une des cinq langues prises en charge : allemand (de), anglais (en), espagnol (es), français (fr) ou japonais (ja), conformément à l'argument `LanguageCode` utilisé avec la commande.
2. Traduire les strings de ressource à partir de l'une des cinq langues prises en charge dans la langue cible. Les strings de ressource sont les contenus des éléments `<string>` dans le fichier XML. Ne pas traduire les variables dans les accolades comme `{option}` ou `{product}`.
3. Contactez [Altova Support](#) pour générer un fichier DLL MapForce Server localisé à partir de votre fichier XML traduit.
4. Après avoir reçu votre fichier DLL localisé de [Altova Support](#), enregistrez le fichier DLL sous le dossier `C:\Program Files (x86)\Altova\MapForceServer2024\bin` . Votre fichier DLL aura un nom de la forme `MapForceServer2024_lc.dll`. La partie `_lc` du nom contient le code de langue. Par exemple, dans `MapForceServer2024_de.dll`, la partie `de` est le code de langue pour l'allemand (Deutsch).
5. Exécutez la commande `setdeflang` pour définir votre fichier DLL localisé DLL en tant qu'application MapForce Server à utiliser. Pour l'argument de la commande `setdeflang`, utilisez le code de langue

qui fait partie du nom DLL.

Note : Altova MapForce Server est livré avec une prise en charge pour les cinq langues : allemand, anglais, espagnol, français et japonais. Donc vous ne devez pas créer de version localisée pour ces langues. Pour définir une de ces langues en tant que langue par défaut, utilisez la commande MapForce Server de `setdeflang`.

5.3 help

Syntaxe et description

La commande `help` prend un seul argument (`Command`), qui est le nom de la commande pour laquelle l'aide est requise. Elle affiche la syntaxe de la commande, ses options, et d'autres informations pertinentes. Si l'argument `Command` n'est pas spécifié, toutes les commandes du programme d'exécution sont recensés, chacun présentant une brève description de texte.

```
mapforceserver help Command
```

Exemple

Exemple de la commande `help` pour afficher des informations concernant la commande `licenseserver` :

```
mapforceserver help licenseserver
```

L'option `--help`

L'information Help à propos d'une commande est également disponible en utilisant l'option `--help` avec la commande pour laquelle l'information d'aide est nécessaire. Les deux commandes ci-dessous produisent les mêmes résultats :

```
mapforceserver licenseserver --help
```

La commande ci-dessus utilise l'option `--help` de la commande `licenseserver`.

```
mapforceserver help licenseserver
```

La commande `help` prend `licenseserver` en tant que son argument.

Les deux commandes affichent des informations d'aide concernant la commande `licenseserver`.

5.4 licenseserver

Syntaxe et description

La commande `licenseserver` enregistre MapForce Server avec Altova LicenseServer spécifié par l'argument `Server-Or-IP-Address`. Pour pouvoir exécuter la commande `licenseserver` correctement, les deux serveurs (MapForce Server et LicenseServer) doivent être connectés sur le réseau et le LicenseServer doit fonctionner. Vous devez posséder des privilèges d'administrateur pour pouvoir enregistrer MapForce Server auprès du LicenseServer.

```
mapforceserver licenseserver [options] Server-Or-IP-Address
```

- L'argument `Server-Or-IP-Address` prend le nom ou l'adresse IP de l'appareil de LicenseServer.

Une fois que MapForce Server a été enregistré avec succès auprès de LicenseServer, vous recevrez un message. Le message affichera aussi l'URL du LicenseServer. Vous pouvez maintenant vous rendre sur LicenseServer pour attribuer une licence à MapForce Server. Pour plus de détails concernant la licence, voir la documentation LicenseServer (<https://www.altova.com/manual/fr/licenseserver/3.14/>).

Exemples

Exemples de la commande `licenseserver` :

```
mapforceserver licenseserver DOC.altova.com
mapforceserver licenseserver localhost
mapforceserver licenseserver 127.0.0.1
```

Les commandes ci-dessus spécifient, respectivement, la machine nommée `DOC.altova.com`, et la machine de l'utilisateur (`localhost` et `127.0.0.1`) qui fait marcher Altova LicenseServer. Dans tous les cas, la commande enregistre MapForce Server avec le LicenseServer sur la machine spécifiée. La dernière commande appelle le programme d'exécution de serveur pour exécuter la commande.

Options

Options are listed in short form (if available) and long form. You can use one or two dashes for both short and long forms. An option may or may not take a value. If it takes a value, it is written like this: `--option=value`. Values can be specified without quotes except in two cases: (i) when the value string contains spaces, or (ii) when explicitly stated in the description of the option that quotes are required. If an option takes a Boolean value and no value is specified, then the option's default value is `TRUE`. Use the `--h, --help` option to display information about the command.

▼ json [j]

```
--j, --json = true|false
```

Les valeurs sont `true|false`. Si `true`, imprime le résultat de la tentative d'enregistrement en tant qu'objet JSON parsable par machine.

5.5 run

La commande `run` exécute un fichier d'exécution MapForce Server (fichier `.mfx`) fourni en tant qu'argument. Le fichier d'exécution MapForce Server est créé avec MapForce; il représente principalement un mappage compilé pour l'exécution de serveur.

Tout fichier d'entrée requis par le mappage est censé se trouver dans le chemin spécifié au niveau du moment de design de mappage dans MapForce. Si MapForce Server n'est pas exécuté sur le même système d'exploitation en tant que MapForce, les fichiers d'entrée requis par le mappage doivent être copiés sur l'appareil cible avec le fichier `.mfx`, et doivent être référencés avec un chemin relatif. Pour plus d'informations concernant la configuration un mappage par rapport aux chemins relatifs ou absolus, se référer à la documentation MapForce (<https://www.altova.com/fr/documentation#mapforce>). Les autres exigences préalables peuvent s'appliquer, selon la manière dont vous avez conçu le mappage, voir [Préparer les mappages pour l'Exécution de serveur](#)⁴³.

Si le mappage retourne une valeur simple, cette sortie est écrite dans le stream `stdout` (sortie standard). D'un autre côté, le succès et les messages d'erreur sont disponibles dans le stream `stderr` (erreur standard). Si vous ne souhaitez pas afficher le stream de sortie standard sur l'écran avec les messages de réussite ou d'erreur, rediriger soit le stream de sortie standard, soit le stream d'erreur the standard (ou les deux) vers les fichiers. Si ni les streams `stdout` ni `stderr` ne sont redirigés, ils sont tous les deux affichés sur l'écran.

Par exemple, pour rediriger le stream de sortie standard vers un fichier, utilisez :

```
mapforceserver run MyMapping.mfx > MyOutput.txt
```

Pour rediriger le stream d'erreur standard sur un fichier, utiliser :

```
mapforceserver run MyMapping.mfx 2> Diagnostics.log
```

Pour rediriger les deux streams simultanément, utiliser :

```
mapforceserver run MyMapping.mfx > MyOutput.txt 2> Diagnostics.log
```

Pour plus d'informations concernant la redirection de stream, veuillez vous référer à la documentation de votre shell de commande du système d'exploitation.

Syntaxe

Windows `MapForceServer run [options] MfxFile`

Linux `mapforceserver run [options] MfxFile`

Mac `mapforceserver run [options] MfxFile`

Options de commande

	--catalog	Spécifie le chemin absolu vers un fichier de catalogue racine qui n'est pas le fichier de catalogue racine installé. La valeur par défaut est le chemin absolu vers le fichier de catalogue racine installé, voir Fichiers Catalogue ¹⁸⁴ . Forme : --catalog=FILE
--cert	--certificatespath	Cette commande s'applique au Serveur MapForce Server qui marche sur Linux. Il spécifie le chemin vers le répertoire où les fichiers de certificats requis par le mappage sont stockés. Forme : --certificatespath=DIRECTORY Voir aussi Digital Certificate Management.
--cred	--credential=KEY:VALUE	La configuration de cette option est utile si le mappage contient des identifiants que vous avez définis au moment de la conception du mappage dans MapForce. KEY est le nom d'identifiant tel qu'il a été défini dans MapForce. VALUE consiste en une ou plusieurs paires nom-valeur, par exemple : name1=value1&name2=value2 Où name est un des suivants : <ul style="list-style-type: none"> • username • password • oauth:token Lorsque vous utilisez plusieurs paires nom-valeur séparées par une esperluette, vous devez contenir la partie the KEY:VALUE entre guillemets. Pour plus de détails, voir Identifiants ⁵⁴ .
--gc	--globalresourceconfig	Cette option est applicable si le mappage consomme des Ressources globales ⁴⁹ . Elle spécifie le nom de la configuration de ressource globale. Cette option doit être utilisée avec l'option --globalresourcefile. Forme : --gc=VALUE
--gr	--globalresourcefile	Cette option est applicable si le mappage consomme des Ressources globales ⁴⁹ . Elle spécifie le chemin du fichier de définition de la ressource globale. Cette option doit être utilisée avec l'option --globalresourceconfig. Forme : --gr=FILE.
--l	--lang	La langue utilisée pour afficher les messages. Forme : --lang=VALUE (en,de,ja,es,fr)
--p	--param	Cette option est applicable si le mappage a été conçu pour prendre des paramètres d'entrée. Elle attribue une valeur à un paramètre défini dans le mappage. Forme: --param=ParamName:ParamValue. Le switch --param doit être utilisé avant chaque paramètre. Utiliser des guillemets si

		<p><i>ParamName</i> ou <i>ParamValue</i> contient un espace. Par exemple : <code>--p=company:"Nanonull Inc"</code>.</p> <p>Pour plus d'informations concernant les mappage qui prennent des paramètres d'entrée, veuillez vous référer à la documentation MapForce (https://www.altova.com/fr/documentation#mapforce).</p>
	<code>--taxonomy-package</code>	<p>Spécifie le chemin absolu vers un paquet de taxonomie XBRL supplémentaire tel que décrit dans les recommandations de Taxonomy Packages 1.0. La valeur de FILE (Fichier) indique l'emplacement du paquet de taxonomie. Ajouter l'option plusieurs fois pour spécifier plus d'un paquet de taxonomie.</p> <p>Forme : <code>--taxonomy-package=FILE</code></p>
	<code>--taxonomy-packages-config-file</code>	<p>Spécifie le chemin vers un fichier de configuration appelé TaxonomyPackagesConfig.json, utilisé pour charger les paquets de taxonomie XBRL. Ce fichier de configuration est mis à jour à chaque fois que vous ajoutez, supprimez, activez ou désactivez les paquets de taxonomie XBRL depuis l'interface d'utilisateur graphique d'Altova XMLSpy, MapForce ou StyleVision. Si vous avez rajouté des paquets de taxonomie personnalisés XBRL avec un des produits ci-dessus, le fichier est situé sous C: \Users\<username>\Documents\Altova.</p> <p>Forme : <code>--taxonomy-packages-config-file=FILE</code></p>

Exemples

Cet exemple vous montre comment exécuter un fichier d'exécution de mappage (.mfx) avec MapForce Server sur Windows. Le mappage utilisé dans cet exemple lit un fichier d'entrée, **Employees.xml**, et produit deux fichiers de sortie (**PersonList.xml** et **Contacts.xml**).

D'abord, générons le fichier d'exécution MapForce Server (.mfx), comme suit :

1. Exécutez MapForce et ouvrez le fichier de design MapForce suivant (fichier .mfd) : **C:\Users\<user>\Documents\Altova\MapForce2024\MapForceExamples\ChainedPersonList.mfd**.
2. Dans le menu **Fichier**, cliquer sur **Compiler sur le fichier d'exécution de MapForce Server**.
3. Sur invitation, enregistrer le fichier .mfx sur le répertoire **C:\temp**. Ce sera le répertoire de travail où le mappage sera exécuté par MapForce Server.

Ensuite, ouvrons une invite de ligne de commande et modifier le répertoire de travail dans **C:\temp**.

```
cd C:\temp
```

Enfin, appelez la commande pour exécuter **ChainedPersonList.mfx**. Dans cet exemple, MapForce Server est appelé en utilisant un chemin absolu. (Pour l'appeler avec un chemin relatif, ajouter le chemin de l'exécutable dans votre variable d'environnement `PATH` de votre système).

```
"C:\Program Files (x86)\Altova\MapForceServer2024\bin\MapForceServer.exe" run  
ChainedPersonList.mfx
```

Les deux fichiers de sortie (**PersonList.xml** et **Contacts.xml**) sont générés dans le répertoire de travail. De manière plus importante, ce mappage est configuré pour utiliser les chemins absolus, ce qui est pourquoi le mappage a été exécuté avec succès et ne nécessite pas que le fichier d'entrée **Employees.xml** existe dans le répertoire de travail. Le fichier **Employees.xml** existe réellement dans le dossier MapForce Examples mentionné ci-dessus et est référencé par le biais d'un chemin absolu. Pour spécifier si des chemins doivent être traités en tant que chemins absolus ou relatifs, cliquer avec la touche de droite sur la mappage dans MapForce, sélectionner **Mapping Settings**, puis sélectionner ou supprimer la case à cocher **Rendre les chemins absolus dans le code généré**. Si vous modifiez les paramètres de mappage, assurez-vous de recompiler le mappage sur .mfx. Pour plus d'informations, voir [Préparer des mappages pour l'exécution de serveur](#)⁴³.

5.6 setdeflang

Syntaxe et description

La commande `setdeflang` (la forme abrégée est `sd1`) définit la langue par défaut de MapForce Server. Les langues disponibles sont anglais (`en`), allemand (`de`), espagnol (`es`), français (`fr`) et japonais (`ja`). La commande prend un argument `LanguageCode` obligatoire.

```
mapforceserver setdeflang [options] LanguageCode
```

- L'argument `LanguageCode` est requis et définit la langue par défaut de MapForce Server. Les valeurs respectives à utiliser sont : `en`, `de`, `es`, `fr`, `ja`.
- Utiliser l'option `--h, --help` pour afficher les informations concernant la commande.

Exemples

Exemples de la commande `setdeflang (sd1)` :

```
mapforceserver sd1 de
mapforceserver setdeflang es
```

- La première commande définit le langage par défaut de MapForce Server en allemand.
- La seconde commande définit le langage par défaut de MapForce Server en espagnol.

Options

Use the `--h, --help` option to display information about the command.

5.7 verifylicense

Syntaxe et description

La commande `verifylicense` contrôle si le produit actuel est mis sous licence. De plus, l'option `--license-key` vous permet de vérifier si une clé de licence spécifique est assignée au produit.

```
mapforceserver verifylicense [options]
```

- Pour contrôler si une licence spécifique est attribuée à MapForce Server, fournir la clé de licence en tant que la valeur de l'option `--license-key`.

Pour plus de détails concernant la licence, voir la documentation LicenseServer (<https://www.altova.com/manual/fr/licenseserver/3.14/>).

Exemples

Exemple de la commande `verifylicense` :

```
mapforceserver verifylicenseserver
mapforceserver verifylicenseserver --license-key=ABCD123-ABCD123-ABCD123-ABCD123-ABCD123-ABCD123
```

- La première commande contrôle si MapForce Server est mis sous licence.
- La seconde commande contrôle si MapForce Server est mis sous licence avec la clé de licence spécifiée avec l'option `--license-key`.

Options

Options are listed in short form (if available) and long form. You can use one or two dashes for both short and long forms. An option may or may not take a value. If it takes a value, it is written like this: `--option=value`. Values can be specified without quotes except in two cases: (i) when the value string contains spaces, or (ii) when explicitly stated in the description of the option that quotes are required. If an option takes a Boolean value and no value is specified, then the option's default value is `TRUE`. Use the `--h, --help` option to display information about the command.

▼ license-key [!]

```
--l, --license-key = Value
```

Contrôle si MapForce Server est mis sous licence avec la clé de licence spécifiée en tant que la valeur de cette option.

5.8 version

Syntaxe et description

La commande `version` affiche le numéro de version de MapForce Server.

```
mapforceserver version
```

Exemple

Exemple de la commande `version` :

```
mapforceserver version
```

6 MapForce Server API

MapForce Server propose une interface de programmation d'application (API) auquel vous pouvez accéder par programme depuis votre code sur base .NET, COM, ou Java.

Pour une introduction à chaque plateforme, veuillez vous référer aux sujets suivants :

- [À propos de l'interface .NET](#) ⁸⁸
- [À propos de l'interface COM](#) ⁹⁶
- [À propos de l'interface Java](#) ¹⁰⁵

Pour une description technique de l'API, veuillez vous référer aux sujets suivants :

- [Référence API \(COM, .NET\)](#) ¹¹⁸
- [Référence API \(Java\)](#) ¹³⁶

6.1 Interface .NET

L'interface .NET est construite en tant que wrapper autour de l'interface COM. Elle est fournie en tant qu'un primary interop assembly signé par Altova et utilise l'espace de noms `Altova.MapForceServer`. Pendant l'installation, MapForce Server sera enregistré automatiquement en tant qu'objet de serveur COM, il n'y a donc aucune nécessité d'une inscription manuelle.

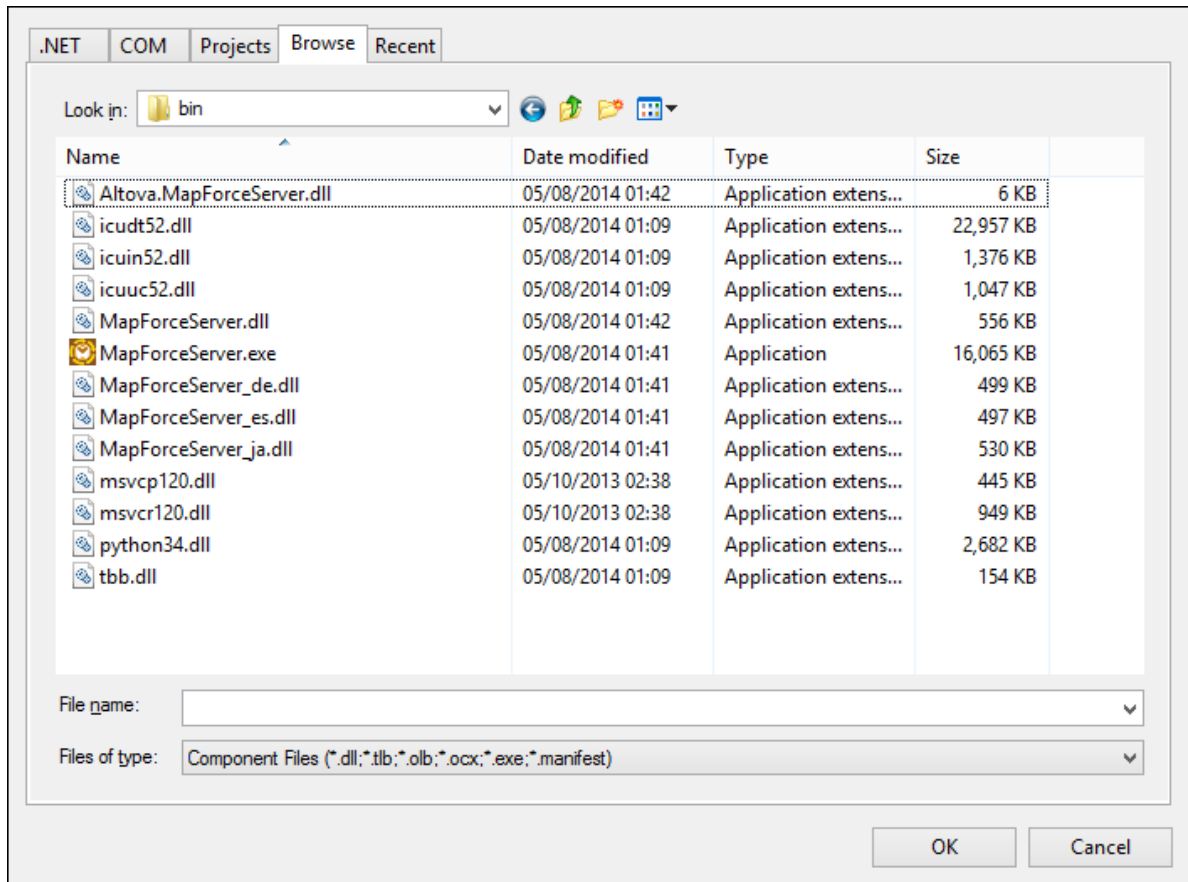
Afin de pouvoir utiliser MapForce Server dans votre projet .NET, ajouter une référence au fichier `Altova.MapForceServer.dll`, tel qu'affiché ci-dessous. L'`Altova.MapForceServer.dll` est situé dans le dossier `bin` du dossier d'installation MapForce Server. Ce fichier .dll est automatiquement ajouté au Global Assembly Cache (GAC) au cours de l'installation de MapForce.

Note : Si vous avez installé une version MapForce Server 64-bit, alors la version 32-bit de `Altova.MapForceServer.dll` sera située dans le dossier `bin\API_32bit`. De la même manière, si vous avez installé MapForce Server 32-bit, alors les fichiers de version 64-bit de `Altova.MapForceServer.dll` seront situés dans le dossier `bin\API_64bit`.

Note : avant .NET Framework 4.0, le GAC était situé dans le répertoire `%windir%\assembly`. Depuis .NET Framework 4.0, le GAC est situé dans le répertoire `%windir%\Microsoft.NET\assembly`. La partie `%windir%` représente le répertoire d'exploitation de Windows, généralement `C:\Windows`.

Pour ajouter une référence à la MapForce Server DLL dans un projet Visual Studio .NET :

1. Le projet .NET ouvert dans Visual Studio, cliquer sur **Projet | Ajouter Référence**.



2. Dans l'onglet Browse, chercher le dossier : `<MapForceServer application folder>/bin`, choisir `Altova.MapForceServer.dll` et cliquer sur OK.

Vous pouvez consulter la structure de l'assembly `Altova.MapForceServer` avec le Visual Studio Object Browser (pour afficher l'Object Browser, cliquer sur **Object Browser** dans le menu **View**).

6.1.1 Exemple C#

L'exemple suivant illustre comment exécuter un fichier d'exécution de mappage (.mfx) depuis le code C#. Sous Windows, les fichiers d'exemple sont disponibles sous le chemin suivant : **C:\Program Files\Altova\MapForceServer2024\etc\Examples**.

Conditions préalables

- MapForce Server est installé et mis sous licence
- Si vous créez un nouveau projet Visual Studio, ajoutez une référence à l'assembly de MapForce Server (voir [Interface .NET](#)⁸⁸). Vous pouvez sauter cette étape si vous exécutez l'exemple API MapForce Server existant, parce que l'exemple référence déjà l'assembly de MapForce Server.
- Sur le menu **Build** de Visual Studio, cliquer sur **Configuration Manager** et définir une plateforme de build correcte, par exemple **Debug | x86** (ou **Debug | x64**, le cas échéant). Ne pas utiliser "Any CPU" en tant que plateforme.
- Si vous avez installé MapForce Server 64-bit, alors l'application qui appelle l'API (comme l'échantillon

ci-dessous) doit aussi être construite pour la plateforme 64-bit dans Visual Studio. De même, le chemin vers le programme d'exécution de MapForce Server doit être ajusté conformément au code.

La solution d'exemple se trouve dans le répertoire "Program Files", qui nécessite des droits administratifs. Vous devez soit exécuter Visual Studio en tant qu'administrateur, soit copier la solution dans un fichier différent dans lequel vous n'avez pas besoin de droits d'administrateur.

Exécuter le fichier .mfx

Le code ci-dessous exécute trois fichiers d'exécution de serveur (.mfx). La table ci-dessous recense les fichiers d'entrée attendus par chaque fichier .mfx, et la sortie qui sera créée après l'exécution.

Fichier d'exécution (.mfx)	Entrée	Sortie
TokenizeString.mfx	AltovaTools.xml	AltovaToolsFeatures.csv
SimpleTotal.mfx	ipo.xml	String
ClassifyTemperatures.mfx	Temperatures.xml	Temperatures_out.xml

Si vous disposez de Altova MapForce, vous pouvez également consulter les mappages originaux à partir desquels les fichiers .mfx ont été compilés pour mieux les comprendre. Ils sont appelés **TokenizeString1.mfd**, **SimpleTotal.mfd** et **ClassifyTemperatures.mfd**, respectivement. Vous trouverez les mappages dans les répertoires suivants : **C:\users\<user>\Altova\MapForce2024\MapForceExamples**.

L'exemple ci-dessous procède comme suit :

- Il crée une nouvelle instance de `Altova.MapForceServer.Server`. Il s'agit de l'objet avec lequel vous travaillerez par la suite.
- Il définit un répertoire de travail dans lequel l'exécution aura lieu. Les fichiers d'entrée sont censés exister dans ce répertoire si vous vous y référez comme utilisant un chemin relatif. Des fichiers de sortie seront aussi créés dans ce répertoire.
- Il exécute le fichier **TokenizeString.mfx**. Le chemin vers ce fichier est fourni en tant qu'un argument pour la méthode `Run` (veuillez noter que le chemin est relatif au répertoire de travail qui a été défini précédemment). En cas d'exécution réussie, un fichier .csv représentant la sortie de mappage sera créé dans le répertoire de travail.
- Il exécute **SimpleTotal.mfx**. Le chemin de fichier est également relatif au répertoire de travail. Ce mappage produit une sortie de string, nous appelons donc la méthode `GetOutputParameter` pour obtenir la sortie de string.
- Il exécute **ClassifyTemperatures.mfx**. Ce mappage attend un paramètre en tant qu'entrée, qui a été fourni avec l'aide de la méthode `AddParameter`

```
namespace MapForceServerAPI_sample
{
    class Program
    {
        static void Main(string[] args)
        {
```

```

try
{
    // Create a MapForce Server object
    Altova.MapForceServer.Server objMFS = new Altova.MapForceServer.Server();

    // Set a working directory - used as a base for relative paths (you may
    need to adapt the path to the installation folder)
    objMFS.WorkingDirectory = "..\\..\\..";

    // Default path to the MapForce Server executable is the installation
    path (same dir with the MapForceServer.dll)
    // In case you moved the binaries on the disk, you need to explicitly set
    the path to the .exe file
    // objMFS.ServerPath = "C:\\Program Files (x86)\\Altova\\
    \\MapForceServer2020\\bin\\MapForceServer.exe";
    // objMFS.ServerPath = "C:\\Program Files\\Altova\\MapForceServer2020\\
    \\bin\\MapForceServer.exe";

    System.Console.WriteLine("Running " + objMFS.ProductNameAndVersion + ".
    \\n");

    // Set global resource file and configuration, if your mapping uses
    global resources
    // objMFS.SetOption( "globalresourcefile", "GlobalResources.xml" ); //
    "gr" can be used as short name for "globalresourcefile"
    // objMFS.SetOption( "globalresourceconfig", "Default" ); // "gc" can be
    used as short name for "globalresourceconfig"

    //
    -----
    // An example with input and output paths stored inside the MFX file
    System.Console.WriteLine("\\nExecuting TokenizeString.mfx...");
    if (objMFS.Run("TokenizeString.mfx"))
        System.Console.WriteLine("Successfully generated file
    'AltovaToolFeatures.csv'.");
    else
    {
        // execution failed. maybe no write permissions in working directory?
        Run this program as administrator.
        System.Console.WriteLine(objMFS.LastExecutionMessage);
    }

    //
    -----
    // An example creating a simple output so that we can retrieve the result
    explicitly
    System.Console.WriteLine("\\nExecuting SimpleTotal.mfx...");
    if (objMFS.Run("SimpleTotal.mfx"))
        System.Console.WriteLine("Mapping result is: " +
    objMFS.GetOutputParameter("total"));
    else
    {
        // execution failed (e.g. somebody deleted file ipo.xml)
        System.Console.WriteLine(objMFS.LastExecutionMessage);
    }
}

```

```

//
-----
// an example with parameterized input
// the default of 'lower = 5' gets changed to the value '10'
// mfx reads file Temperatures.xml and writes its output to
Temperatures_out.xml.
System.Console.WriteLine("\nExecuting ClassifyTemperatures.mfx with
parameter 'lower' set to '10' ...");
objMFS.AddParameter("lower", "10");
if (objMFS.Run("ClassifyTemperatures.mfx"))
    System.Console.WriteLine("File Temperatures_out.xml has been written
successfully.");
else
{
    // execution failed. maybe no write permissions in working directory?
Run this program as administrator.
    System.Console.WriteLine(objMFS.LastExecutionMessage);
}
}
catch (System.Runtime.InteropServices.COMException ex)
{
    System.Console.WriteLine("Internal Error - " + ex.Message);
}
}
}
}

```

6.1.2 Exemple Visual Basic .NET

L'exemple suivant illustre comment exécuter un fichier d'exécution de mappage (.mfx) depuis le code Visual Basic. Sous Windows, les fichiers d'exemple sont disponibles sous le chemin suivant : **C:\Program Files\Altova\MapForceServer2024\etc\Examples**.

Conditions préalables

- MapForce Server est installé et mis sous licence
- Si vous créez un nouveau projet Visual Studio, ajoutez une référence à l'assembly de MapForce Server (voir [Interface .NET](#)⁸⁸). Vous pouvez sauter cette étape si vous exécutez l'exemple API MapForce Server existant, parce que l'exemple référence déjà l'assembly de MapForce Server.
- Sur le menu **Build** de Visual Studio, cliquer sur **Configuration Manager** et définir une plateforme de build correcte, par exemple **Debug | x86** (ou **Debug | x64**, le cas échéant). Ne pas utiliser "Any CPU" en tant que plateforme.
- Si vous avez installé MapForce Server 64-bit, alors l'application qui appelle l'API (comme l'échantillon ci-dessous) doit aussi être construite pour la plateforme 64-bit dans Visual Studio. De même, le chemin vers le programme d'exécution de MapForce Server doit être ajusté conformément au code.

La solution d'exemple se trouve dans le répertoire "Program Files", qui nécessite des droits administratifs. Vous devez soit exécuter Visual Studio en tant qu'administrateur, soit copier la solution dans un fichier différent dans lequel vous n'avez pas besoin de droits d'administrateur.

Exécuter le fichier .mfx

Le code ci-dessous exécute trois fichiers d'exécution de serveur (.mfx). La table ci-dessous recense les fichiers d'entrée attendus par chaque fichier .mfx, et la sortie qui sera créée après l'exécution.

Fichier d'exécution (.mfx)	Entrée	Sortie
TokenizeString.mfx	AltovaTools.xml	AltovaToolsFeatures.csv
SimpleTotal.mfx	ipo.xml	<i>String</i>
ClassifyTemperatures.mfx	Temperatures.xml	Temperatures_out.xml

Si vous disposez de Altova MapForce, vous pouvez également consulter les mappages originaux à partir desquels les fichiers .mfx ont été compilés pour mieux les comprendre. Ils sont appelés

TokenizeString1.mfd, **SimpleTotal.mfd** et **ClassifyTemperatures.mfd**, respectivement. Vous trouverez les mappages dans les répertoires suivants : **C:\users<user>\Altova\MapForce2024\MapForceExamples**.

L'exemple ci-dessous procède comme suit :

- Il crée une nouvelle instance de `Altova.MapForceServer.Server`. Il s'agit de l'objet avec lequel vous travaillerez par la suite.
- Il définit un répertoire de travail dans lequel l'exécution aura lieu. Les fichiers d'entrée sont censés exister dans ce répertoire si vous vous y référez comme utilisant un chemin relatif. Des fichiers de sortie seront aussi créés dans ce répertoire.
- Il exécute le fichier **TokenizeString.mfx**. Le chemin vers ce fichier est fourni en tant qu'un argument pour la méthode `Run` (veuillez noter que le chemin est relatif au répertoire de travail qui a été défini précédemment). En cas d'exécution réussie, un fichier .csv représentant la sortie de mappage sera créé dans le répertoire de travail.
- Il exécute **SimpleTotal.mfx**. Le chemin de fichier est également relatif au répertoire de travail. Ce mappage produit une sortie de string, nous appelons donc la méthode `GetOutputParameter` pour obtenir la sortie de string.
- Il exécute **ClassifyTemperatures.mfx**. Ce mappage attend un paramètre en tant qu'entrée, qui a été fourni avec l'aide de la méthode `AddParameter`

```
Option Explicit On

Module Program

    Sub Main()

        Try
            'Create a MapForce Server object;
            Dim objMFS As Altova.MapForceServer.Server = New Altova.MapForceServer.Server

            'Set a working directory - used as a base for relative paths for the MapForce
            server execution (.mfx) file.
            objMFS.WorkingDirectory = "C:\Program Files (x86)
            \Altova\MapForceServer2020\etc\Examples"
```

```

objMFS.WorkingDirectory = "..\..\.."

'Default path to the MapForce Server executable is the installation path
(same dir with the MapForceServer.dll)
'In case you moved the binaries on the disk, you need to explicitly set the
path to the .exe file
objMFS.ServerPath = "C:\Program Files (x86)
\Altova\MapForceServer2024\bin\MapForceServer.exe"
objMFS.ServerPath = "C:\Program
Files\Altova\MapForceServer2024\bin\MapForceServer.exe"

'Set global resource file and configuration, if your mapping uses global
resources
objMFS.SetOption("globalresourcefile", "GlobalResources.xml") 'gr" can be
used as short name for "globalresourcefile"
objMFS.SetOption("globalresourceconfig", "Config2") 'gc" can be used as
short name for "globalresourceconfig"

'-----
'An example with input and output paths stored inside the MFX file
System.Console.WriteLine(vbCrLf & "Executing TokenizeString.mfx...")
If (objMFS.Run("TokenizeString.mfx")) Then
    System.Console.WriteLine("Successfully generated file
'AltovaToolFeatures.csv'.")
Else
    'execution failed. maybe no write permissions in working directory? Run
this program as administrator.
    System.Console.WriteLine(objMFS.LastExecutionMessage)
End If

'-----
'An example creating a simple output so that we can retrieve the result
explicitly
System.Console.WriteLine(vbCrLf & "Executing SimpleTotal.mfx...")
If (objMFS.Run("SimpleTotal.mfx")) Then
    System.Console.WriteLine("Mapping result is: " &
objMFS.GetOutputParameter("total"))
Else
    'execution failed (e.g. somebody deleted file ipo.xml)
    System.Console.WriteLine(objMFS.LastExecutionMessage)
End If

'-----
'an example with parameterized input
' the default of 'lower=5' gets changed to the value '10'
' mfx reads file Temperatures.xml and writes its output to
Temperatures_out.xml.
System.Console.WriteLine(vbCrLf & "Executing ClassifyTemperatures.mfx with
parameter 'lower' set to '10' ...")

```

```
objMFS.AddParameter("lower", "10")
  If (objMFS.Run("ClassifyTemperatures.mfx")) Then
    System.Console.WriteLine("File Temperatures_out.xml has been written
successfully.")
  Else
    'execution failed. maybe no write permissions in working directory? Run
this program as administrator.
    System.Console.WriteLine(objMFS.LastExecutionMessage)
  End If

Catch ex As Exception
  System.Console.WriteLine("Internal Error - " & ex.Message())
End Try

End Sub

End Module
```

6.2 Interface COM

MapForce Server est enregistré automatiquement en tant qu'objet de serveur COM pendant l'installation. Pour vérifier si l'enregistrement a réussi, ouvrir l'éditeur de registre (par exemple, en saisissant la commande `regedit.exe` dans la ligne de commande). Si l'enregistrement a réussi, le Registre contiendra les classes suivantes :

- **MapForce.Server** (pour 32-bit MapForce Server)
- **MapForce_x64.Server** (pour 64-bit MapForce Server)

Ces classes se trouvent sous `HKEY_LOCAL_MACHINE\SOFTWARE\Classes`.

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. Si vous souhaitez changer l'emplacement du paquet d'installation MapForce Server, il vaut mieux désinstaller MapForce Server et puis le réinstaller à l'emplacement requis. De cette manière, l'enregistrement ou le désenregistrement sont effectués par le processus de l'installateur.

6.2.1 Exemple C++

L'exemple suivant illustre comment exécuter un fichier d'exécution de mappage (.mfx) depuis le code C++. Sous Windows, les fichiers d'exemple sont disponibles sous le chemin suivant : **C:\Program Files\Altova\MapForceServer2024\etc\Examples**.

Conditions préalables

Avant d'exécuter le code ci-dessous, assurez-vous que les conditions préalables suivantes sont correctes :

- MapForce Server est installé et mis sous licence
- MapForce Server est disponible en tant qu'objet de serveur COM (normalement, ce processus a lieu automatiquement pendant l'installation MapForce Server ; pour vérifier si l'enregistrement a bien réussi, voir [Interface COM](#)⁹⁶).

Exécuter le fichier .mfx

Le code ci-dessous exécute trois fichiers d'exécution de serveur (.mfx). La table ci-dessous recense les fichiers d'entrée attendus par chaque fichier .mfx, et la sortie qui sera créée après l'exécution.

Fichier d'exécution (.mfx)	Entrée	Sortie
TokenizeString.mfx	AltovaTools.xml	AltovaToolsFeatures.csv
SimpleTotal.mfx	ipo.xml	<i>String</i>
ClassifyTemperatures.mfx	Temperatures.xml	Temperatures_out.xml

Si vous disposez de Altova MapForce, vous pouvez également consulter les mappages originaux à partir desquels les fichiers .mfx ont été compilés pour mieux les comprendre. Ils sont appelés **TokenizeString1.mfd**, **SimpleTotal.mfd** et **ClassifyTemperatures.mfd**, respectivement. Vous trouverez les

mappages dans les répertoires suivants : **C:\users\<user>\Altova\MapForce2024\MapForceExamples**.

L'exemple ci-dessous procède comme suit :

- Il crée une nouvelle instance de `Altova.MapForceServer.Server`. Il s'agit de l'objet avec lequel vous travaillerez par la suite.
- Il définit un répertoire de travail dans lequel l'exécution aura lieu. Les fichiers d'entrée sont censés exister dans ce répertoire si vous vous y référez comme utilisant un chemin relatif. Des fichiers de sortie seront aussi créés dans ce répertoire.
- Il exécute le fichier **TokenizeString.mfx**. Le chemin vers ce fichier est fourni en tant qu'un argument pour la méthode `Run` (veuillez noter que le chemin est relatif au répertoire de travail qui a été défini précédemment). En cas d'exécution réussie, un fichier `.csv` représentant la sortie de mappage sera créé dans le répertoire de travail.
- Il exécute **SimpleTotal.mfx**. Le chemin de fichier est également relatif au répertoire de travail. Ce mappage produit une sortie de string, nous appelons donc la méthode `GetOutputParameter` pour obtenir la sortie de string.
- Il exécute **ClassifyTemperatures.mfx**. Ce mappage attend un paramètre en tant qu'entrée, qui a été fourni avec l'aide de la méthode `AddParameter`

```
// MapForceServerAPI_sample.cpp : Defines the entry point for the console application.
//
#include <iostream>
#include "atlbase.h"

#ifdef _WIN64
// 32-bit MapForce Server
#import "progid:MapForce.Server"
#else
// 64-bit MapForce Server
#import "progid:MapForce_x64.Server"
#endif

int _tmain(int argc, _TCHAR* argv[])
{
    CoInitialize( NULL );

    try
    {
        // Create a MapForce Server object
        MapForceServerLib::IServerPtr pMFS;
        CoCreateInstance( __uuidof( MapForceServerLib::Server ), NULL, CLSCTX_ALL,
            __uuidof( MapForceServerLib::IServer ), reinterpret_cast< void** >( &pMFS ) );

        //Set a working directory - used as a base for relative paths (you may need to
        adapt the path to the installation folder)
        pMFS->WorkingDirectory = ".."; // this is relative to this applications' working
        directory (the project folder)

        // Default path to the MapForce Server executable is the installation path (same
        dir with the MapForceServer.dll)
        // In case you moved the binaries on the disk, you need to explicitly set the path
    }
}
```

```

to the .exe file
    // pMFS.ServerPath = "C:\\Program Files (x86)\\Altova\\MapForceServer2024\\bin\\
\\MapForceServer.exe";
    // pMFS.ServerPath = "C:\\Program Files\\Altova\\MapForceServer2024\\bin\\
\\MapForceServer.exe";

    //Set global resource file and configuration, if your mapping uses global resources
    //pMFS->SetOption( "globalresourcefile", "GlobalResources.xml" ); // "gr" can be
used as short name for "globalresourcefile"
    //pMFS->SetOption( "globalresourceconfig", "Default" ); // "gc" can be used as
short name for "globalresourceconfig"

    //
-----
    // An example with input and output paths stored inside the MFX file
std::cout << "\nExecuting TokenizeString.mfx..." << std::endl;
if ( pMFS->Run( "TokenizeString.mfx" ) == VARIANT_TRUE )
    std::cout << "Successfully generated file 'AltovaToolFeatures.csv'." <<
std::endl;
else
{
    // execution failed. maybe no write permissions in working directory? Run this
program as administrator.
    std::cout << pMFS->LastExecutionMessage << std::endl;
}

//
-----
    // An example creating a simple output so that we can retrieve the result
explicitly
std::cout << "\nExecuting SimpleTotal.mfx..." << std::endl;
if ( pMFS->Run( "SimpleTotal.mfx" ) )
    std::cout << "Mapping result is: " + pMFS->GetOutputParameter( "total" ) <<
std::endl;
else
{
    // execution failed (e.g. somebody deleted file ipo.xml)
    std::cout << pMFS->LastExecutionMessage << std::endl;
}

//
-----
    // an example with parameterized input
    // the default of 'lower = 5' gets changed to the value '10'
    // mfx reads file Temperatures.xml and writes its output to Temperatures_out.xml.
std::cout << "\nExecuting ClassifyTemperatures.mfx with parameter 'lower' set to
'10' ..." << std::endl;
pMFS->AddParameter("lower", "10");
if ( pMFS->Run( "ClassifyTemperatures.mfx" ) )
    std::cout << "File Temperatures_out.xml has been written successfully." <<
std::endl;
else
{

```

```

        // execution failed. maybe no write permissions in working directory? Run this
        program as administrator.
        std::cout << pMFS->LastExecutionMessage << std::endl;
    }
}
catch (_com_error& err )
{
    BSTR bstrMessage;
    (err).ErrorInfo()->GetDescription( &bstrMessage );
    std::cout << "Exception occurred: " <<
_com_util::ConvertBSTRToString( bstrMessage ) << std::endl;
}

CoUninitialize();

return 0;
}

```

6.2.2 Exemple VBScript

L'exemple suivant illustre comment exécuter un fichier d'exécution de mappage (.mfx) depuis le code VBScript. Sous Windows, les fichiers d'exemple sont disponibles sous le chemin suivant : **C:\Program Files\Altova\MapForceServer2024\etc\Examples**.

Avant de tester cet exemple, assurez-vous que les conditions préalables suivantes sont correctes :

- MapForce Server est installé et mis sous licence
- MapForce Server est disponible en tant qu'objet de serveur COM (normalement, ce processus a lieu automatiquement pendant l'installation MapForce Server ; pour vérifier si l'enregistrement a bien réussi, voir [Interface COM](#)⁹⁶).

```

Option Explicit

REM This script produces extensive output.
REM It is best called from a cmd.exe console with "cscript MapForceServerAPI_sample.vbs"

'Create the MapForce Server object
Dim objMFS
' Since we load a COM-DLL we need care about the process architecture
On Error Resume Next ' ignore any COM errors avoiding uncontrolled script termination
Dim WshShell
Dim WshProcEnv
Set WshShell = CreateObject("WScript.Shell")
Set WshProcEnv = WshShell.Environment("Process")
Dim process_architecture
process_architecture= WshProcEnv("PROCESSOR_ARCHITECTURE")
If process_architecture = "x86" Then
    Set objMFS = WScript.GetObject( "", "MapForce.Server" )
    If Err.Number <> 0 then
        WScript.Echo("You are running in a 32-bit process but MapForce Server COM-API 32-
bit seems not to be installed on your system.")
    End If
End If

```

```

WScript.Quit -1
End If
Else
Set objMFS = WScript.GetObject( "", "MapForce_x64.Server" )
If Err.Number <> 0 then
WScript.Echo("You are running in a 64-bit process but MapForce Server COM-API 64-
bit seems not to be installed on your system.")
WScript.Echo("If you have installed 32-bit MapForce Server consider calling your
script from the 32-bit console 'C:\Windows\SysWOW64\cmd.exe.'")
WScript.Quit -1
End If
End If
On Error Goto 0      ' re-enable default error promotion

' Set a working directory - used as a base for relative paths (you may need to adapt the
path to the installation folder)
REM objMFS.WorkingDirectory = "C:\Program Files (x86)
\Altova\MapForceServer2020\etc\Examples"
Dim currDir
Dim fso
Set fso = CreateObject("Scripting.FileSystemObject")
currDir = fso.GetParentFolderName(Wscript.ScriptFullName)
'set working folder to parent of this script
objMFS.WorkingDirectory = fso.GetParentFolderName( currDir )

'Default path to the MapForce Server executable is the installation path (same dir with
the MapForceServer.dll)
'In case you moved the binaries on the disk, you need to explicitly set the path to the
.exe file
objMFS.ServerPath = "C:\Program Files (x86)
\Altova\MapForceServer2024\bin\MapForceServer.exe"

'Set global resource file and configuration, if your mapping uses global resources
'Call objMFS.SetOption("globalresourcefile", "GlobalResources.xml") 'gr" can be used as
short name for "globalresourcefile"
'Call objMFS.SetOption("globalresourceconfig", "Config2") 'gc" can be used as short name
for "globalresourceconfig"

WScript.Echo( "Running " & objMFS.ProductNameAndVersion & vbCrlf )

' The Run method will return 'True' if the execution of the mfx file was successful
otherwise 'False'.
' In the case of fundamental errors like termination of the server process a COM error
will be raised which
' can be handled using the VBScript Err object.
On Error Resume Next      ' ignore any COM errors avoiding uncontrolled script termination
Err.Clear

REM -----
REM run an example with input and output paths stored inside the MFX file
' the path to the mfx file can be absolute or relative to the working directory
' depends on existence of file AltovaTools.xml in working directory

```

```

' creates output file AltovaToolFeatures.csv in working directory
WScript.Echo( "Processing TokenizeString.mfx..." )
If ( objMFS.Run( "TokenizeString.mfx" ) ) Then
    'WScript.Echo( objMFS.LastExecutionMessage ) ' execution log
    WScript.Echo( "Successfully generated file AltovaToolFeatures.csv." )
Else
    'execution failed (e.g. somebody deleted file AltovaTools.xml)
    WScript.Echo( objMFS.LastExecutionMessage )
End If
WScript.Echo( "" )
' handle COM errors
If Err.Number <> 0 Then
    WScript.Echo( "Internal error - " & Err.Description )
    WScript.Quit -1
End If

REM
-----
REM this is an example creating a simple output so that we can retrieve the result
explicitly
' depends on input XML file ipo.xml
WScript.Echo( "Processing SimpleTotal.mfx..." )
If ( objMFS.Run( "SimpleTotal.mfx" ) ) Then
    'WScript.Echo( objMFS.LastExecutionMessage )
    WScript.Echo( "Mapping result is: " & objMFS.GetOutputParameter("total") )
Else
    'execution failed (e.g. somebody deleted file ipo.xml)
    WScript.Echo( objMFS.LastExecutionMessage )
End If
WScript.Echo( "" )
' handle COM errors
If Err.Number <> 0 Then
    WScript.Echo( "Internal error - " & Err.Description )
    WScript.Quit -1
End If

REM -----
REM This is an example with parameterized input
' the default of 'lower=5' gets changed to the value '10'
' mfx reads file Temperatures.xml and writes its output to Temperatures_out.xml.
WScript.Echo( "Processing ClassifyTemperatures.mfx with parameter 'lower' set to '10'
..." )
call objMFS.AddParameter( "lower", "10" )
If ( objMFS.Run( "ClassifyTemperatures.mfx" ) ) Then
    'WScript.Echo( objMFS.LastExecutionMessage )
    WScript.Echo( "File Temperatures_out.xml has been written successfully." )
Else
    'execution failed (e.g. somebody locks file Temperatures_out.xml)
    WScript.Echo( objMFS.LastExecutionMessage )
End If
call objMFS.ClearParameterList()
WScript.Echo( "" )

```

```
' handle COM errors
If Err.Number <> 0 Then
    WScript.Echo("Internal error - " & Err.Description )
    WScript.Quit -1
End If

On Error Goto 0      ' re-enable default error promotion
```

6.2.3 Exemple VBA

Microsoft Visual Basic for Applications (VBA) est principalement utilisé pour automatiser les tâches dans Microsoft Office. Néanmoins, il est aussi possible d'appeler l'API MapForce Server depuis VBA et d'exécuter les mappages. Les instructions suivantes ont été testées sur MapForce Server et VBA pour Microsoft Office 2013. Les instructions peuvent varier si vous utilisez un autre environnement de développement VBA.

Exigences préalables

Avant de pouvoir appeler les fonctions API MapForce Server depuis votre projet VBA, veuillez noter les exigences suivantes :

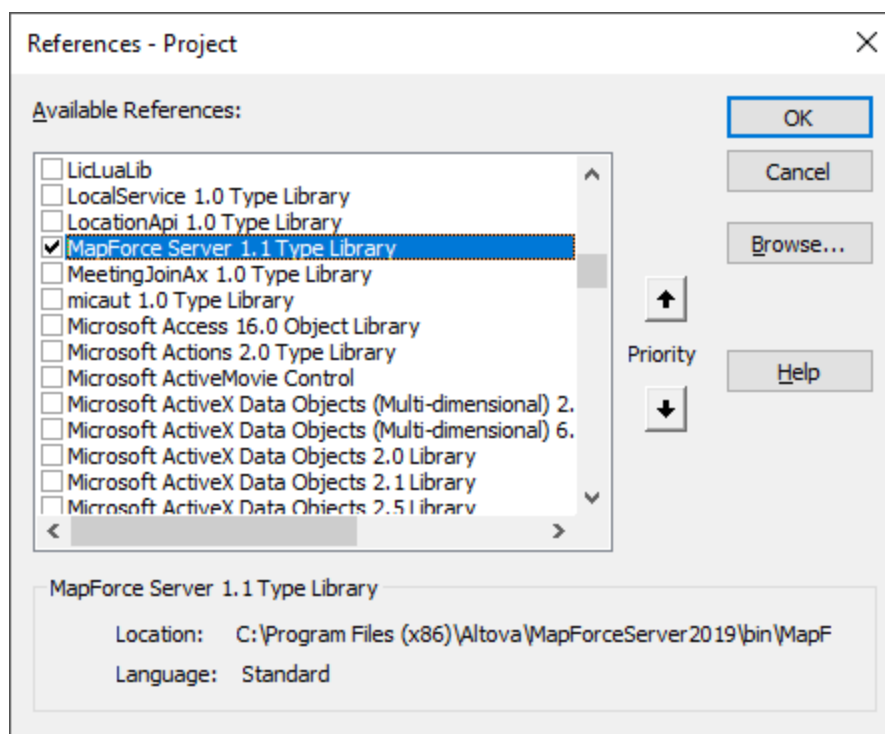
1. Microsoft Office et MapForce Server doivent être installés sur le même appareil.
2. L'architecture de MapForce Server (32-bit ou 64-bit) doit correspondre à celle de Microsoft Office. Par exemple, si vous exécutez VBA sur Microsoft Office 32-bit, veuillez vous assurer que vous utilisez MapForce Server 32-bit. Pour savoir si votre produit Office est exécuté sur 64-bit, cliquez sur l'onglet **Fichier**, cliquez sur **Account**, puis sur "About Excel" (ou "About Word").
3. La bibliothèque MapForce Server doit être référencée depuis votre projet VBA (voir instructions ci-dessous).

Comment ajouter une référence à la Bibliothèque MapForce Server Library depuis votre projet VBA

1. Dans un document Microsoft Office activé pour macro (.docm, .xlsm), sur l'onglet **Developer**, cliquez sur **Visual Basic**.

Par défaut, l'onglet **Developer** n'est pas activé dans Microsoft Office. Pour l'activer dans un programme d'Office 2013, cliquez avec la touche de droite dans le ruban et sélectionnez **Customize the Ribbon** depuis le menu contextuel. Ensuite, dans le dialogue Options, choisissez la case **Developer** sous "Main Tabs".

2. Dans l'environnement de développement VBA, dans le menu **Tools**, cliquez sur **References**.



3. Cliquer sur pour choisir la **MapForce Server Type Library**.

Comment appeler l'API de MapForce Server

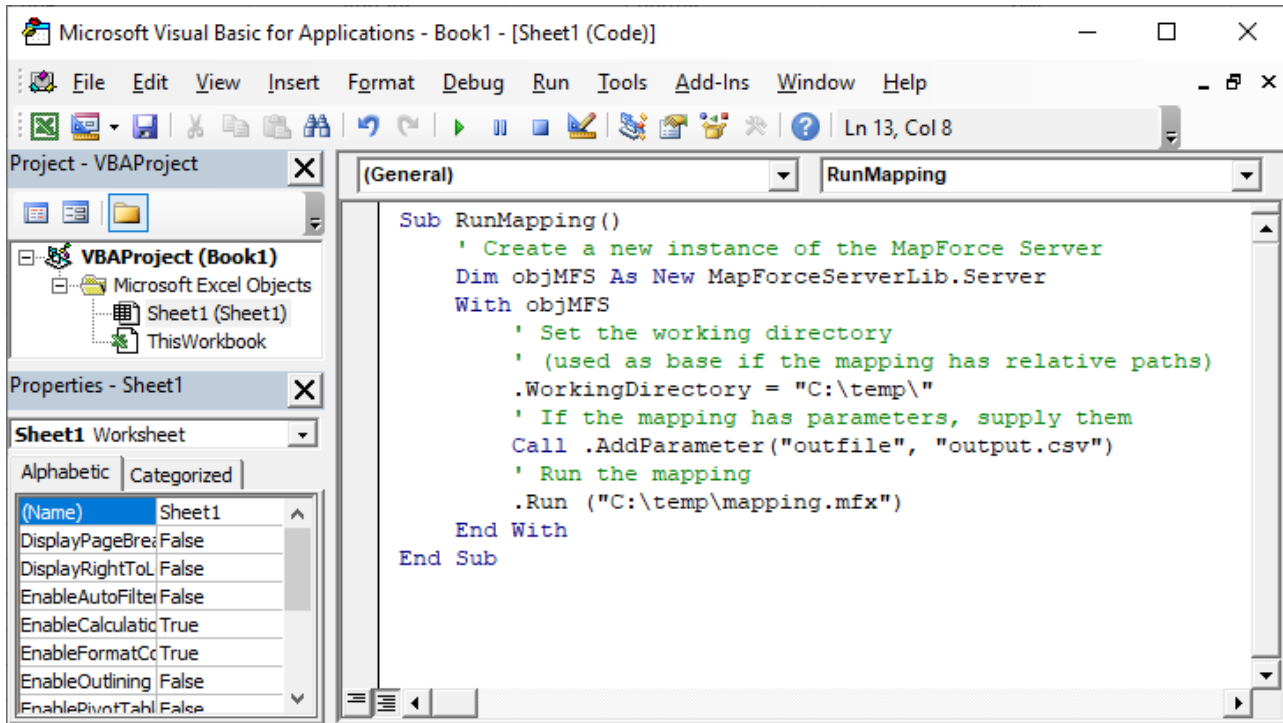
Une fois que vous avez ajouté une référence à la MapForce Server Library dans votre projet VBA, vous pouvez saisir le code VBA dans la fenêtre de l'éditeur de code. Par exemple, l'échantillon de code suivant appelle MapForce Server et exécute un fichier d'exécution de mappage (**mapping.mfx**) qui prend un paramètre d'entrée appelé "outfile" possédant la valeur "output.csv".

```

Sub RunMapping()
    ' Create a new instance of the MapForce Server
    Dim objMFS As New MapForceServerLib.Server
    With objMFS
        ' Set the working directory
        ' (used as base if the mapping has relative paths)
        .WorkingDirectory = "C:\temp\"
        ' If the mapping has parameters, supply them
        Call .AddParameter("outfile", "output.csv")
        ' Run the mapping
        .Run ("C:\temp\mapping.mfx")
    End With
End Sub

```

Appuyer sur **F5** pour déboguer le code VBA et exécuter le mappage.



6.3 Interface Java

L'API consiste en un fichier JAR (`MapForceServer.jar`) et en un fichier JNI (`MapForceServer.dll`). Ces deux fichiers, de même que d'autres fichiers API associés, sont disponibles dans le dossier `bin` du dossier d'installation MapForce Server. Vous pouvez soit référencer le fichier depuis leur emplacement d'origine soit les copier à un autre emplacement si cela convient à votre configuration de projet. (Sur les systèmes Windows, vous aurez besoin de droits administratifs pour exécuter le programme depuis son emplacement original.)

Note : Si vous avez installé MapForce Server 64-bit, alors les fichiers de version 32-bit de `MapForceServer.jar` et (`MapForceServer.dll` seront situés dans le dossier `bin\API_32bit` du dossier d'installation MapForce Server . Vous allez avoir besoin de ces fichiers si vous utilisez une version Java 32-bit. De la même manière, si vous avez installé MapForce Server 32-bit, alors les fichiers de version 64-bit de `MapForceServer.jar` et (`MapForceServer.dll` seront situés dans le dossier `bin\API_64bit`. Vous allez avoir besoin de l'utilisation de ces fichiers si vous utilisez une version Java 64-bit.

Pour accéder à l'API MapForce Server depuis le code Java, ajouter les références suivantes au fichier `.classpath` de votre projet Java.

<code>MapForceServer.jar</code>	La bibliothèque qui communique avec MapForce Server
<code>MapForceServer_JavaDoc.zip</code>	Documentation de l'API MapForce Server

En plus, les besoins `java.library.path` doivent inclure le dossier où le fichier bibliothèque JNI (`MapForceServer.dll`) est situé.

Si vous déployer votre projet à un serveur d'application, assurez-vous que `MapForceServer.jar` et `MapForceServer.dll` sont correctement configurés avec Java sur l'appareil de serveur.

Pour un exemple comment utiliser les fichiers bibliothèque d'API, voir l'exemple de fichier batch `buildAndRun.bat` (*listed below*), qui est situé dans le dossier `etc\Examples\Java` du dossier d'installation MapForce Server.

Construire et exécuter un programme Java pour utiliser l'API

Pour voir comment construire et exécuter un programme Java qui utilise l'API MapForce Server, voir l'exemple de fichier batch `buildAndRun.bat`. Vous pouvez réutiliser ce fichier à exécuter vos propres programmes Java en le modifiant, les cas échéant.

Démarrer le fichier dans une interface de ligne de commande avec la commande suivante :

```
buildAndRun "path_to_Java_bin_folder"
```

Note : Pour vérifier si Java est dans votre classpath, vous pouvez exécuter la commande `java --version`. Si Java n'est pas dans votre classpath, alors vous devez fournir le chemin comme paramètre de la commande `buildAndRun`. Si le chemin contient des espaces, alors utiliser des guillemets autour du chemin.

☐ Liste de `buildAndRun.bat`

```
@echo off
```

```
if %1.==. goto error

REM The location of the JAVA API binaries, the JAR file and the JNI library.
REM Adapt to your needs.
SETLOCAL
Set JavaAPIBinPath=%PROGRAMFILES%\Altova\MapForceServer2024\bin

REM Compile sample java
REM The -cp option (classpath) needs to point to the installed jar file (here, in its
original location)
REM "Program.java" is the Java program you want to compile
%1\javac.exe -cp "%JavaAPIBinPath%\MapForceServer.jar" Program.java

REM Run sample java
REM The -cp option (classpath) needs to point to the MapForceServer.jar file
REM The java.library.path needs to include the folder where the JNI library
MapForceServer.dll is located.
%1\java.exe -cp "%JavaAPIBinPath%\MapForceServer.jar;." -Djava.library.path="%
JavaAPIBinPath%" Program

@echo off
goto end

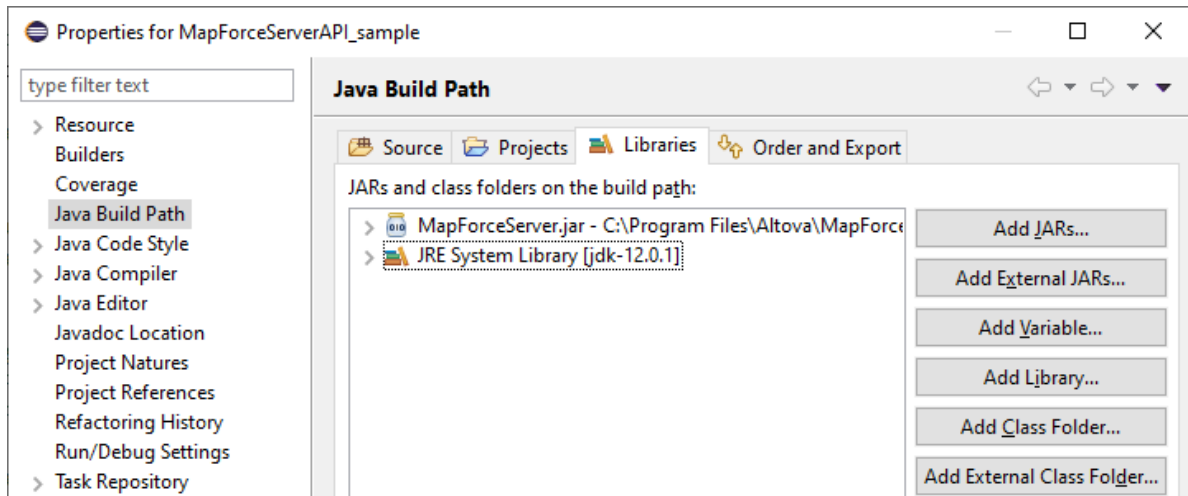
:error
echo Usage: buildAndRun "<path_to_java_bin_folder>"

:end
```

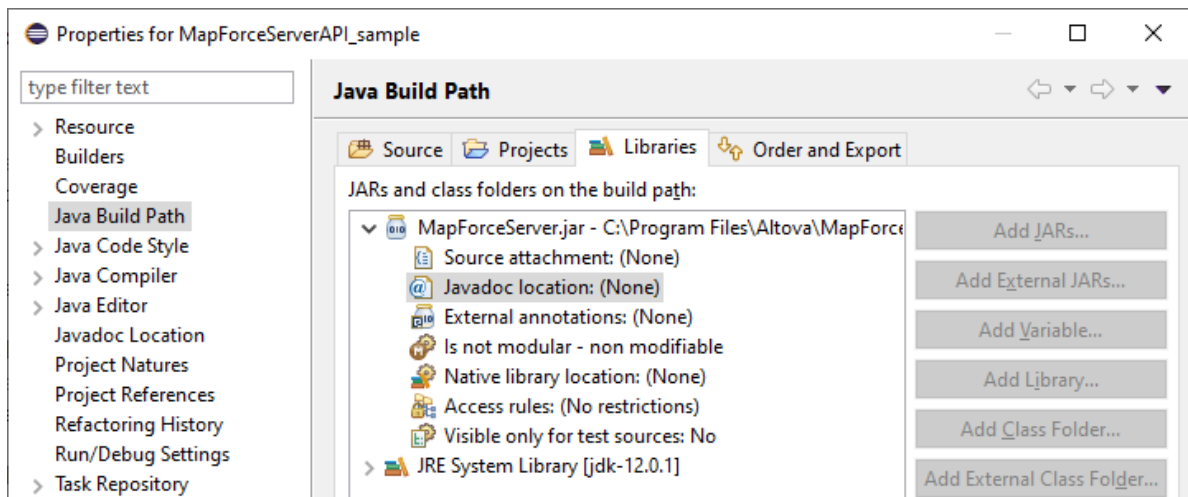
Ajouter des bibliothèques dans Eclipse

Si vous utilisez Eclipse en tant qu'environnement de développement Java, vous pouvez ajouter la référence de bibliothèque requise dans le CLASSPATH en éditant les propriétés du projet Java, comme indiqué ci-dessous. En option, vous pouvez aussi joindre une documentation en format JavaDoc à la bibliothèque .jar. Vous trouverez le JavaDoc dans le dossier **bin** du dossier d'installation MapForce Server ; les instructions ci-dessous illustrent comment rendre la documentation JavaDoc visible depuis Eclipse.

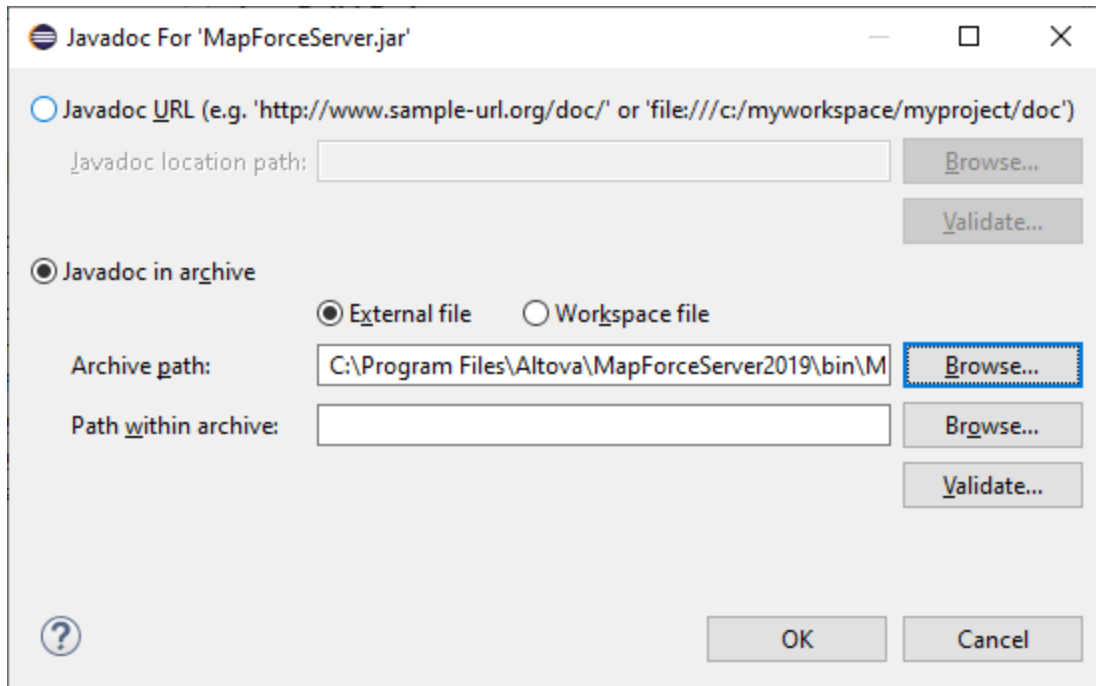
1. Le projet ouvert dans Eclipse, sur le menu **Project**, cliquez sur **Properties**.
2. Cliquer sur **Java Build Path**.
3. Dans l'onglet **Bibliothèques**, cliquez sur **Add External JARs**, puis chercher sur le fichier `MapForceServer.jar` situé dans le dossier d'installation de MapForce Server.



4. En option, pour ajouter l'archive JavaDoc, agrandir l'enregistrement **MapForceServer.jar**, puis double-cliquer sur l'emplacement **Javadoc location** : enregistrement (**None**).



5. S'assurer que les options **Javadoc in archive** et **External file** sont sélectionnées, puis chercher le fichier `MapForceServer_JavaDoc.zip` situé dans le dossier d'installation MapForce Server.



6. Cliquer sur OK.

Ci-dessous, vous trouvez un exemple de quoi ressemblerait le fichier Eclipse .classpath si vous référencez les fichiers depuis le dossier d'installation original, sur un système d'exploitation 64-bit exécutant un MapForce Server 64-bit (les lignes pertinentes sont marquées en jaune) :

```
<?xml version="1.0" encoding="UTF-8"?>
<classpath>
  <classpathentry kind="src" path=""/>
  <classpathentry kind="lib" path="C:/Program
Files/Altova/MapForceServer2024/bin/MapForceServer.jar">
    <attributes>
      <attribute name="javadoc_location" value="jar:file:/C:/Program%
20Files/Altova/MapForceServer2024/bin/MapForceServer_JavaDoc.zip!/">
    </attributes>
  </classpathentry>
  <classpathentry kind="con" path="org.eclipse.jdt.launching.JRE_CONTAINER">
    <attributes>
      <attribute name="module" value="true"/>
    </attributes>
  </classpathentry>
  <classpathentry kind="output" path=""/>
</classpath>
```

6.3.1 Exemple Java

L'exemple suivant illustre comment exécuter un fichier d'exécution de mappage (.mfx) depuis le code Java. Sur Windows, tous les exemples de fichiers sont disponibles sous le chemin suivant : **C:\Program Files\Altova\MapForceServer2024\etc\Examples**. Ils fonctionnent comme suit :

- **TokenizeString.mfx** - un fichier d'exécution de MapForce Server qui sera exécuté par le programme Java.
- **AltovaTools.xml** - un fichier XML qui fournit des données d'entrée au programme de mappage.

Sur les plateformes Linux et Mac, aucun exemple n'est préinstallé ; néanmoins, vous pouvez préparer un fichier .mfx exécutable comme celui exécuté dans cet exemple comme suit :

1. Ouvrir le fichier de design de mappage désiré (.mfd) avec MapForce sur Windows.
2. Dans le menu **Fichier**, sélectionner **Paramètres de mappage**, et décocher la case **Rendre les chemins absolus dans le code généré** en cas de sélection.
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. Se référer à la documentation MapForce pour plus d'informations concernant la gestion avec des chemins relatifs et absolus tout en concevant des mappages
4. Dans le menu **Fichier**, Choisir sur **Compiler vers un fichier d'exécution (.mfx) de MapForce Server**. Cela génère le fichier .mfx que vous allez exécuter par la suite avec MapForce Server, comme indiqué dans l'extrait de code suivant.

Prérequis

Avant de tester ce code ci-dessous, assurez-vous que les exigences suivantes sont correctes :

- MapForce Server est installé et mis sous licence
- Le Java CLASSPATH comprend une référence à la bibliothèque **MapForceServer.jar** (pour un exemple, voir [À propos de l'interface Java](#)¹⁰⁵).

Si vous utilisez un fichier .mfx personnalisé comme indiqué ci-dessus, il peut y avoir d'autres exigences préalables, selon le type de données traitées par le mappage. Pour plus d'informations, voir [Préparer les mappages pour l'exécution de serveur](#)⁴³.

Sur Windows, le projet d'exemple Java se trouve dans le répertoire "Program Files", qui nécessite des droits administratifs. Vous devez soit exécuter votre environnement de développement Java (par exemple, Eclipse) en tant qu'administrateur, ou copier l'exemple dans un dossier différent où vous ne nécessitez pas de droits d'administrateur.

Exécuter le programme Java

L'extrait de code ci-dessous crée tout d'abord un objet MapForce Server. Ensuite, il définit le répertoire de travail dans lequel l'application devra chercher des fichiers qui agissent en tant qu'entrée dans le mappage, et dans lequel il doit générer les fichiers de sortie de mappage. Comme mentionné ci-dessus, le fichier de

mappage d'exemple lit des données provenant d'un fichier XML de source, veuillez donc vous assurer que le fichier XML et son schéma existent dans le répertoire de travail.

La méthode `setServerPath` spécifie le chemin de l'exécutable MapForce Server. Par exemple, sur Ubuntu, ceci serait `/opt/Altova/MapForceServer2024/bin/mapforceserver`. Vous pouvez omettre le `setServerPath` si vous n'avez pas déplacé le **MapForceServer.jar** depuis son emplacement par défaut.

Enfin, la méthode `run` exécute un mappage (fichier .mfx) qui a été compilé avec MapForce (dans cet exemple, **TokenizeString.mfx**). En cas de succès, le programme ci-dessous génère un fichier CSV et XML dans le répertoire de travail, de même qu'un texte de sortie sur la ligne de commande. En cas d'erreur, le programme tente d'imprimer le dernier message d'exécution généré par MapForce Server.

```
import com.altova.mapforceserver.MapForceServer;

public class Program {

    public static void main(String[] args)
    {
        MapForceServer objMFS;
        try
        {
            // set up the server
            objMFS = new MapForceServer();

            // The default location of the server is the directory that the java native
            library is in.
            // With the following line you could select a different server binary.
            // objMFS.setServerPath(strServerPath);

            // The sample data is located in the parent folder of the Java sample code
            objMFS.setWorkingDirectory("..");

            System.out.println("Running " + objMFS.getProductNameAndVersion());

            //Set global resource file and configuration, if your mapping uses global
            resources
            //objMFS.setOption( "globalresourcefile", "GlobalResources.xml" ); // "gr"
            can be used as short name for "globalresourcefile"
            //objMFS.setOption( "globalresourceconfig", "Default" ); // "gc" can be used
            as short name for "globalresourceconfig"

            //
            -----
            // An example with input and output paths stored inside the MFX file
            System.out.println("\nExecuting TokenizeString.mfx...");
            if (objMFS.run("TokenizeString.mfx"))
                System.out.println("Success: " + objMFS.getLastExecutionMessage());
            else
                System.out.println("Unsuccessful: " +
            objMFS.getLastExecutionMessage());

            //
            -----
        }
    }
}
```

```
// An example creating a simple output so that we can retrieve the result
explicitly
    System.out.println("\nExecuting TokenizeString.mfx...");
    if (objMFS.run("SimpleTotal.mfx"))
        System.out.println("Mapping result is: " +
objMFS.getOutputParameter("total"));
    else
    {
        // execution failed (e.g. somebody deleted file ipo.xml)
        System.out.println(objMFS.getLastExecutionMessage());
    }
//
-----

// An example with parameterized input
// the default of 'lower = 5' gets changed to the value '10'
// mfx reads file Temperatures.xml and writes its output to
Temperatures_out.xml.
    System.out.println("\nExecuting ClassifyTemperatures.mfx with parameter
'lower' set to '10' ...");
    objMFS.addParameter("lower", "10");
    if (objMFS.run("ClassifyTemperatures.mfx"))
        System.out.println("File Temperatures_out.xml has been written
successfully.");
    else
    {
        // execution failed. maybe no write permissions in working directory? Run
this program as administrator.
        System.out.println(objMFS.getLastExecutionMessage());
    }

    // You can stop the server explicitly by invoking the 'stop' method if you
don't want to let the garbage collector decide.
    objMFS.stop();
}

catch (Exception e)
{
    System.out.println("ERROR: " + e.getMessage());
}
}
}
```

6.4 Exemple : Exécuter un mappage avec des paramètres

Cet exemple vous montre comment compiler un mappage MapForce vers un fichier d'exécution MapForce Server (.mfx) et l'exécuter depuis l'API MapForce. L'exemple illustre spécifiquement le scénario selon lequel le mappage prend le nom de fichier d'entrée en tant que paramètre. Voir aussi un des exemples [C#](#)⁸⁹, [C++](#)⁹⁶, [VB.NET](#)⁹², [VBScript](#)⁹⁹ ou [Java](#)¹⁰⁹ précédents.

Dans cet exemple, MapForce est utilisé de manière à ce que vous pouvez consulter et comprendre le design de mappage d'origine. MapForce est aussi utilisé pour compiler le mappage dans un fichier d'exécution MapForce Server (.mfx) et de configurer les paramètres tels que des chemins relatifs par rapport à des chemins absolus.

La plateforme de serveur utilisée dans l'exemple est Windows. Elle peut se trouver soit sur le même appareil sur lequel MapForce est installé, soit un autre appareil. Vous pouvez aussi exécuter cet exemple sur un appareil Linux ou Mac (dans Java), à condition que vous ajustiez les chemins de style Windows pour qu'ils soient applicables à votre plateforme.

Exigences préalables

L'exécution de ce mappage nécessite les mêmes exigences préalables que celles décrites dans les exemples [C#](#)⁸⁹, [C++](#)⁹⁶, [VB.NET](#)⁹², [VBScript](#)⁹⁹ ou [Java](#)¹⁰⁹ précédents.

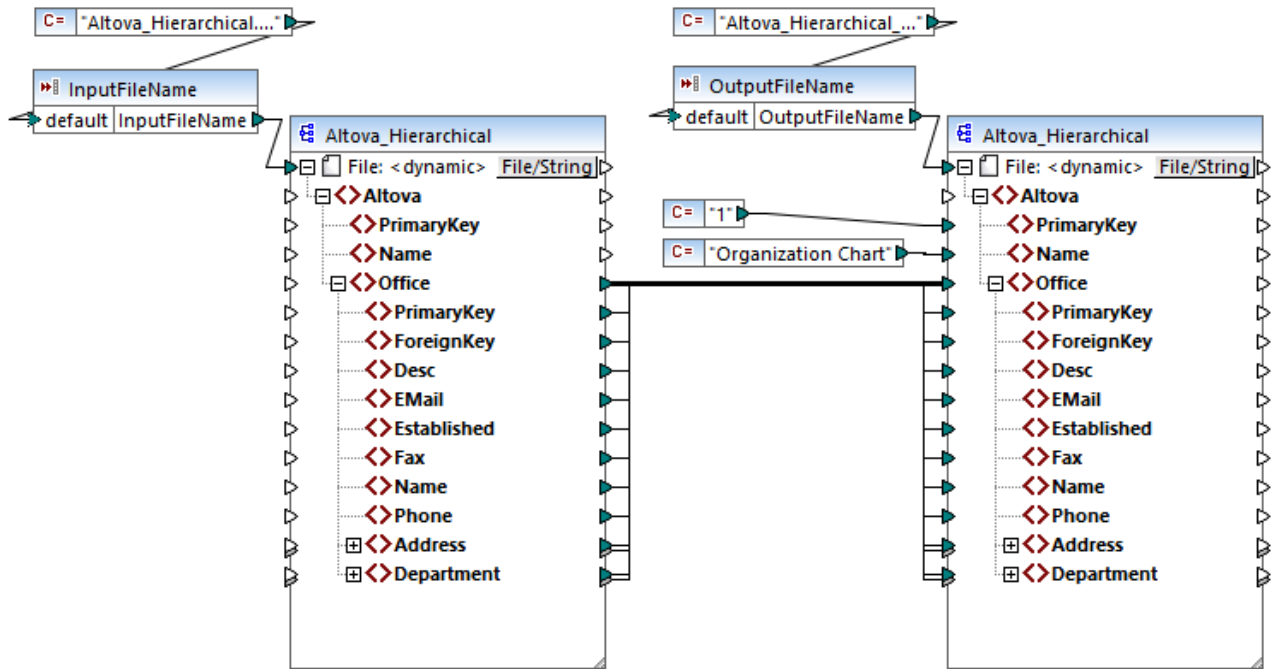
Préparer le mappage pour l'exécution de serveur

Une fois avoir installé MapForce et avoir exécuté MapForce pour la première fois, plusieurs fichiers de design de mappage de démonstration sont disponibles sous :

C:\Users\<username>\Documents\Altova\MapForce2024\MapForceExamples

Assurez-vous de modifier le chemin ci-dessus si vous avez une version différente de MapForce.

Le design de mappage utilisé dans cet exemple est appelé **FileNamesAsParameters.mfd**. Comme vous pouvez le constater ci-dessous, il s'agit d'un mappage classique qui copie des données provenant d'une source vers un fichier XML cible, presque sans modifications. Seuls les champs **PrimaryKey** et **Name** du fichier XML cible sont remplis avec des constantes provenant du mappage.



FileNamesAsParameters.mfd

Le mappage a deux paramètres d'entrée : **InputFileName** et **OutputFileName**, respectivement. Afin de pouvoir consulter le mappage dans MapForce, les valeurs de paramètre sont fournies par deux constantes. Comme vous le verrez par la suite, vous pourrez remplacer les valeurs de paramètre avec vos valeurs lorsque le mappage est exécuté avec MapForce Server.

Veillez noter que les composants de mappage source et cible sont configurés pour recevoir le nom de fichier dynamiquement depuis le mappage. Concrètement, un connecteur d'entrée *File: <dynamic>* est disponible tout en haut du composant et il lit des données depuis les paramètres de mappage respectifs (**InputFileName** ou **OutputFileName**). Dans MapForce, vous pouvez définir ou modifier cette option en cliquant sur la touche **Fichier/String** dans le coin situé en haut à droite du composant. Veillez noter que le nom de fichier d'entrée fourni en tant que paramètre doit être une référence à un fichier XML physique qui apporte des données d'entrée dans le mappage (dans cet exemple, **Altova_Hierarchical.xml** provenant du même dossier que le fichier de design de mappage). Le nom de fichier de sortie peut être n'importe quel nom valide, par exemple **Output.xml**.

Avant de compiler le mappage dans un fichier d'exécution de MapForce Server (.mfx), vous devrez revoir les paramètres de mappage. Cliquer avec la touche de droite sur une zone vide dans le mappage et sélectionner **Paramètres de mappage** dans le menu contextuel.

Paramètres de mappage

Génération de code

Nom d'application :

Nom pack de base Java :

Rendre les chemins absolus dans le code généré

Bibliothèques de référence avec des chemins relatifs aux fichiers XSLT / XQue

Assurer la convention de chemin Windows pour la sortie de chemin de fichier pour les fichiers d'un système de fichier local

Paramètres de fichier de sortie

Terminaisons de ligne :

(prises en charge dans l'exécution Built-in et la génération de code C#, Java et C++)

Modifiez les paramètres tels qu'indiqués ci-dessus. Lorsque vous décochez la case **Rendre les chemins absolus dans le code généré**, tous les chemins absolus vers les fichiers d'entrée et de sortie utilisés par le mappage sont résolus en tant que relatifs par rapport au fichier de design de mappage (.mfd). Lors de l'exécution du mappage, MapForce Server cherchera ces chemins dans le répertoire de travail du programme. Le répertoire de travail par défaut dépend de la plateforme que vous utilisez pour accéder au API de MapForce Server. Par exemple, dans le cas d'une application C#, il s'agit du même répertoire que le programme d'exécution. De plus, comme vous le verrez ci-dessous, vous pouvez modifier le répertoire de travail avec l'aide d'une propriété d'API appelée [WorkingDirectory](#)¹²⁷ (ou méthode Java [setWorkingDirectory](#)¹⁴⁶, le cas échéant).

Notes:

- Dans le dialogue Paramètres de mappage, les seuls paramètres qui affectent directement la compilation du fichier .mfx sont **Rendre les chemins absolus dans le code généré** et **Terminaisons de ligne**.
- Dans ce mappage, il n'était pas nécessaire d'éditer chaque composant de mappage dans MapForce afin de modifier des chemins absolus en chemins relatifs parce que tous les chemins étaient déjà relatifs. Sinon, vous devriez effectuer cette étape supplémentaire également, voir [Préparer des mappages pour l'exécution de serveur](#)⁴³.

Dans cet exemple, nous allons utiliser **C:\MappingExample** en tant que répertoire de travail. Copier le fichier d'entrée **Altova_Hierarchical.xml** référencé par le mappage depuis **C:\Users\\Documents\Altova\MapForce2024\MapForceExamples** vers le répertoire de travail.

Dans cet exemple, la source et la cible sont des fichiers XML, et il n'est pas nécessaire de copier le fichier de schéma XML vers le serveur car l'information qui en découle est intégrée dans le fichier .mfx pendant la compilation. Si votre mappage utilise d'autres types de composants (par exemple, des bases de données), il se peut que vous nécessitez d'autres exigences préalables, comme décrit dans [Préparer des mappages pour l'exécution de serveur](#)⁴³.

Enfin, pour compiler le mappage dans un fichier d'exécution de serveur, procéder comme suit :

- Dans le menu **Fichier**, cliquer sur **Compiler vers le fichier d'exécution MapForce Server** et choisir un répertoire de cible. Dans cet exemple, le répertoire de cible est le même que le répertoire de travail, **C:\MappingExample**.

Les listes de code suivantes illustrent comment exécuter le mappage depuis des environnements variés, en utilisant l'API MapForce Server. Dans les extraits de code ci-dessous, veuillez noter que les deux paramètres sont référencés par le même nom que celui contenu dans le design de mappage. De plus, les valeurs de paramètre sont le même type de données que celui défini dans le mappage (dans ce cas, `string`).

Exécuter le mappage depuis C++

Vous pouvez maintenant exécuter le fichier `.mfx` en utilisant le code C++ comme celui ci-dessous. Pour pouvoir exécuter cet extrait de code, assurez-vous que MapForce Server ait une licence valide, voir aussi la commande CLI [verifylicense](#)⁸⁵.

C++

```
#include <iostream>
#include "atlbase.h"

// 32-bit MapForce Server
#import "progid:MapForce.Server"

int _tmain(int argc, _TCHAR* argv[])
{
    CoInitialize( NULL );

    try
    {
        //Create a MapForce Server object
        MapForceServerLib::IServerPtr pMFS;
        CoCreateInstance( __uuidof( MapForceServerLib::Server ), NULL, CLSCTX_ALL,
            __uuidof( MapForceServerLib::IServer ), reinterpret_cast< void** >( &pMFS ) );

        //Set a working directory - used as a base for relative paths
        pMFS->WorkingDirectory = "C:\\\\MappingExample";

        //Prepare the parameters
        pMFS->AddParameter( "InputFileName", "Altova_Hierarchical.xml" );
        pMFS->AddParameter( "OutputFileName", "Output.xml" );

        //Run the mapping
        if (pMFS->Run( "FileNamesAsParameters.mfx" ))
            std::cout << "Success - finished execution" << std::endl;
        else
            std::cout << pMFS->LastExecutionMessage << std::endl;
    }
    catch ( _com_error& err )
    {
        BSTR bstrMessage;
        (err).ErrorInfo()->GetDescription( &bstrMessage );
        std::cout << "Exception occurred: " <<
            _com_util::ConvertBSTRToString( bstrMessage ) << std::endl;
    }
}
```

```

}

CoUninitialize();
return 0;

}

```

Exécuter le mappage depuis C#

Vous pouvez maintenant exécuter le fichier .mfx en utilisant le code C# comme celui ci-dessous. Pour pouvoir exécuter cet extrait de code, il faut tout d'abord ajouter une référence au MapForce Server DLL dans Visual Studio, comme décrit dans [.NET Interface](#)⁸⁸, et s'assurer que MapForce Server a une licence valide, voir aussi la commande CLI [verifylicense](#)⁸⁵.

C#

```

static void Main(string[] args)
{
    try
    {
        // Create a MapForce Server object
        Altova.MapForceServer.Server objMFS = new Altova.MapForceServer.Server();

        // Set the working directory - all relative paths will be resolved against it
        objMFS.WorkingDirectory = "C:\\MappingExample";

        //Prepare the parameters
        objMFS.AddParameter("InputFileName", "Altova_Hierarchical.xml");
        objMFS.AddParameter("OutputFileName", "Output.xml");

        //Run the mapping
        if (objMFS.Run("FileNamesAsParameters.mfx"))
            System.Console.WriteLine("Success - finished execution");
        else
            System.Console.WriteLine(objMFS.LastExecutionMessage);
    }
    catch(System.Runtime.InteropServices.COMException ex)
    {
        System.Console.WriteLine("Internal Error: " + ex.Message);
    }
}

```

Exécuter le mappage depuis Java

Vous pouvez maintenant exécuter le fichier .mfx en utilisant le code Java comme celui ci-dessous. Pour pouvoir exécuter cet extrait de code, il faut tout d'abord que :

- la variable d'environnement Java CLASSPATH inclue une référence à la bibliothèque **MapForceServer.jar**, comme décrit dans [A propos de l'interface Java](#)¹⁰⁵
- MapForce Server ait une licence valide, voir aussi la commande CLI [verifylicense](#)⁸⁵.

Java

```
public static void main(String[] args) {
    try {
        // Create a MapForce Server object
        com.altova.mapforceserver.MapForceServer objMFS = new
com.altova.mapforceserver.MapForceServer();

        // Set the working directory - all relative paths will be resolved against it
        // objMFS.setWorkingDirectory("/home/ubuntu/Downloads/MappingExample");
        objMFS.setWorkingDirectory("C:\\MappingExample");

        // Add the mapping parameters
        objMFS.addParameter("InputFileName", "Altova_Hierarchical.xml");
        objMFS.addParameter("OutputFileName", "Output.xml");

        // Run the mapping
        if ( objMFS.run( "FileNamesAsParameters.mfx" ) )
            System.out.println( "Success - finished execution" );
        else
            System.out.println( objMFS.getLastExecutionMessage() );

    } catch (Exception e) {
        e.printStackTrace();
    }

    System.out.println("Finished execution");
}
```

6.5 API Reference (COM, .NET)

Cette section fournit une référence générale aux éléments API MapForce Server (comme les interfaces et les méthodes) applicables au code écrit pour les plateformes COM ou .NET.

6.5.1 Interfaces

6.5.1.1 IServer

L'interface `IServer` crée une instance d'objet de MapForce Server, et fournit l'accès au moteur de MapForce Server.

Si vous utilisez C++ sous une plateforme COM, le nom de l'interface principale est `IServer`. Si vous utilisez un langage .NET comme C# ou VB.NET, le nom de l'interface principale est `Server`.

Propriétés

Nom	Description
APIMajorVersion ¹²¹	Read-only. Obtient la version majeure de l'API de MapForce Server. Cela peut différer de la version de produit si l'API est connectée à un autre serveur.
APIMinorVersion ¹²¹	Read-only. Obtient la version mineure de l'API de MapForce Server. Cela peut différer de la version de produit si l'API est connectée à un autre serveur.
APIServicePackVersion ¹²²	Read-only. Obtient la version de service de l'API MapForce Server. Cela peut différer de la version de produit si l'API est connectée à un autre serveur.
Is64Bit ¹²³	Read-only. Retourne true si le moteur MapForce Server est un exécutable 64-bit.
LastExecutionMessage ¹²³	Read-only. Obtient le message reçu pendant la dernière commande Run .
MajorVersion ¹²⁴	Read-only. Obtient la version majeure du produit, en tant que nombre des années débutant avec 1998 (par exemple, "20" pour Altova MapForce Server 2018 r2 sp1 (x64)).
MinorVersion ¹²⁴	Read-only. Obtient la version mineure du produit (par exemple, "2" pour Altova MapForce Server 2018 r2 sp1 (x64)).

Nom	Description
ProductName ¹²⁵	Read-only. Obtient le nom du produit (par exemple, "Altova MapForce Server").
ProductNameAndVersion ¹²⁵	Read-only. Obtient le nom complet et la version du produit (par exemple, "Altova MapForce Server 2018 r2 sp1 (x64)").
ServerPath ¹²⁶	Obtient ou définit le chemin de l'exécutable MapForce Server.
ServicePackVersion ¹²⁶	Read-only. Obtient la version de pack de service du produit (par exemple, "1" pour Altova MapForce Server 2018 r2 sp1).
WorkingDirectory ¹²⁷	Obtient ou définit le répertoire actuel pour exécuter les tâches (les chemins relatifs seront évalués par rapport au répertoire de travail).

Méthodes

Nom	Description
AddCredentialProperty ¹²⁷	Ajoute une propriété à l'identifiant actuel (par exemple, le nom d'utilisateur, le mot de passe, ou les deux). Le premier argument spécifie le nom de la propriété, et le second argument spécifie la valeur de la propriété. Noms de propriété valides : <code>username</code> , <code>password</code> . Dans MapForce Server Advanced Edition, le nom de propriété <code>oauth:token</code> est également pris en charge. Cette méthode doit être appelée après avoir appelé <code>BeginCredential()</code> et avant d'appeler <code>EndCredential()</code> .
Add Parameter ¹²⁸	Assigne une valeur à un paramètre défini dans le mappage. Le premier argument spécifie le nom du paramètre tel que défini dans le mappage ; le second argument spécifie la valeur de paramètre.
BeginCredential ¹²⁹	Crée un nouvel identifiant avec le nom fourni en tant qu'argument. Si vous appelez cette méthode, vous devez aussi y ajouter des propriétés en utilisant <code>AddCredentialProperty()</code> , puis fermer l'identifiant en appelant <code>EndCredential()</code> .
ClearCredentialList ¹³⁰	Supprime la liste des identifiants définis précédemment. Tous les identifiants sont valides pour la durée de vie de l'objet. Appeler cette méthode si vous souhaitez supprimer explicitement tous les identifiants définis précédemment.
Clear Options ¹³⁰	Efface la liste des options définies précédemment par le biais de la méthode <code>SetOption</code> . Toutes les options définies par la méthode <code>SetOption</code> sont valides pour la durée de vie de l'objet.

Nom	Description
	Appeler cette méthode si vous souhaitez supprimer explicitement toutes les options définies précédemment.
ClearParameterList ¹³¹	Supprime la liste des paramètres qui étaient définis précédemment en utilisant la méthode <code>AddParameter</code> .
EndCredential ¹³¹	Ferme un objet d'identifiant qui a été créé précédemment en utilisant la méthode <code>BeginCredential</code> .
GetOutputParameter ¹³²	Obtient la sortie de string générée par la dernière commande <code>run</code> . Retourne null si aucune sortie n'est générée. Cette fonction requiert un paramètre de string qui identifie le nom du composant de sortie dans MapForce.
Run ¹³³	Exécute un fichier MapForce Server Execution (fichier <code>.mfx</code>). Retourne true en cas de succès ; sinon, false .
SetOption ¹³³	<p>Définit une option avant d'exécuter le mappage. Le premier argument spécifie le nom de l'option ; le second argument spécifie la valeur d'option. Cette méthode est particulièrement utile lorsqu'un mappage a été conçu pour consommer des Ressources globales (voir Altova Global Resources ⁴⁹). Les options prises en charge actuellement sont les suivantes :</p> <ul style="list-style-type: none"> • globalresourcefile (ou gr) - Un chemin de fichier Ressource Globale . (Lorsque cette option est spécifiée, un nom de configuration de Ressource Globale doit aussi être spécifié, voir item suivant). • globalresourceconfig (ou gc) - Un nom de configuration de Ressource Globale. (Lorsque cette option est spécifiée, un chemin de fichier de Ressource Globale doit aussi être spécifié, voir item précédent). • catalog - Le chemin vers un fichier RootCatalog.xml personnalisé. Cette option vous permet de spécifier un fichier de catalogue personnalisé utilisé pour résoudre des URL utilisés par le mappage. Le catalogue par défaut se trouve dans le sous-dossier etc du répertoire d'installation du programme. • taxonomy-package - Le chemin vers un paquet de taxonomie XBRL personnalisé, si un tel pack est requis par le mappage. • taxonomy-packages-config-file - Le chemin vers une configuration de taxonomie XBRL personnalisée, si un tel pack est requis par le mappage. <p>Toutes les options définies sont valides pour la durée de vie de l'objet. Si vous définissez une option avec le même nom deux fois, l'option précédente sera contournée. Pour supprimer explicitement toutes les options, appeler la méthode <code>ClearOptions()</code>.</p>

Nom	Description
StopServerProcess ¹³⁴	Cette méthode stoppe <i>explicitement</i> le processus connecté avec l'objet COM, sans relâcher l'objet. Le processus s'arrête implicitement lorsque l'objet COM est relâché.

Exemples

Voir les exemples suivants qui illustrent comment initialiser et exécuter MapForce Server dans des langages différents :

- [Exemple C++](#) ⁹⁶
- [Exemple C#](#) ⁸⁹
- [Exemple VBScript](#) ⁹⁹
- [Exemple VB.NET](#) ⁹²

6.5.1.1.1 Propriétés

6.5.1.1.1.1 APIMajorVersion

Obtient la version majeure de l'API de MapForce Server. Cela peut différer de la version de produit si l'API est connectée à un autre serveur.

Signature

```
APIMajorVersion : Integer
```

Signature générique

```
int APIMajorVersion { get; }
```

C#

```
HRESULT APIMajorVersion([out, retval] INT* pRetVal);
```

C++

```
ReadOnly Property APIMajorVersion As Integer
```

VB.NET

6.5.1.1.1.2 APIMinorVersion

Obtient la version mineure de l'API de MapForce Server. Cela peut différer de la version de produit si l'API est connectée à un autre serveur.

Signature

```
APIMinorVersion : Integer
```

Signature générique

```
int APIMinorVersion { get; }
```

C#

```
HRESULT APIMinorVersion([out, retval] INT* pRetVal);
```

C++

```
ReadOnly Property APIMinorVersion As Integer
```

VB.NET

6.5.1.1.1.3 APIServicePackVersion

Obtient la version de service de l'API MapForce Server. Cela peut différer de la version de produit si l'API est connectée à un autre serveur.

Signature

```
APIServicePackVersion : Integer
```

Signature générique

```
int APIServicePackVersion { get; }
```

C#

```
HRESULT APIServicePackVersion([out, retval] INT* pRetVal);
```

C++

```
ReadOnly Property APIServicePackVersion As Integer
```

VB.NET

6.5.1.1.1.4 Is64Bit

Retourne **true** si le moteur MapForce Server est un exécutable 64-bit.

Signature

```
Is64Bit : Boolean
```

Signature générique

```
bool Is64Bit { get; }
```

C#

```
HRESULT Is64Bit([out, retval] VARIANT_BOOL* pbVal);
```

C++

```
ReadOnly Property Is64Bit As Boolean
```

VB.NET

6.5.1.1.1.5 LastExecutionMessage

Obtient le message reçu pendant la dernière commande **Run**.

Signature

```
LastExecutionMessage : String
```

Signature générique

```
string LastExecutionMessage { get; }
```

C#

```
HRESULT LastExecutionMessage([out, retval] BSTR* pbstrResult );
```

C++

```
ReadOnly Property LastExecutionMessage As String
```

VB.NET

6.5.1.1.1.6 MajorVersion

Obtient la version majeure du produit, en tant que nombre des années débutant avec 1998 (par exemple, "20" pour Altova MapForce Server 2018).

Signature

```
MajorVersion : Integer
```

Signature générique

```
int MajorVersion { get; }
```

C#

```
HRESULT MajorVersion([out, retval] INT* pnVal);
```

C++

```
ReadOnly Property MajorVersion As Integer
```

VB.NET

6.5.1.1.1.7 MinorVersion

Obtient la version mineure du produit (par exemple, "2" pour Altova MapForce Server 2018 r2).

Signature

```
MinorVersion : Integer
```

Signature générique

```
int MinorVersion { get; }
```

C#

```
HRESULT MinorVersion([out, retval] INT* pnVal);
```

C++

```
ReadOnly Property MinorVersion As Integer
```

VB.NET

6.5.1.1.1.8 *ProductName*

Obtient le nom du produit (par exemple, "Altova MapForce Server").

Signature

```
ProductName : String
```

Signature générique

```
string ProductName { get; }
```

C#

```
HRESULT ProductName([out, retval] BSTR* pstrVal);
```

C++

```
ReadOnly Property ProductName As String
```

VB.NET

6.5.1.1.1.9 *ProductNameAndVersion*

Obtient le nom complet et la version du produit (par exemple, "Altova MapForce Server 2018 r2 sp1 (x64)").

Signature

```
ProductNameAndVersion : String
```

Signature générique

```
string ProductNameAndVersion { get; }
```

C#

```
HRESULT ProductNameAndVersion([out, retval] BSTR* pstrVal);
```

C++

```
ReadOnly Property ProductNameAndVersion As String
```

VB.NET

6.5.1.1.10 *ServerPath*

Obtient ou définit le chemin de l'exécutable MapForce Server.

Signature

```
ServerPath : String
```

Signature générique

```
string ServerPath { set; get; }
```

C#

```
HRESULT ServerPath([in] BSTR bstrServerFile );  
HRESULT ServerPath([out, retval] BSTR* pbstrServerFile );
```

C++

```
Property ServerPath As String
```

VB.NET

6.5.1.1.11 *ServicePackVersion*

Obtient la version de pack de service du produit (par exemple, "1" pour Altova MapForce Server 2018 r2 sp1).

Signature

```
ServicePackVersion : Integer
```

Signature générique

```
int ServicePackVersion { get; }
```

C#

```
HRESULT ServicePackVersion([out, retval] INT* pnVal);
```

C++

```
ReadOnly Property ServicePackVersion As Integer
```

VB.NET

6.5.1.1.12 WorkingDirectory

Obtient ou définit le répertoire actuel pour exécuter les tâches (les chemins relatifs seront évalués par rapport au répertoire de travail).

Signature

```
WorkingDirectory : String
```

Signature générique

```
string WorkingDirectory { set; get; }
```

C#

```
HRESULT WorkingDirectory([in] BSTR bstrWorkingDirectory );  
HRESULT WorkingDirectory([out, retval] BSTR* pbstrWorkingDirectory );
```

C++

```
Property WorkingDirectory As String
```

VB.NET

6.5.1.1.2 Méthodes

6.5.1.1.2.1 AddCredentialProperty

Ajoute une propriété à l'identifiant actuel (par exemple, le nom d'utilisateur, le mot de passe, ou les deux). Le premier argument spécifie le nom de la propriété, et le second argument spécifie la valeur de la propriété. Noms de propriété valides : `username`, `password`. Dans MapForce Server Advanced Edition, le nom de propriété `oauth:token` est également pris en charge.

Cette méthode doit être appelée après avoir appelé `BeginCredential()` et avant d'appeler `EndCredential()`.

Signature

```
AddCredentialProperty(in bstrName: System.String, in bstrValue: System.String) -> Void
```

Signature générique

```
void AddCredentialProperty(string bstrName, string bstrValue)
```

C#

```
HRESULT AddCredentialProperty([in] BSTR bstrName, [in] BSTR bstrValue );
```

C++

```
Sub AddCredentialProperty(ByVal bstrName As String, ByVal bstrValue As String)
```

VB.NET

Paramètres

Name	Type	Description
bstrName	<code>System.String</code>	Spécifie le nom de la propriété de l'identifiant.
bstrValue	<code>System.String</code>	Spécifie la valeur de la propriété de l'identifiant

Exemples

L'extrait de code suivant illustre comment déclarer un identifiant appelé "mycredential" dans C#. Le nom de l'identifiant doit être celui donné à l'identifiant dans MapForce au moment de la création du design.

```
//Create a MapForce Server object
Altova.MapForceServer.Server objMFS = new Altova.MapForceServer.Server();

objMFS.BeginCredential("mycredential");
objMFS.AddCredentialProperty("username", "altova");
objMFS.AddCredentialProperty("password", "b45ax78!");
objMFS.EndCredential();
```

6.5.1.1.2.2 AddParameter

Assigne une valeur à un paramètre défini dans le mappage. Le premier argument spécifie le nom du paramètre tel que défini dans le mappage ; le second argument spécifie la valeur de paramètre.

Signature

```
AddParameter(in bstrName:String, in bstrValue:String) -> Void
```

Signature générique

```
void AddParameter(string bstrName, string bstrValue)
```

C#

```
HRESULT AddParameter([in] BSTR bstrName, [in] BSTR bstrValue );
```


C++

```
Sub AddParameter(ByVal bstrName As String, ByVal bstrValue As String)
```

VB.NET

Paramètres

Nom	Type	Description
bstrName	String	Spécifie le nom du paramètre.
bstrValue	String	Spécifie la valeur de paramètre.

6.5.1.1.2.3 *BeginCredential*

Crée un nouvel identifiant avec le nom fourni en tant qu'argument. Si vous appelez cette méthode, vous devez aussi y ajouter des propriétés en utilisant `AddCredentialProperty()`, puis fermer l'identifiant en appelant `EndCredential()`.

Signature

```
BeginCredential(in bstrCredentialName:String) -> Void
```

Signature générique

```
void BeginCredential(string bstrCredentialName)
```

C#

```
HRESULT BeginCredential([in] BSTR bstrCredentialName);
```

C++

```
Sub BeginCredential(ByVal bstrCredentialName As String)
```

VB.NET

Paramètres

Nom	Type	Description
bstrCredentialName	String	Spécifie le nom de l'identifiant tel qu'il a été défini dans MapForce.

6.5.1.1.2.4 *ClearCredentialList*

Supprime la liste des identifiants définis précédemment. Tous les identifiants sont valides pour la durée de vie de l'objet. Appeler cette méthode si vous souhaitez supprimer explicitement tous les identifiants définis précédemment.

Signature

```
ClearCredentialList() -> Void
```

Signature générique

```
void ClearCredentialList()
```

C#

```
HRESULT ClearCredentialList();
```

C++

```
Sub ClearCredentialList()
```

VB.NET

6.5.1.1.2.5 *ClearOptions*

Supprime la liste des options définies précédemment par le biais de la méthode `SetOption`. Toutes les options définies par la méthode `SetOption` sont valides pour la durée de vie de l'objet. Appeler cette méthode si vous souhaitez supprimer explicitement toutes les options définies précédemment.

Signature

```
ClearOptions() -> Void
```

Signature générique

```
void ClearOptions()
```

C#

```
HRESULT ClearOptions();
```

C++

```
Sub ClearOptions()
```

VB.NET

6.5.1.1.2.6 *ClearParameterList*

Supprime la liste des paramètres qui étaient définis précédemment en utilisant la méthode `AddParameter`.

Signature

```
ClearParameterList() -> Void
```

Signature générique

```
void ClearParameterList()
```

C#

```
HRESULT ClearParameterList();
```

C++

```
Sub ClearParameterList()
```

VB.NET

6.5.1.1.2.7 *EndCredential*

Ferme un objet d'identifiant qui a été créé précédemment en utilisant la méthode `BeginCredential`.

Signature

```
EndCredential() -> Void
```

Signature générique

```
void EndCredential()
```

C#

```
HRESULT EndCredential();
```

C++

```
Sub EndCredential()
```

VB.NET

6.5.1.1.2.8 *GetOutputParameter*

Obtient la sortie de string générée par la dernière commande `run`. Retourne null si aucune sortie n'est générée. Cette fonction requiert un paramètre de string qui identifie le nom du composant de sortie dans MapForce.

Signature

```
GetOutputParameter(in bstrName:String) -> String
```

Signature générique

```
string GetOutputParameter(string bstrName)
```

C#

```
HRESULT GetOutputParameter([in] BSTR bstrName, [out, retval] BSTR* pbstrValue );
```

C++

```
Function GetOutputParameter(bstrName As String) As String
```

VB.NET

Paramètres

Nom	Type	Description
bstrName	<code>String</code>	Spécifie le nom du composant de la sortie tel qu'il apparaît dans MapForce. Le nom est affiché dans la barre de titre de chaque composant dans le mappage (ou lorsque vous cliquez avec la touche de droite sur l'en-tête de composant, et que vous choisissez Propriétés).

6.5.1.1.2.9 Run

Exécute un fichier MapForce Server Execution (fichier .mfx). Retourne **true** en cas de succès ; sinon, **false**.

Signature

```
Run(in bstrMappingPath:String) -> Boolean
```

Signature générique

```
bool Run(string bstrMappingPath)
```

C#

```
HRESULT Run( [in] BSTR bstrMappingPath, [out, retval] VARIANT_BOOL* pbSuccess );
```

C++

```
Function Run(ByVal bstrMappingPath As String) As Boolean
```

VB.NET

Paramètres

Nom	Type	Description
bstrMappingPath	<i>String</i>	Spécifie le chemin vers le fichier d'exécution MapForce Server (.mfx). Si vous spécifiez un chemin relatif, il sera résolu par rapport au répertoire de travail. Vous pouvez définir le répertoire de travail depuis la propriété <code>WorkingDirectory</code> .

6.5.1.1.2.10 SetOption

Définit une option avant d'exécuter le mappage. Le premier argument spécifie le nom de l'option ; le second argument spécifie la valeur d'option. Cette méthode est particulièrement utile lorsqu'un mappage a été conçu pour consommer des Ressources globales (voir [Altova Global Resources](#)⁴⁹). Les options prises en charge actuellement sont les suivantes :

- **globalresourcefile (ou gr)** - Un chemin de fichier Ressource Globale. (Lorsque cette option est spécifiée, un nom de configuration de Ressource Globale doit aussi être spécifié, voir item suivant).
- **globalresourceconfig (ou gc)** - Un nom de configuration de Ressource Globale. (Lorsque cette option est spécifiée, un chemin de fichier de Ressource Globale doit aussi être spécifié, voir item précédent).

- **catalog** - Le chemin vers un fichier **RootCatalog.xml** personnalisé. Cette option vous permet de spécifier un fichier de catalogue personnalisé utilisé pour résoudre des URL utilisés par le mappage. Le catalogue par défaut se trouve dans le sous-dossier, etc. du répertoire d'installation du programme.
- **taxonomy-package** - Le chemin vers un paquet de taxonomie XBRL personnalisé, si un tel pack est requis par le mappage.
- **taxonomy-packages-config-file** - Le chemin vers une configuration de taxonomie XBRL personnalisée, si un tel pack est requis par le mappage.

Toutes les options définies sont valides pour la durée de vie de l'objet. Si vous définissez une option avec le même nom deux fois, l'option précédente sera contournée. Pour supprimer explicitement toutes les options, appeler la méthode `ClearOptions()`.

Signature

```
SetOption(in bstrName:String, in bstrValue:String) -> Void
```

Signature générique

```
void SetOption(ByVal bstrName As String, ByVal bstrValue As String)
```

C#

```
HRESULT SetOption([in] BSTR bstrName, [in] BSTR bstrValue );
```

C++

```
Sub SetOption(ByVal bstrName As String, ByVal bstrValue As String)
```

VB.NET

Paramètres

Nom	Type	Description
bstrName	<code>String</code>	Spécifie le nom de l'option à définir.
bstrValue	<code>String</code>	Spécifie la valeur de l'option à définir.

6.5.1.1.2.11 StopServerProcess

Cette méthode stoppe *explicitement* le processus connecté avec l'objet COM, sans relâcher l'objet. Le processus s'interrompt implicitement lorsque l'objet COM est relâché.

Signature

```
StopServerProcess() -> System.Void
```

Signature générique

```
void StopServerProcess()
```

C#

```
Sub StopServerProcess()
```

VB.NET

6.6 API Reference (Java)

Cette section fournit une référence générale aux éléments API MapForce Server (comme les classes et les méthodes) applicables au code écrit pour les plateformes Java.

6.6.1 Classes

6.6.1.1 MapForceServer

La classe `MapForceServer` crée une nouvelle instance d'objet MapForce Server, et fournit un accès au moteur MapForce Server.

Méthodes

Nom	Description
addCredentialPropertiesFromMap ¹³⁹	Ajoute des propriétés d'un mappage de propriété de l'identifiant à un identifiant actuel Cette méthode prend un mappage de propriété de l'identifiant comme argument (<i>property_name</i> , <i>property_value</i>). Cette méthode doit être appelée après avoir appelé <code>beginCredential()</code> et avant d'appeler <code>endCredential()</code> . En alternative, vous pouvez aussi appeler <code>AddCredentialProperty()</code> .
addCredentialProperty ¹³⁹	Ajoute une propriété à l'identifiant actuel (par exemple, le nom utilisateur, le mot de passe ou les deux). Le premier argument spécifie le nom de propriété ; le second argument spécifie la valeur de propriété. Noms de propriété valides : <code>username</code> , <code>password</code> . Dans MapForce Server Advanced Edition, le nom de propriété <code>oauth:token</code> est de plus pris en charge. Cette méthode doit être appelée après avoir appelé <code>beginCredential()</code> et avant d'appeler <code>endCredential()</code> . En alternative, vous pouvez aussi appeler <code>addCredentialPropertiesFromMap()</code> .
addParameter ¹⁴⁰	Assigne une valeur à un paramètre défini dans le mappage.
beginCredential ¹⁴¹	Crée un nouvel identifiant avec le nom fourni comme argument. Si vous appelez cette méthode, vous devez également ajouter des propriétés en utilisant <code>addCredentialProperty()</code> ou <code>addCredentialPropertiesFromMap()</code> , puis fermer l'identifiant en appelant <code>endCredential()</code> .
clearCredentialList ¹⁴¹	Obtient la liste des identifiants définis précédemment. Tous les identifiants sont valides pour la durée de vie de l'objet. Appeler cette méthode si vous souhaitez supprimer explicitement tous les identifiants définis précédemment.

Nom	Description
clearOptions ¹⁴¹	Efface la liste des options définies précédemment par le biais de la () méthode <code>SetOption</code> . Toutes les options définies par la méthode <code>SetOption</code> sont valides pour la durée de vie de l'objet. Appeler cette méthode si vous souhaitez supprimer explicitement toutes les options définies précédemment.
clearParameterList ¹⁴²	Supprime la liste des paramètres qui étaient définis précédemment en utilisant la méthode <code>addParameter</code> .
endCredential ¹⁴²	Ferme un objet d'identification qui a été précédemment créé utilisant la méthode <code>beginCredential</code> .
getAPIMajorVersion ¹⁴²	Obtient la version majeure de l'API de MapForce Server. Cela peut différer de la version de produit si l'API est connectée à un autre serveur.
getAPIMinorVersion ¹⁴²	Obtient la version mineure de l'API de MapForce Server. Cela peut différer de la version de produit si l'API est connectée à un autre serveur.
getAPIServicePackVersion ¹⁴²	Obtient la version de pack de service de l'API MapForce Server. Cela peut différer de la version de produit si l'API est connectée à un autre serveur.
getLastExecutionMessage ¹⁴³	Obtient le message reçu pendant la dernière commande <code>run</code> .
getMajorVersion ¹⁴³	Obtient la version majeure du produit, en tant que nombre des années débutant avec 1998 (par exemple, "20" pour Altova MapForce Server 2018 r2 sp1 (x64)).
getMinorVersion ¹⁴³	Obtient la version mineure du produit (par exemple, "2" pour Altova MapForce Server 2018 r2 sp1 (x64)).
getOutputParameter ¹⁴³	Obtient la sortie de string générée par la dernière commande <code>run</code> . Retourne null si aucune sortie n'est générée. Cette fonction requiert un paramètre de string qui identifie le nom du composant de sortie dans MapForce.
getProductName ¹⁴⁴	Obtient le nom du produit (par exemple, "Altova MapForce Server").
getProductNameAndVersion ¹⁴⁴	Obtient le nom complet et la version du produit (par exemple, "Altova MapForce Server 2018 r2 sp1 (x64)").
<code>getServerPath</code>	Obtient le chemin vers le fichier exécutable binaire du serveur.
getServicePackVersion ¹⁴⁴	Obtient la version de pack de service du produit (par exemple, "1" pour Altova MapForce Server 2018 r2 sp1 (x64)).
<code>getWorkingDirectory</code>	Obtient le Répertoire de travail actuel.
is64bit ¹⁴⁵	Retourne true si le moteur MapForce Server est un exécutable 64-bit.

Nom	Description
exécuter ¹⁴⁵	Exécute un fichier MapForce Server Execution (fichier .mfx). Retourne true en cas de succès ; sinon, false .
setOption ¹⁴⁵	<p>Définit une option avant d'exécuter le mappage. Le premier argument spécifie le nom de l'option ; le second argument spécifie la valeur d'option. Cette méthode est particulièrement utile lorsqu'un mappage a été conçu pour consommer des Ressources globales (voir Altova Global Resources ⁴⁹). Les options prises en charge actuellement sont les suivantes :</p> <ul style="list-style-type: none"> • globalresourcefile (ou gr) un chemin de fichier Ressource Globale. (Lorsque cette option est spécifiée, un nom de configuration de Ressource Globale doit aussi être spécifié, voir item suivant). • globalresourceconfig (ou gc) - un nom de configuration de Ressource Globale. (Lorsque cette option est spécifiée, un chemin de fichier de Ressource Globale doit aussi être spécifié, voir item précédent). • catalog - le chemin vers un fichier RootCatalog.xml personnalisé. Cette option vous permet de spécifier un fichier de catalogue personnalisé utilisé pour résoudre des URL utilisés par le mappage. Le catalogue par défaut se trouve dans le sous-dossier etc du répertoire d'installation du programme. • taxonomy-package - le chemin vers un paquet de taxonomie XBRL personnalisé, si un tel pack est requis par le mappage. • taxonomy-packages-config-file - le chemin vers une configuration de taxonomie XBRL personnalisée, si un tel pack est requis par le mappage. <p>Toutes les options définies sont valides pour la durée de vie de l'objet. Si vous définissez une option avec le même nom deux fois, l'option précédente sera contournée. Pour supprimer explicitement toutes les options, appeler la méthode <code>ClearOptions()</code>.</p>
setServerPath ¹⁴⁶	Définit le chemin de l'exécutable MapForce Server.
setWorkingDirectory ¹⁴⁶	Définit le répertoire actuel pour les tâches en cours (les chemins relatifs seront évalués par rapport au répertoire de travail).
arrêter	Arrête le processus du serveur.

Exemples

Pour voir un exemple de création d'une nouvelle instance de MapForceServer dans Java, voir l'[exemple Java](#) ¹⁰⁹.

6.6.1.1.1 Méthodes

6.6.1.1.1.1 *addCredentialPropertiesFromMap*

Ajoute des propriétés depuis un mappage de propriété d'identifiant dans l'identifiant actuel. Cette méthode prend en tant qu'argument un mappage de propriété d'identifiant (*property_name*, *property_value*). Cette méthode doit être appelée avant d'appeler `beginCredential()` et avant d'appeler `endCredential()`. Au lieu d'appeler cette méthode, vous pouvez aussi appeler `AddCredentialProperty()`.

Signature

```
addCredentialPropertiesFromMap(arg0:Map) -> void
```

Signature générique

Paramètres

Nom	Type	Description
arg0	Map	Un mappage qui fournit le nom de la propriété d'identifiant et sa valeur.

Exemples

L'extrait de code suivant illustre l'ajout d'un identifiant appelé "mycredential" au contexte actuel en utilisant la méthode `addCredentialPropertiesFromMap`.

```
//Create a MapForce Server object
com.altova.mapforceserver.MapForceServer objMFS = new com.altova.mapforceserver.MapForceS
erver();

objMFS.beginCredential("mycredential");
java.util.Map<String, String> credentialMap = new java.util.HashMap<String,String>();
credentialMap.put("username", "altova");
credentialMap.put("password", "b45ax78!");
objMFS.addCredentialPropertiesFromMap(credentialMap);
objMFS.endCredential();
```

6.6.1.1.1.2 *addCredentialProperty*

Ajoute une propriété à l'identifiant actuel (par exemple, le nom d'utilisateur, le mot de passe, ou les deux). Le premier argument spécifie le nom de la propriété, et le second argument spécifie la valeur de la propriété. Noms de propriété valides : `username`, `password`. Dans MapForce Server Advanced Edition, le nom de propriété `oauth:token` est également pris en charge.

Cette méthode doit être appelée après avoir appelé `beginCredential()` et avant d'appeler `endCredential()`. En alternative de l'appel de cette méthode, vous pouvez aussi appeler `addCredentialPropertiesFromMap()`.

Signature

```
addCredentialProperty(arg0:String, arg1:String) -> void
```

Signature générique

Paramètres

Nom	Type	Description
arg0	String	Le nom de la propriété de l'identifiant (par exemple, "username" ou "password").
arg1	String	La valeur de la propriété de l'identifiant.

Exemples

L'extrait de code suivant illustre comment déclarer un identifiant appelé "mycredential". Le nom de l'identifiant doit être celui donné à l'identifiant dans MapForce lors de la création du design.

```
//Create a MapForce Server object
com.altova.mapforceserver.MapForceServer objMFS = new com.altova.mapforceserver.MapForceS
erver();

objMFS.beginCredential("mycredential");
objMFS.addCredentialProperty("username", "altova");
objMFS.addCredentialProperty("password", "b45ax78!");
objMFS.endCredential();
```

6.6.1.1.1.3 addParameter

Assigne une valeur à un paramètre défini dans le mappage.

Signature

```
addParameter(arg0:String, arg1:String) -> void
```

Signature générique

Paramètres

Nom	Type	Description
arg0	String	Spécifie le nom du paramètre.
arg1	String	Spécifie la valeur de paramètre.

6.6.1.1.1.4 *beginCredential*

Crée un nouvel identifiant avec le nom fourni en tant qu'argument. Si vous appelez cette méthode, vous devez aussi y ajouter des propriétés en utilisant `addCredentialProperty()` ou `addCredentialPropertiesFromMap()`, puis fermer l'identifiant en appelant `endCredential()`.

Signature

```
beginCredential(arg0:String) -> void
```

Signature générique

Paramètres

Name	Type	Description
arg0	<code>String</code>	The name of the credential as it was defined in MapForce.

6.6.1.1.1.5 *clearCredentialList*

Supprime la liste des identifiants établis précédemment. Tous les identifiants sont valides pour la durée de vie de l'objet. Appeler cette méthode si vous souhaitez supprimer explicitement tous les identifiants définis précédemment.

Signature

```
clearCredentialList() -> void
```

Signature générique

6.6.1.1.1.6 *clearOptions*

Efface la liste des options définies précédemment par le biais de la méthode `SetOption`. Toutes les options définies par la méthode `SetOption` sont valides pour la durée de vie de l'objet. Appeler cette méthode si vous souhaitez supprimer explicitement toutes les options définies précédemment.

Signature

```
clearOptions() -> void
```

Signature générique

6.6.1.1.1.7 *clearParameterList*

Supprime la liste des paramètres qui étaient définis précédemment en utilisant la méthode `AddParameter`.

Signature

```
clearParameterList() -> void
```

Signature générique

6.6.1.1.1.8 *endCredential*

Ferme un objet d'identifiant qui a été créé précédemment avec la méthode `beginCredential`.

Signature

```
endCredential() -> void
```

Signature générique

6.6.1.1.1.9 *getAPIMajorVersion*

Obtient la version majeure de l'API de MapForce Server. Cela peut différer de la version de produit si l'API est connectée à un autre serveur.

Signature

```
getAPIMajorVersion() -> int
```

Signature générique

6.6.1.1.1.10 *getAPIMinorVersion*

Obtient la version mineure de l'API de MapForce Server. Cela peut différer de la version de produit si l'API est connectée à un autre serveur.

Signature

```
getAPIMinorVersion() -> int
```

Signature générique

6.6.1.1.1.11 *getAPIServicePackVersion*

Obtient la version de pack de service de l'API MapForce Server. Cela peut différer de la version de produit si l'API est connectée à un autre serveur.

Signature

```
getAPIServicePackVersion() -> int
```

Signature générique

6.6.1.1.12 *getLastExecutionMessage*

Obtient le message reçu pendant la dernière commande `Run`.

Signature

```
getLastExecutionMessage() -> java.lang.String
```

Signature générique

6.6.1.1.13 *getMajorVersion*

Obtient la version majeure du produit, en tant que nombre des années débutant avec 1998 (par exemple, "20" pour Altova MapForce Server 2018 r2 sp1 (x64)).

Signature

```
getMajorVersion() -> int
```

Signature générique

6.6.1.1.14 *getMinorVersion*

Obtient la version mineure du produit (par exemple, "2" pour Altova MapForce Server 2018 r2

Signature

```
getMinorVersion() -> int
```

Signature générique

6.6.1.1.15 *getOutputParameter*

Obtient la sortie de string générée par la dernière commande `run`. Retourne null si aucune sortie n'est générée. Cette fonction requiert un paramètre de string qui identifie le nom du composant de sortie dans MapForce.

Signature

```
getOutputParameter(arg0:String) -> java.lang.String
```

Signature générique

Paramètres

Nom	Type	Description
arg0	<code>String</code>	Spécifie le nom du composant de la sortie tel qu'il apparaît dans MapForce. Le nom est affiché dans la barre de titre de chaque composant dans le mappage (ou lorsque vous cliquez avec la touche de droite sur l'en-tête de composant, et que vous choisissez Propriétés).

6.6.1.1.1.16 `getProductName`

Obtient le nom du produit (par exemple, "Altova MapForce Server").

Signature

```
getProductName() -> java.lang.String
```

Signature générique

6.6.1.1.1.17 `getProductNameAndVersion`

Obtient le nom complet et la version du produit (par exemple, "Altova MapForce Server 2018 r2 sp1 (x64)").

Signature

```
getProductNameAndVersion() -> java.lang.String
```

Signature générique

6.6.1.1.1.18 `getServicePackVersion`

Obtient la version de pack de service du produit (par exemple, "1" pour Altova MapForce Server 2018 r2 sp1 (x64)).

Signature

```
getServicePackVersion() -> int
```

Signature générique

6.6.1.1.19 *is64bit*

Retourne **true** si le moteur MapForce Server est un exécutable 64-bit.

Signature

```
is64bit() -> boolean
```

Signature générique

6.6.1.1.20 *run*

Exécute un fichier MapForce Server Execution (fichier .mfx). Retourne **true** en cas de succès ; sinon, **false**.

Signature

```
run(arg0:String) : boolean
```

Signature générique

Paramètres

Nom	Type	Description
arg0	String	Spécifie le chemin vers le fichier d'exécution de MapForce Server (fichier .mfx).

6.6.1.1.21 *setOption*

Définit une option avant d'exécuter le mappage. Le premier argument spécifie le nom de l'option ; le second argument spécifie la valeur d'option. Cette méthode est particulièrement utile lorsqu'un mappage a été conçu pour consommer des Ressources globales (voir [Altova Global Resources](#)⁴⁹). Les options prises en charge actuellement sont les suivantes :

- **globalresourcefile (ou gr)** - Un chemin de fichier Ressource Globale . (Lorsque cette option est spécifiée, un nom de configuration de Ressource Globale doit aussi être spécifié, voir item suivant).
- **globalresourceconfig (ou gc)** - Un nom de configuration de Ressource Globale. (Lorsque cette option est spécifiée, un chemin de fichier de Ressource Globale doit aussi être spécifié, voir item précédent).
- **catalog** - Le chemin vers un fichier **RootCatalog.xml** personnalisé. Cette option vous permet de spécifier un fichier de catalogue personnalisé utilisé pour résoudre des URL utilisés par le mappage. Le catalogue par défaut se trouve dans le sous-dossier etc du répertoire d'installation du programme.
- **taxonomy-package** - Le chemin vers un paquet de taxonomie XBRL personnalisé, si un tel pack est requis par le mappage.
- **taxonomy-packages-config-file** - Le chemin vers une configuration de taxonomie XBRL personnalisée, si un tel pack est requis par le mappage.

Toutes les options définies sont valides pour la durée de vie de l'objet. Si vous définissez une option avec le même nom deux fois, l'option précédente sera contournée. Pour supprimer explicitement toutes les options, appeler la méthode `clearOptions()`.

Signature

```
setOption(arg0:String, arg1:String) -> void
```

Signature générique

Paramètres

Nom	Type	Description
arg0	String	Spécifie le nom de l'option à définir.
arg1	String	Spécifie la valeur de l'option à définir.

6.6.1.1.1.22 `setServerPath`

Définit le chemin de l'exécutable MapForce Server.

Signature

```
setServerPath(arg0:String) -> void
```

Signature générique

Paramètres

Nom	Type	Description
arg0	String	Spécifie le chemin de l'exécutable MapForce Server.

6.6.1.1.1.23 `setWorkingDirectory`

Définit le répertoire actuel pour les tâches en cours (les chemins relatifs seront évalués par rapport au répertoire de travail).

Signature

```
setWorkingDirectory(arg0:String) -> void
```

Signature générique

Paramètres

Nom	Type	Description
arg0	<code>String</code>	Spécifie le chemin vers le répertoire de travail.

6.6.1.2 MapForceServerException

La classe `MapForceServerException` fournit un accès programmatique aux exceptions lancées par la classe `MapForceServer`.

```
public class MapForceServerException extends Exception
```

7 Gestion du certificat numérique

La gestion du certificat numérique constitue un élément incontournable de l'échange de données sûr entre un ordinateur de client et un serveur web. Puisque les mappages peuvent être exécutés non seulement sur Windows avec MapForce, mais aussi sur un serveur Windows, Linux ou macOS avec MapForce Server (soit autonome, soit dans une exécution FlowForce Server), cette section se charge de la gestion des certificats HTTPS sur plusieurs plateformes.

Dans le contexte d'un HyperText Transport Protocol (HTTPS) de sécurité, il est important de distinguer entre les certificats de serveur et de client.

Certificats de serveur

Un certificat de serveur est ce qui identifie un serveur en tant qu'une entité de confiance par rapport à une application de client comme MapForce. Le certificat de serveur peut être auto-signé numériquement par une Autorité de Certificat commerciale, ou il peut être signé par votre organisation. Dans tous les cas, tout en créant le mappage dans MapForce, vous pouvez spécifier les paramètres suivants :

- Si le certificat de serveur doit être contrôlé.
- Si la requête doit être poursuivie en cas de détection de non-concordance entre le certificat de nom et le nom de l'hôte.

Ces paramètres sont disponibles dans le dialogue Paramètres de sécurité HTTP de MapForce. Lorsque vous activez les contrôles de certificat de serveur, veuillez considérer les éléments suivants :

- Si vous appelez un serveur Web dont le certificat est signé par une Autorité de Certificat de confiance, votre système d'exécution sera probablement déjà configuré pour faire confiance au certificat de serveur, et aucune configuration supplémentaire n'est nécessaire.
- Si vous appelez un serveur Web qui fournit un certificat auto-signé (par exemple, un serveur de réseau local dans le cadre de votre organisation), vous devrez configurer votre système d'exploitation et faire confiance à ce certificat.

Dans la plupart des cas, vous pouvez contrôler le niveau de confiance entre votre système d'exploitation et le serveur Web en saisissant l'URL du service Web dans la barre d'adresse du navigateur. Si le serveur n'est pas fiable, ou si votre système d'exploitation n'est pas configuré pour faire confiance au serveur, votre navigateur affichera un message comme "Cette connexion n'est pas fiable", ou "Il existe un problème avec le certificat de ce site Web". Veuillez noter que vous ne pouvez pas utiliser le navigateur pour contrôler le niveau de confiance avec un serveur Web si le navigateur utilise une base de données de certificat autre que celle du système d'exploitation (par exemple, Firefox sur Ubuntu).

Sur Windows, vous pouvez établir un lien de confiance avec le serveur en suivant les instructions du navigateur et en important ou en installant les certificats requis dans le magasin de Trusted Root Authorities de votre système (voir [Certificats de serveur de confiance sur Windows](#)¹⁵³). Sur macOS, vous pouvez procéder à l'opération équivalente dans Keychain Access (voir [Certificats de serveur de confiance sur macOS](#)¹⁵²). Pour des instructions applicables à Linux, voir [Certificats de serveur de confiance sur Linux](#)¹⁵¹.

Certificat de client

Alors que les certificats de serveur sont utilisés pour identifier un serveur en tant qu'une entité de confiance, les certificats de client sont principalement utilisés pour authentifier l'appelant par rapport au serveur Web server. Si vous avez l'intention d'appeler un serveur Web qui requiert les certificats client, vous devrez éventuellement contacter l'administrateur du serveur Web pour les instructions de configuration pour le client. Prenant IIS

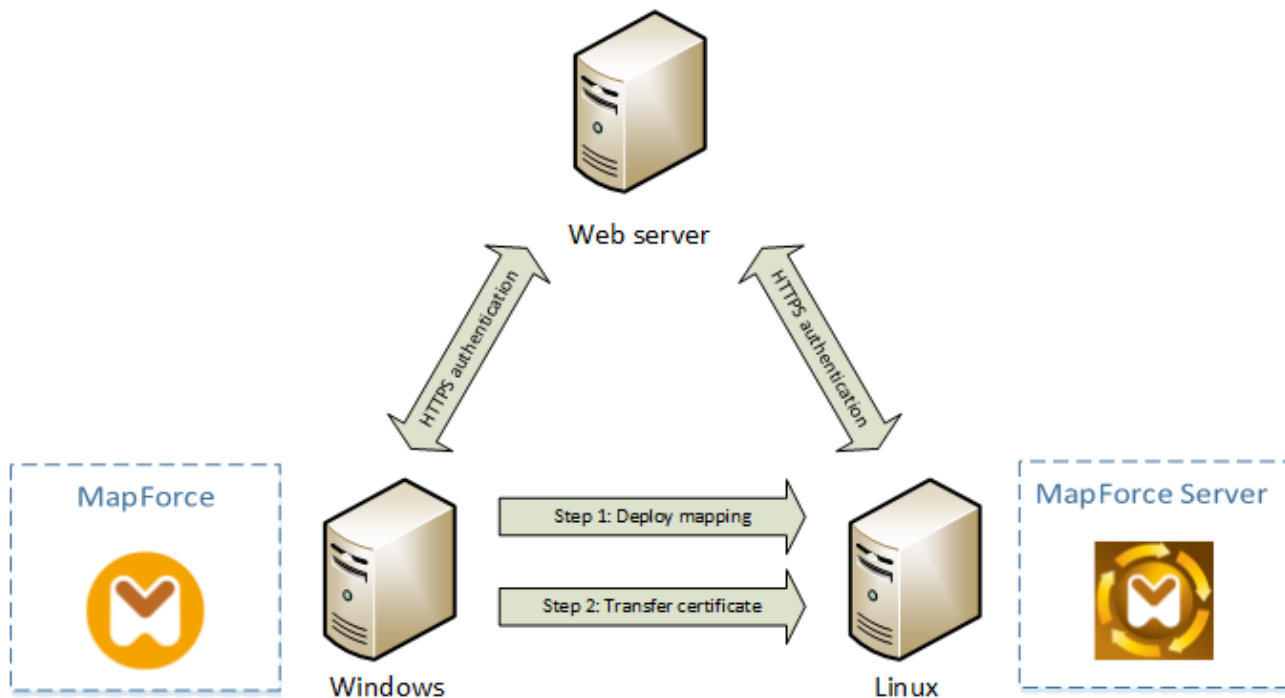
(Internet Information Services) en tant qu'exemple, le serveur Web peut être configuré pour gérer HTTPS et les certificats client de l'une des manières suivantes :

- Requérir HTTPS et ignorer les certificats client
- Requérir HTTPS et accepter les certificats client
- Requérir HTTPS et requérir les certificats client

Le succès ou l'échec de la requête de service Web dépend aussi bien la configuration du serveur Web que de l'application client. Par exemple, si le serveur Web est configuré pour requérir un certificat client, alors, pour que l'appel soit réussi, l'application d'appel doit présenter un certificat de client valide.

D'un point de vue d'une perspective MapForce, la même chose est valable pour les mappages qui comprennent des appels de service Web par le biais de HTTPS. En particulier, d'exécuter de tels mappages avec succès, on assume que le serveur Web a été configuré pour accepter ou requérir le certificat client, et que c'est le système d'exploitation sur lequel le mappage est exécuté qui fournit le certificat de client au serveur Web.

Le diagramme ci-dessous illustre un scénario où un certificat client utilisé dans MapForce est transféré sur un serveur Linux exécutant MapForce Server. Une fois que le certificat a été transféré sur le système d'exploitation cible, MapForce Server peut l'utiliser pour s'authentifier par rapport au serveur Web et exécuter le mappage avec succès.



Déploiement des mappages avec des certificats de client vers un autre ordinateur

Pour l'authentification HTTPS dans les appels de service Web, MapForce est capable d'utiliser Transport Layer Security (TLS) en plus de HTTP, qui est le successeur du protocole Secure Sockets Layer (SSL). Veuillez noter que le fallback vers SSL peut se produire soit lorsque l'implémentation de client soit lorsque le serveur ne prend pas en charge TLS.

Afin de prendre en charge les appels Web avec l'authentification de certificat client sur plusieurs plateformes, MapForce (et MapForce Server) tablent sur l'implémentation de gestion du certificat sur chaque plateforme, permettant ainsi d'assurer que la gestion de certificat est toujours incluse dans le système d'exploitation sous-jacent. Chaque système d'exploitation offre plusieurs supports pour la gestion du certificat, tel que montré dans la table ci-dessous.

Plateforme	Gestion et implémentation de certificat
Windows	<p>Sur Windows, vous pouvez gérer des certificats utilisant le snap-in Certificat (voir Accéder aux magasins de Certificat sur Windows¹⁵⁴).</p> <p>La prise en charge de TLS est disponible par le biais de la bibliothèque <i>Secure Channel</i> (aussi connu sous <i>SChannel</i>).</p>
Linux	<p>Sur Linux, vous pouvez gérer des certificats utilisant l'outil de ligne de commande et la bibliothèque OpenSSL (<code>openssl</code>). Si la prise en charge OpenSSL n'est pas déjà disponible sur l'appareil Linux où MapForce Server est installé, vous devrez le télécharger et l'installer avant de pouvoir gérer des certificats.</p> <p>La prise en charge de TLS est disponible par le biais de la bibliothèque OpenSSL (https://www.openssl.org/).</p>
macOS	<p>Sur macOS, vous pouvez gérer des certificats utilisant <i>Keychain Access Manager</i>, situé sous Finder > Applications > Utilities.</p> <p>La prise en charge de TLS est disponible par le biais de la bibliothèque <i>Secure Transport</i> native au système d'exploitation.</p>

Si vous exécutez le mappage sur un système d'exploitation Windows où vous pouvez déjà utiliser avec succès le même service Web avec lequel vous souhaitez appeler MapForce, aucune configuration de certificat supplémentaire n'est normalement requise (pour les conditions d'exécuter le mappage avec succès sur Windows, voir [Certificats client sur Windows](#)¹⁶⁵). Néanmoins, si vous créez des mappages avec MapForce sur un ordinateur Windows, pour les déployer ensuite sur un autre ordinateur (capable d'exécuter un système d'exploitation différent), le certificat client n'est pas stocké ou copié avec le package déployé. Pour que l'appel de service Web (et le mappage) soit exécuté avec succès, le certificat client doit aussi exister sur le système d'exploitation cible.

Pour transférer un certificat depuis un système Windows vers un autre ordinateur sur base Windows, exporter le certificat requis (avec une clé privée) depuis le système de source (voir [Exporter les certificats depuis Windows](#)¹⁶⁵). Ensuite importer le même certificat sur le magasin **Current User\Personal** dans le système d'exploitation cible (voir [Certificats client sur Windows](#)¹⁶⁵).

Pour consulter des instructions pour savoir comment transférer aux certificats client sur les plateformes Linux et macOS, voir [Certificats client sur Linux](#)¹⁶¹ et [Certificats client sur macOS](#)¹⁶³, respectivement.

7.1 Certificats de serveur de confiance sur Linux

Sur Linux vous pouvez importer un certificat de confiance dans la boutique de certificat du système tel que montré dans l'exemple ci-dessous. N'effectuez les étapes suivantes que si vous êtes sûr de l'authenticité du certificat du serveur Web.

Sur Debian et Ubuntu, suivre les étapes suivantes :

1. Copier le fichier de certificat du serveur Web dans le répertoire suivant.

```
sudo cp /home/downloads/server_cert.crt /usr/local/share/ca-certificates/
```

2. Mettre à jour le magasin de certificat comme suit :

```
sudo update-ca-certificates
```

Sur CentOS, suivre les étapes suivantes :

1. Installer le pack `ca-certificates` :

```
yum install ca-certificates
```

2. Activer la fonction de configuration d'autorité de certificat dynamique :

```
update-ca-trust enable
```

3. Copier le certificat de serveur dans le répertoire suivant :

```
cp server_cert.crt /etc/pki/ca-trust/source/anchors/
```

4. Utilisez la commande suivante :

```
update-ca-trust extract
```

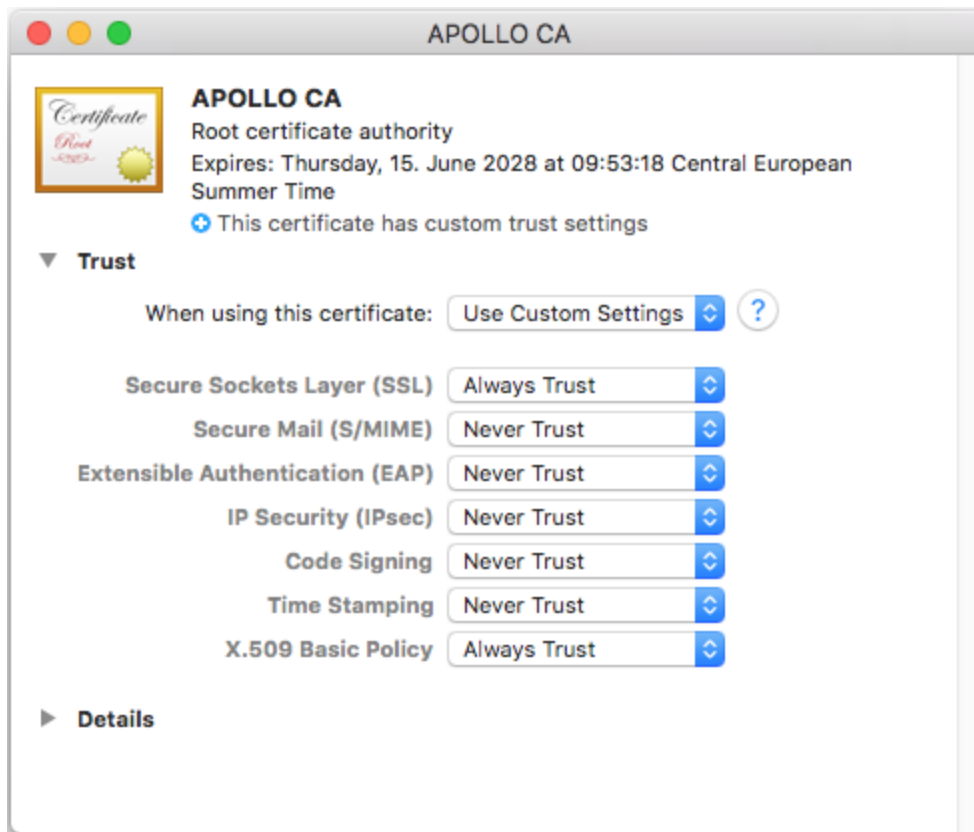
Dans les cas où vous devez accéder au serveur uniquement par le biais du navigateur, il suffit d'importer le certificat dans la boutique du certificat du navigateur. Les instructions exactes peuvent varier pour chaque navigateur. Par exemple, dans Firefox 59.0.2, vous pouvez suivre les instructions suivantes :

1. Sous **Options | Privacy & Security**, cliquer sur **View Certificates**.
2. Dans l'onglet **Authorities**, cliquer sur **Import** et chercher le fichier de certificat racine créé précédemment.
3. Lorsque vous y êtes invité, choisir **Trust this CA to identify websites** et cliquer sur **OK**

7.2 Certificats de serveur de confiance sur macOS

Sur macOS, vous pouvez importer un certificat de confiance dans Keychain Access comme suit :

1. Exécuter Keychain Access.
2. Cliquer sur **System**, puis sur **Certificates**.
3. Ouvrir le menu **Fichier** et cliquer sur **Importer Items**.
4. Chercher le certificat de confiance et cliquez sur **Open**.
5. Saisir le mot de passe Keychain Access lorsque l'invite apparaît, puis cliquer sur **Modify Keychain**.
6. Double-cliquer sur le certificat, agrandir la section *Trust*, et choisir **Always Trust**.



7.3 Certificats de serveur de confiance sur Windows

Sur Windows, vous pouvez importer un certificat de confiance dans la boutique des certificats du système comme suit :

1. Ouvrir la boutique des certificats Windows *pour le compte ordinateur*, voir [Accéder à la boutique de certificats Windows](#)¹⁵⁴.
2. Élargir le dossier *Trusted Root Certification Authorities* de l'arborescence *Certificats (ordinateur local)*, cliquez avec la touche de droite sur **Certificats**, sélectionnez **Toutes tâches | Importer** et suivez l'assistant d'importation du Certificat.

Pour plus d'information, voir l'article [Importer un Certificat sur le site web de Microsoft](#).

7.4 Accéder aux boutiques de certificat sur Windows

Sur Windows, vous pouvez gérer des certificats soit depuis le snap-in Certificates Microsoft Management Console (MMC), soit depuis l'Internet Explorer.

Pour ouvrir les Certificats snap-in pour l'*utilisateur Windows actuel*, exécutez la commande suivante dans la ligne de commande :

```
certmgr.msc
```

Pour ouvrir les Certificats snap-in pour le *compte de l'ordinateur*, suivez les étapes suivantes :

1. Exécuter `mmc` dans la ligne de commande.
2. Allez au menu **Fichier** du MMC et cliquez sur **Add/Remove Snap-in**.
3. Cliquez sur **Certificates**, puis cliquez sur **Ajouter**.
4. Sélectionnez **Compte d'ordinateur** et cliquez sur **Suivant**.
5. Sélectionnez **Ordinateur local** et cliquez sur **Terminer**.

7.5 Exporter des certificats depuis Windows

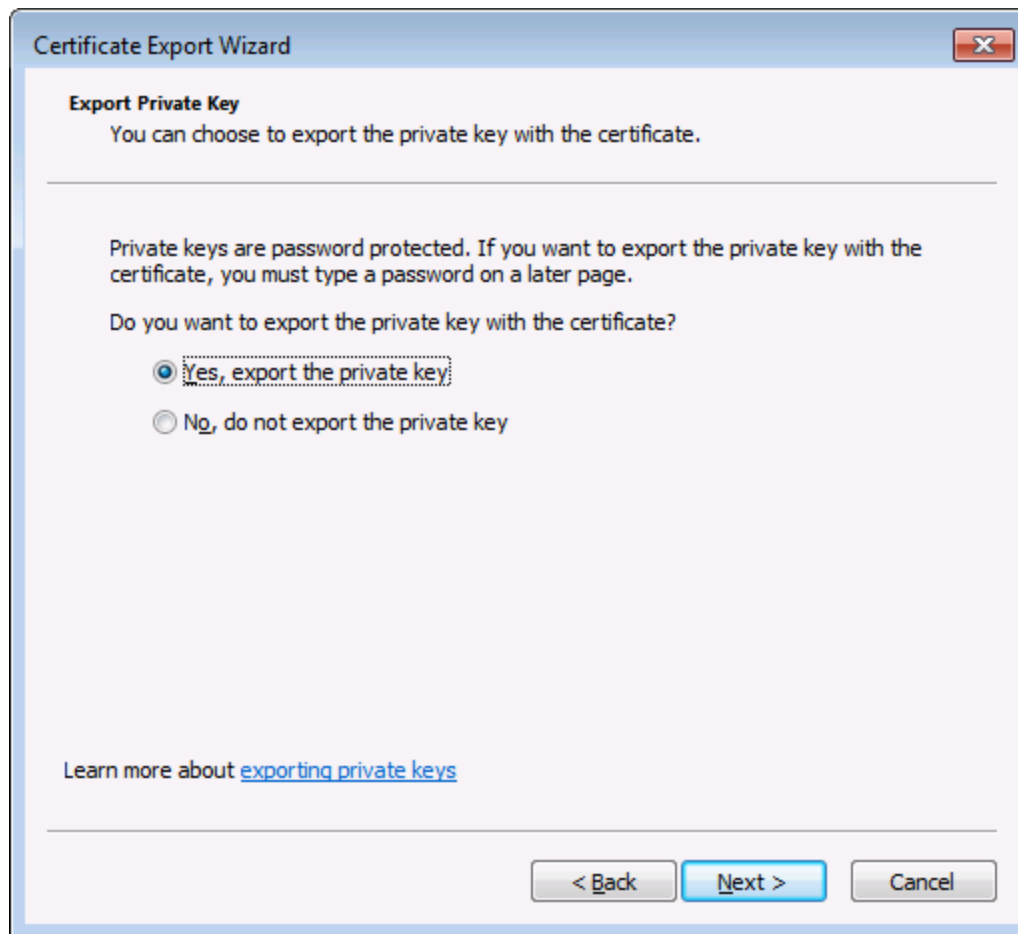
Concernant des mappages qui appellent des services Web via HTTPS et sont déployés sur un serveur macOS ou Linux exécutant MapForce Server ou FlowForce Server, le même certificat client doit être disponible sur le système d'exploitation non-Windows comme étant celui utilisé sur Windows pour concevoir et tester le mappage. Pour exécuter ces mappages sur un système d'exploitation non-Windows avec MapForce Server, exporter le certificat requis avec la clé privée depuis Windows puis l'importer dans le système d'exploitation cible.

Pour exporter un certificat avec une clé privée depuis Windows :

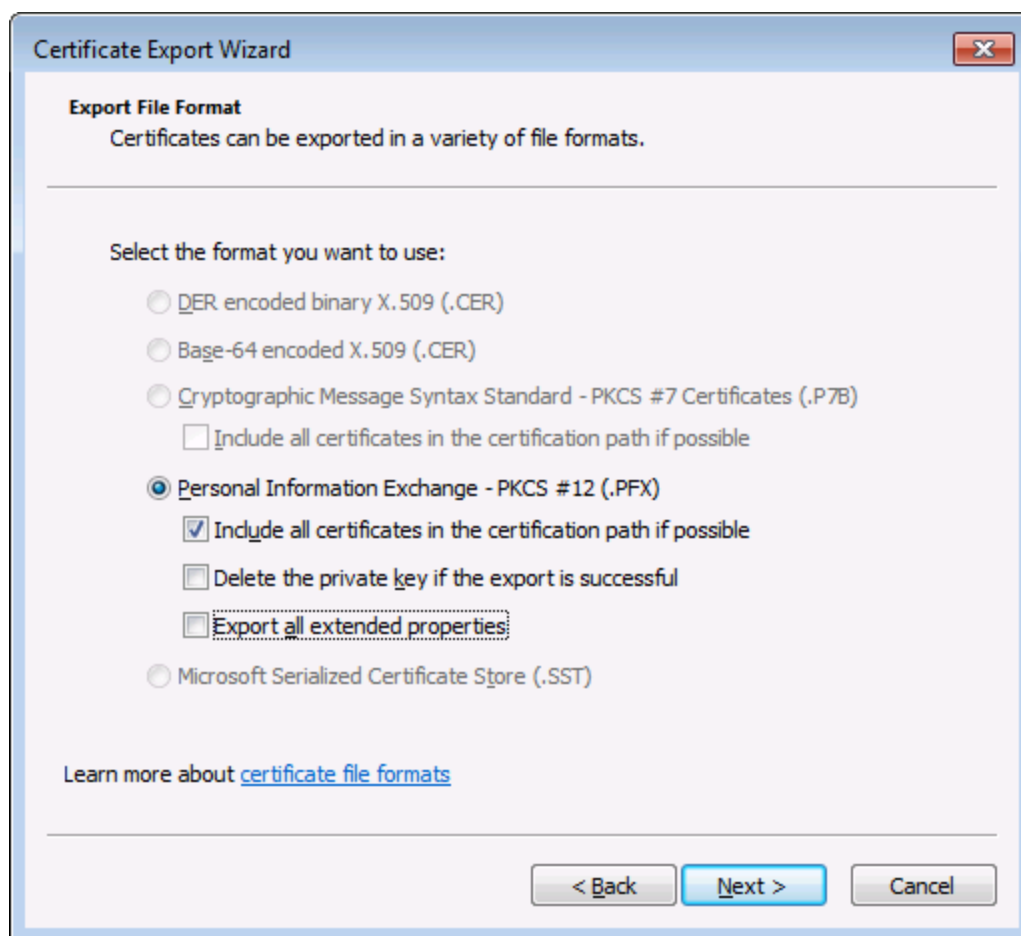
1. Sur Windows, ouvrir le snap-in Certificates (voir [Accéder aux magasins de Certificat sur Windows](#)¹⁵⁴).
2. Cliquer avec la touche de droite sur le certificat que vous souhaitez exporter, pointer sur **All Tasks**, puis cliquer sur **Export**.
3. Cliquer sur **Next**.



4. Choisir d'exporter depuis Windows le certificat avec sa clé privée, puis cliquer sur **Next**.

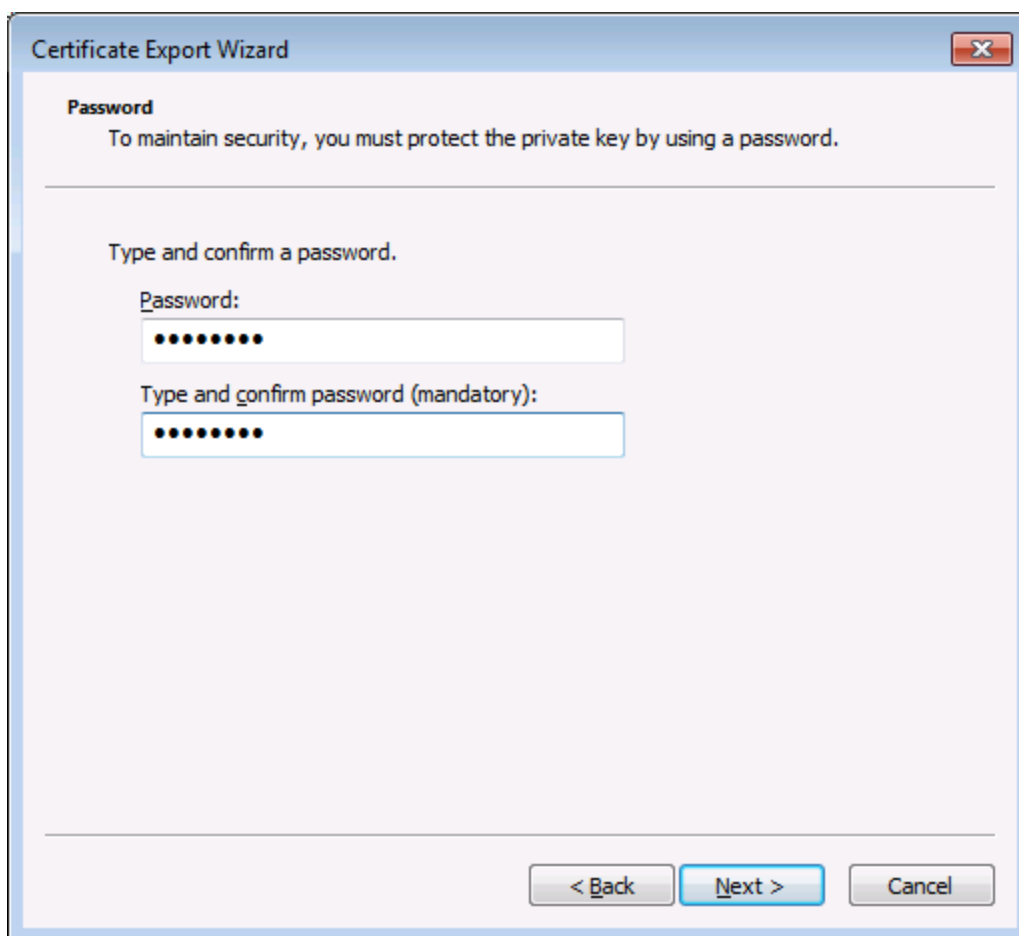


5. Choisir le format de fichier *Personal Information Exchange - PKCS #12 (.pfx)*, puis cliquer sur **Next**.

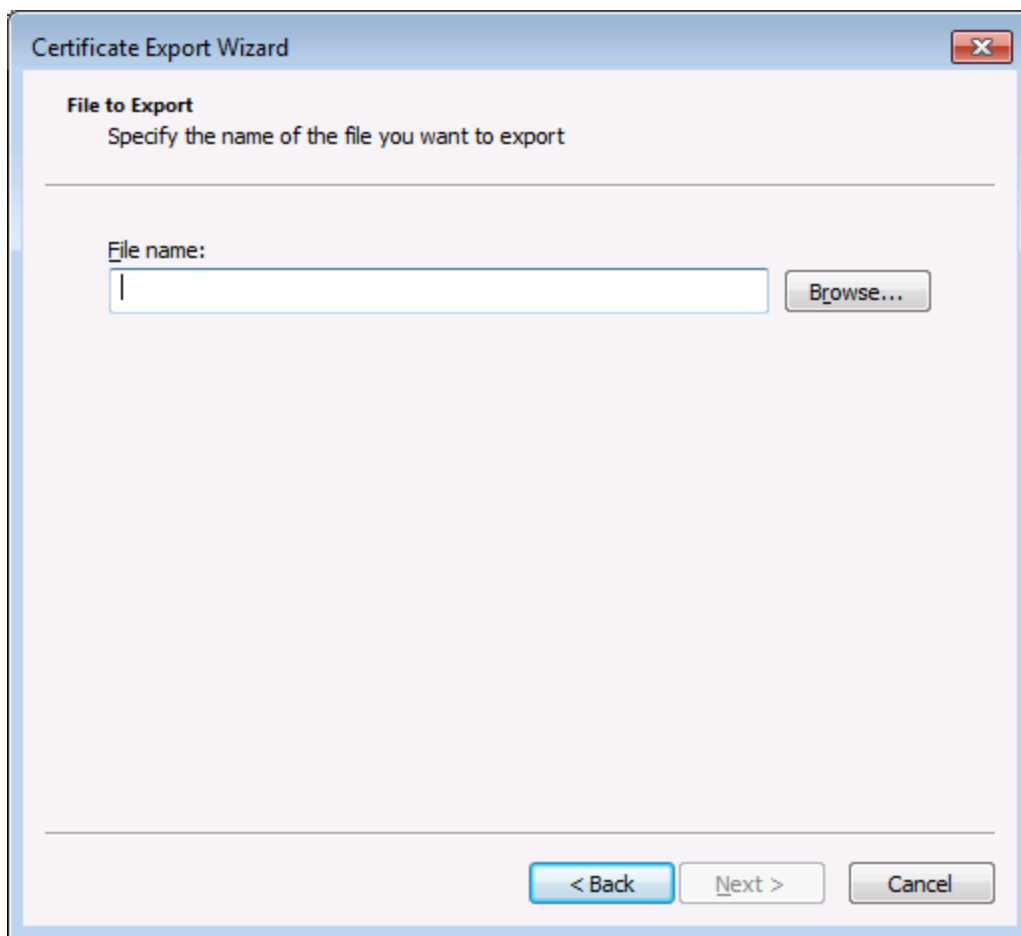


Note : Veuillez vous assurer de ne pas choisir l'option **Delete the private key if the export is successful**, autrement vous ne pourrez pas utiliser le certificat après son exportation.

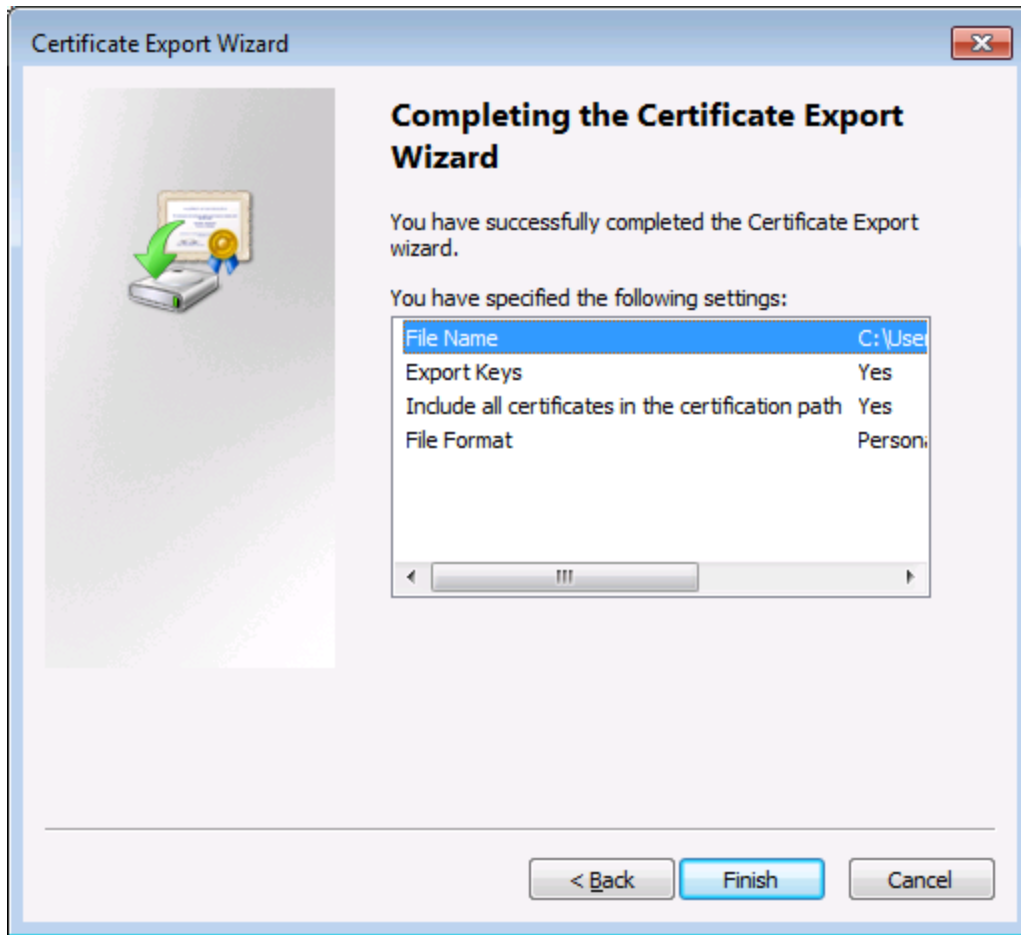
6. Saisir un mot de passe, puis cliquer sur **Next**. Vous nécessitez ce mot de passe une fois avoir copié le certificat sur le système d'exploitation cible.



7. Chercher l'emplacement du fichier à exporter, puis cliquer sur **Next**.



8. Cliquer sur **Finish**.



7.6 Certificats client sur Linux

Si vos mappages comprennent une authentification de service Web par le biais de certificats de client HTTPS, suivez ces étapes pour déployer ces mappages sur un appareil Linux exécutant MapForce Server :

1. Ouvrir dans MapForce le mappage qui appelle le service Web.
2. Double-cliquer sur l'en-tête du composant de Service Web. La fenêtre de dialogue des Paramètres de composant apparaît.
3. Cliquer sur **Edit** à côté des Paramètres de sécurité HTTP.
4. Dans la fenêtre de dialogue de HTTP Security Settings, cliquer sur **Client Certificate**, puis choisir le certificat requis à partir du magasin **Current User\Personal** sur Windows .
5. Enregistrer et déployer le mappage sur le système d'exploitation cible.
6. Transférer le certificat de client requis par l'appel de service Web vers le système d'exploitation cible. Assurez-vous que le certificat a une clé privée, et que la propriété **Enhanced Key Usage** du certificat comprend "Client authentication" en tant qu'objectif

Pour transférer le certificat client sur Linux :

1. Exporter le certificat client avec une clé privée à partir de Windows, dans le format de fichier *Personal Information Exchange - PKCS #12 (.pfx)* (voir [Exporter des certificats depuis Windows](#)¹⁵⁵).
2. Copier le fichier de certificat sur la machine Linux.
3. Convertir le fichier .pfx sur le format .pem en utilisant la commande :

```
openssl pkcs12 -in cert.pfx -out "John Doe.pem" -nodes
```

Cette commande parse le fichier .pfx et émet un fichier .pem, sans encoder la clé privée. Les certificats avec clé privée encodée envoient une invite de mot de passe et ne sont pas pris en charge dans l'exécution de serveur.

Exécuter le mappage

Pour instruire MapForce Server pour utiliser le fichier .pem en tant que certificat client, définir le paramètre `--certificatespath` lors de l'exécution du mappage. Le paramètre `--certificatespath` définit le chemin du répertoire où tous les certificats requis par le mappage actuel sont stockés. Par exemple, si le chemin du certificat est `/home/John/John Doe.pem`, alors `--certificatespath` doit être défini sur `/home/John`.

Par défaut, si le paramètre `--certificatespath` n'est pas fourni, MapForce Server cherche des certificats dans le répertoire `$HOME/.config/altova/certificates` de l'utilisateur actuel.

Pour que le mappage soit exécuté correctement, le fichier de certificat devrait présenter l'extension .pem et le nom de fichier doit correspondre au Common Name (CN) du certificat, y compris les espaces (par exemple, **John Doe.pem**). Si le CN contient une barre oblique (/), elle doit être remplacée avec un caractère de tiret bas (_).

Si vous avez l'intention d'exécuter le mappage en tant que tâche FlowForce Server, copier le fichier de certificat dans le répertoire `$HOME/.config/altova/certificates`. Lors de l'exécution de la tâche, FlowForce Server utilisera le répertoire pour chercher tout fichier de certificat requis par le mappage.

Pour des considérations de sécurité, assurez-vous que les fichiers de certificat ne sont pas lisibles par d'autres utilisateurs, puisqu'ils contiennent des informations sensibles.

7.7 Certificats client sur macOS

Si vos mappages comprennent une authentification de service Web par le biais de certificats de client HTTPS, suivez ces étapes pour déployer ces mappages sur un appareil macOS exécutant MapForce Server :

1. Ouvrir dans MapForce le mappage qui appelle le service Web.
2. Double-cliquer sur l'en-tête du composant de Service Web. La fenêtre de dialogue des Paramètres de composant apparaît.
3. Cliquer sur **Edit** à côté des Paramètres de sécurité HTTP.
4. Dans la fenêtre de dialogue de HTTP Security Settings de MapForce, cliquer sur **Client Certificate**, puis choisir le certificat requis à partir du magasin **Current User\Personal** sur Windows .
5. Si le nom du certificat n'est pas parfaitement identique au nom de l'hôte du serveur, choisir **Allow name mismatch between certificate and request**.
6. Enregistrer et déployer le mappage sur le système d'exploitation cible .
7. Transférer le certificat de client requis par l'appel de service Web vers le système d'exploitation cible. Assurez-vous que le certificat a une clé privée, et que la propriété **Enhanced Key Usage** du certificat comprend "Client authentication" en tant qu'objectif.

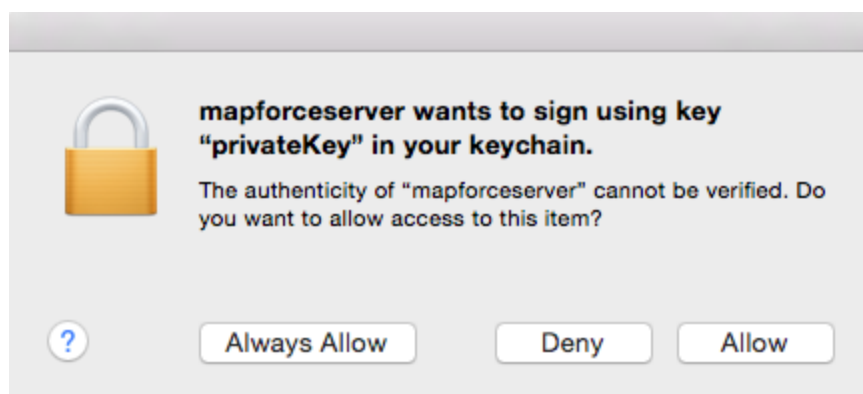
Pour transférer le certificat client sur macOS :

1. Exporter le certificat client avec une clé privée à partir de Windows, dans le format de fichier *Personal Information Exchange - PKCS #12 (.pfx)* (voir [Exporter des certificats depuis Windows](#)¹⁵⁵) et copier le fichier .pfx vers macOS.
2. Si cela n'a pas encore été fait, veuillez vous assurer que le système d'exploitation fasse confiance au certificat de serveur (voir [Certificats de serveur de confiance sur Mac OS](#)¹⁵²).
3. Exécuter l'accès à la Keychain depuis **Finder > Applications > Utilities**.
4. Dans le menu **File**, cliquer sur **Import Items**.
5. Chercher le certificat client exporté depuis Windows dans l'étape 1 et sélectionner une keychain de destination.
6. Cliquer sur **Open** et saisir le mot de passe avec lequel le certificat a été encodé.

Exécuter le mappage

Vous êtes maintenant prêt à exécuter le mappage en utilisant la commande `run` de MapForce Server. Veuillez noter les éléments suivants :

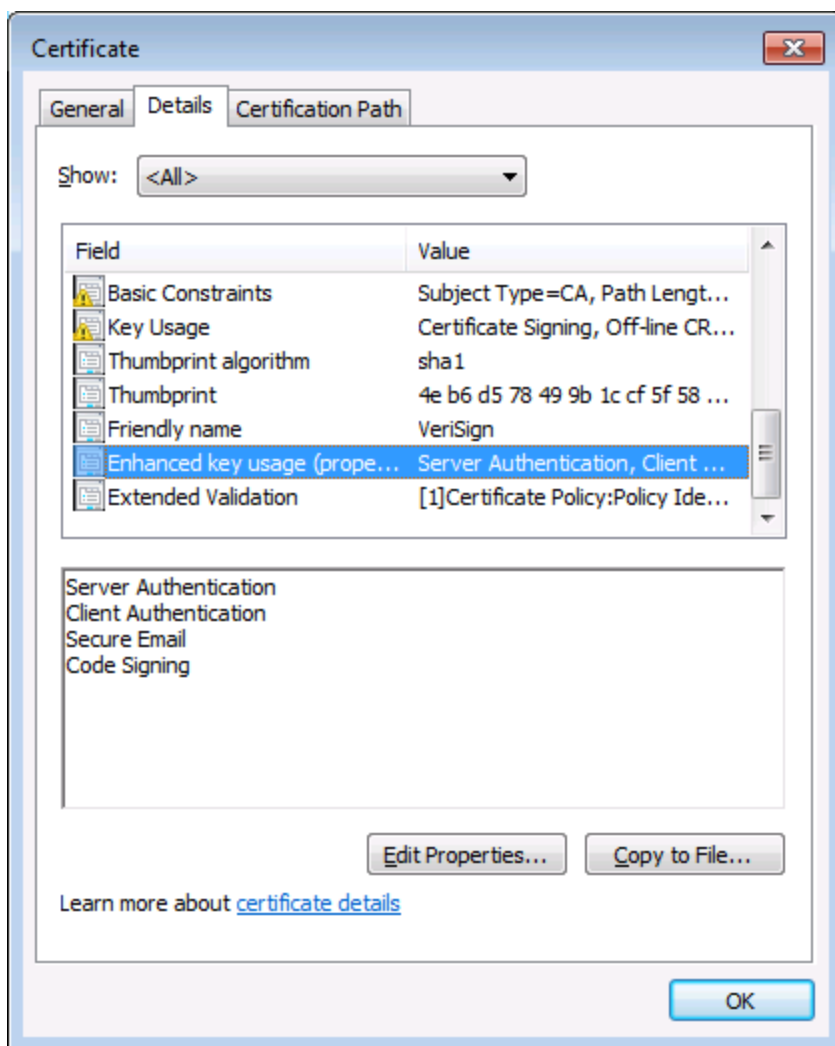
- Si vous exécutez le mappage à distance through SSH, tout d'abord, déverrouiller la keychain avec la commande `security unlock-keychain`.
- Si vous exécutez le mappage par le biais de l'interface utilisateur graphique macOS, lorsque vous serez invité à permettre à MapForce Server l'accès à la keychain, cliquer sur **Allow**.



7.8 Certificats client sur Windows

Lorsque vous exécutez un mappage sur Windows qui requiert des certificats de client, les conditions pour exécuter le mappage avec succès sont les suivantes :

- Le certificat client doit exister dans le magasin de certificat **Current User\Personal** (aussi référé en tant que **My** magasin). Pour que le certificat puisse exister dans ce magasin, il doit être importé par le biais de l'assistant d'Importation de certificat. Pour plus d'instructions, voir [https://technet.microsoft.com/en-us/library/cc754489\(v=ws.11\).aspx](https://technet.microsoft.com/en-us/library/cc754489(v=ws.11).aspx).
- Le certificat doit avoir une clé privée.
- La propriété **Enhanced Key Usage** d'un certificat doit inclure "Client authentication" en tant qu'objectif.



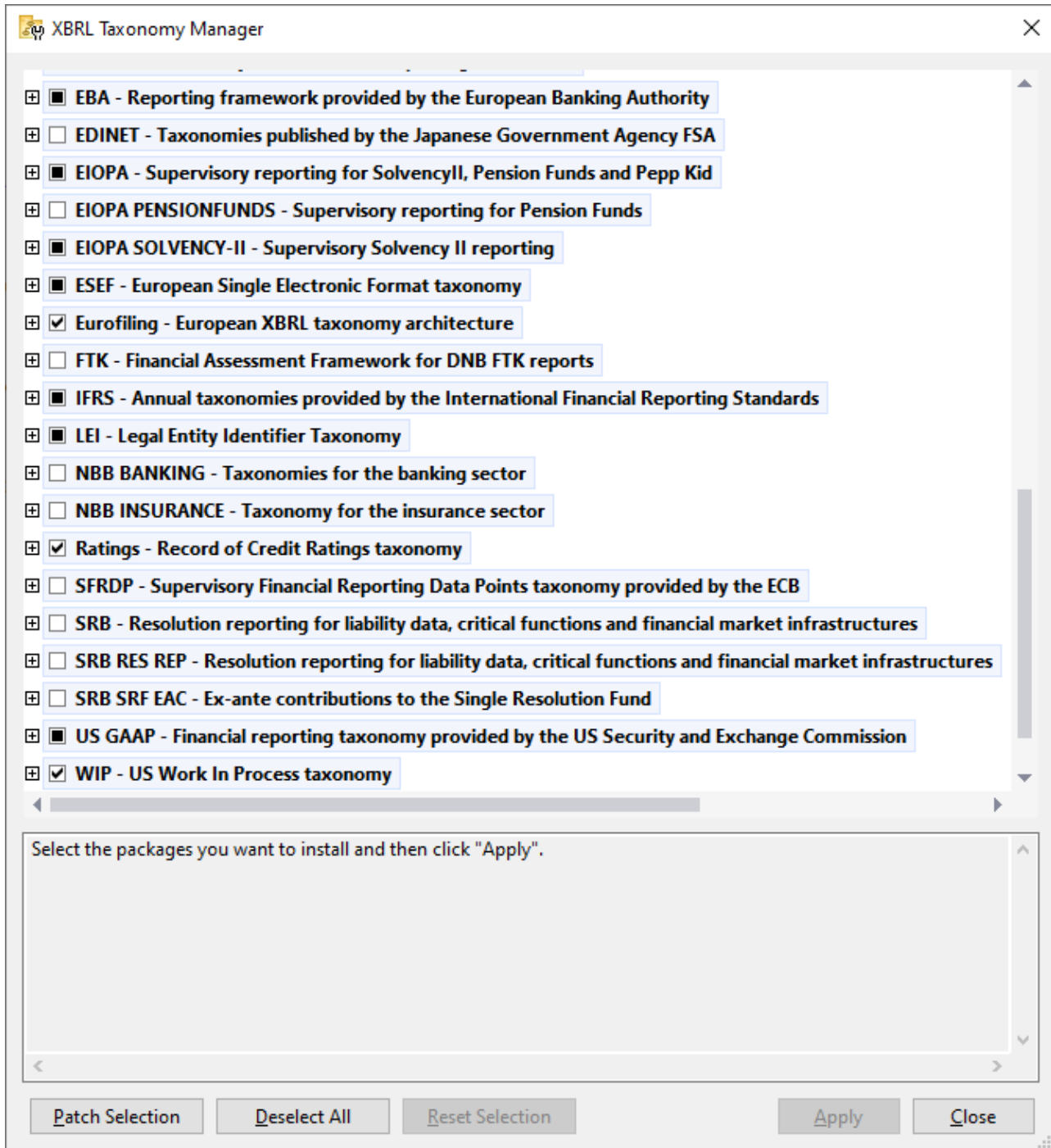
Dans la version actuelle de MapForce, en raison d'une limitation de la bibliothèque par MapForce, Windows choisira le certificat requis automatiquement depuis le magasin de certificat lorsque vous exécutez le mappage. Le mappage sera exécuté avec succès, après avoir filtré le magasin de certificat **Current User\Personal**, le serveur trouve un certificat convenable. Veuillez noter que l'authentification HTTPS (et l'opération de sélection du certificat) est gérée par Windows et n'est pas contrôlée par MapForce ou MapForce Server. Dans certains cas, si plusieurs certificats existent dans le magasin **Current User\Personal**, un

certificat non convenable peut être choisi automatiquement par le système d'exploitation, ce qui entraîne l'échec de l'exécution de mappage. Cette situation peut être évitée en limitant le nombre de certificats disponibles dans le magasin **Current User\Personal**.

7.9 Gestionnaire de taxonomie

XBRL Gestionnaire de taxonomie est un outil qui propose un moyen centralisé d'installer et de gérer des taxonomies XBRL pour une utilisation sur toutes les applications XBRL Altova, y compris MapForce Server

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



Les applications activées par XBRL d'Altova

Applications desktop (Windows uniquement)	Applications de serveur (Windows, Linux, macOS)
---	---

Altova XBRL Add-ins pour Excel (EBA, Solvency II, II)	MapForce Server (Standard et Advanced Edition).
MapForce Enterprise Edition	RaptorXML+XBRL Server
StyleVision Enterprise Edition	StyleVision Server
XMLSpy Enterprise Edition	

Installation et désinstallation de Gestionnaire de taxonomie

Gestionnaire de taxonomie est installé automatiquement quand vous installez d'abord une nouvelle version de l'Altova Mission Kit Enterprise Edition ou toute application activée par XBRL d'Altova (voir la table ci-dessus).

De même, il est supprimé automatiquement lorsque vous désinstallez la dernière application activée pour XBRL Altova depuis votre ordinateur.

Fonctions <% TAXON-MANAGER%>

Gestionnaire de taxonomie propose les fonctions suivantes :

- Consulter les taxonomies XBRL installées sur votre ordinateur, et contrôlez si de nouvelles versions sont disponibles pour le téléchargement.
- Télécharger des versions plus récentes des taxonomies XBRL indépendamment du cycle de release des produits Altova. (Altova stocke des taxonomies en ligne et vous pouvez les télécharger via Gestionnaire de taxonomie.)
- Installer ou désinstaller une des versions d'une taxonomie donnée (ou toutes les versions si nécessaire).
- Une taxonomie XBRL peut avoir des dépendances sur d'autres taxonomies. Lorsque vous installez ou désinstallez une taxonomie particulière, Gestionnaire de taxonomie vous informe sur d'autres taxonomies dépendantes et les installera ou désinstallera également automatiquement.
- Gestionnaire de taxonomie utilise le mécanisme du [catalogue XML](#) pour mapper les références de schéma aux fichiers locaux. Dans le cas de taxonomies XBRL larges, le traitement sera plus rapide que si les taxonomies étaient à un emplacement à distance.
- Toutes les taxonomies majeures sont disponibles via Gestionnaire de taxonomie et sont régulièrement mises à jour pour les dernières versions. Ceci vous fournit une ressource unique pour gérer toutes vos taxonomies et les mettre à disposition de toutes les applications activées par XBRL d'Altova.
- Les changements réalisés dans Gestionnaire de taxonomie prennent effet pour tous les produits d'Altova sur cet appareil.

Taxonomies XBRL personnalisées

Si vous devez travailler avec des taxonomies XBRL personnalisées qui ne sont pas incluses dans Gestionnaire de taxonomie, vous pouvez ajouter ces taxonomies à l'ensemble des packs personnalisés référencés par <% appName%>. Pour ce faire, suivez les étapes suivantes :

- *Dans les applications desktop d'Altova* : Sélectionnez la commande de menu **Outils | Options** et allez à la section **XBRL | Packages de taxonomie**. Recherchez le package ZIP de votre taxonomie XBRL personnalisée. Pour plus d'information, voir la description de cette commande dans votre documentation de produit desktop.
- *Dans les applications de serveur Altova* : Lorsque vous exécutez des commandes de la ligne de commande qui appuie les taxonomies personnalisées, fournissez l'option `--taxonomy-package of --`

`taxonomy-package-config-file`. Par exemple : Dans RaptorXML+XBRL Server, ces options sont appuyées par les commandes de validation XBRL telles que `valxbrl` ou `valxbrltaxonomy` ; dans MapForce, elles sont prises en charge par la commande d'exécution.

Comment cela fonctionne ?

Altova stocke toutes les taxonomies XBRL utilisées dans les produits Altova en ligne. Ce référentiel est mis à jour lorsque de nouvelles versions des taxonomies sont publiées. Gestionnaire de taxonomie affiche des informations sur les dernières taxonomies disponibles lorsqu'elles sont appelées dans son formulaire GUI de même que sur CLI. Vous pouvez ensuite installer, mettre à jour ou désinstaller les taxonomies via Gestionnaire de taxonomie.

Gestionnaire de taxonomie installe également les taxonomies d'une autre manière. Sur le site web d'Altova (<https://www.altova.com/taxonomy-manager>), vous pouvez sélectionner une taxonomie et ses taxonomies dépendantes que vous souhaitez installer. Le site web préparera un fichier de type `.altova_taxonomies` pour le téléchargement qui contient des informations sur la sélection de taxonomie. Lorsque vous double-cliquez sur ce fichier ou le passez à Gestionnaire de taxonomie via CLI comme argument de la commande `installer`¹⁷⁹, Gestionnaire de taxonomie installera les taxonomies que vous avez sélectionnées.

Cache local : suivre vos taxonomies

Toutes les informations sur les taxonomies installées sont suivies dans un répertoire cache centralisé sur votre ordinateur, situé ici :

<i>Fenêtres</i>	C:\ProgramData\Altova\pkgs\cache
<i>Linux</i>	/var/opt/Altova/pkgs/cache
<i>macOS</i>	/var/Altova/pkgs

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

- À chaque fois que vous lancez Gestionnaire de taxonomie.
- Lorsque vous exécutez MapForce Server pour la première fois dans un jour donné du calendrier.
- Si MapForce Server est ouvert plus de 24 heures, le cache est mis à jour toutes les 24 heures.
- Vous pouvez aussi mettre à jour le cache en exécutant la commande de `mise à jour`¹⁸² dans l'interface de ligne de commande.

Pour cette raison, le cache permet à Gestionnaire de taxonomie de suivre continuellement vos taxonomies installées par rapport aux taxonomies disponibles en ligne sur le site web d'Altova.

Ne modifiez pas le cache manuellement !

Le répertoire de cache local est entretenu automatiquement sur la base des taxonomies que vous installez ou désinstallez ; il ne doit pas être modifié ou supprimé manuellement. Il ne devrait pas être altéré ou supprimé manuellement. Si vous êtes amené à réinitialiser Gestionnaire de taxonomie à son état original "intact", alors, sur l'interface de la ligne de commande (CLI) : (i) exécutez la commande `reset`¹⁸⁰, et (ii) exécutez la commande `initialize`¹⁷⁸. (En alternative, exécutez la commande `reset` avec l'option `--i`.)

Proxy HTTP

Vous pouvez utiliser un proxy HTTP pour les connexions Gestionnaire de taxonomie. Les paramètres Proxy du système seront utilisés.

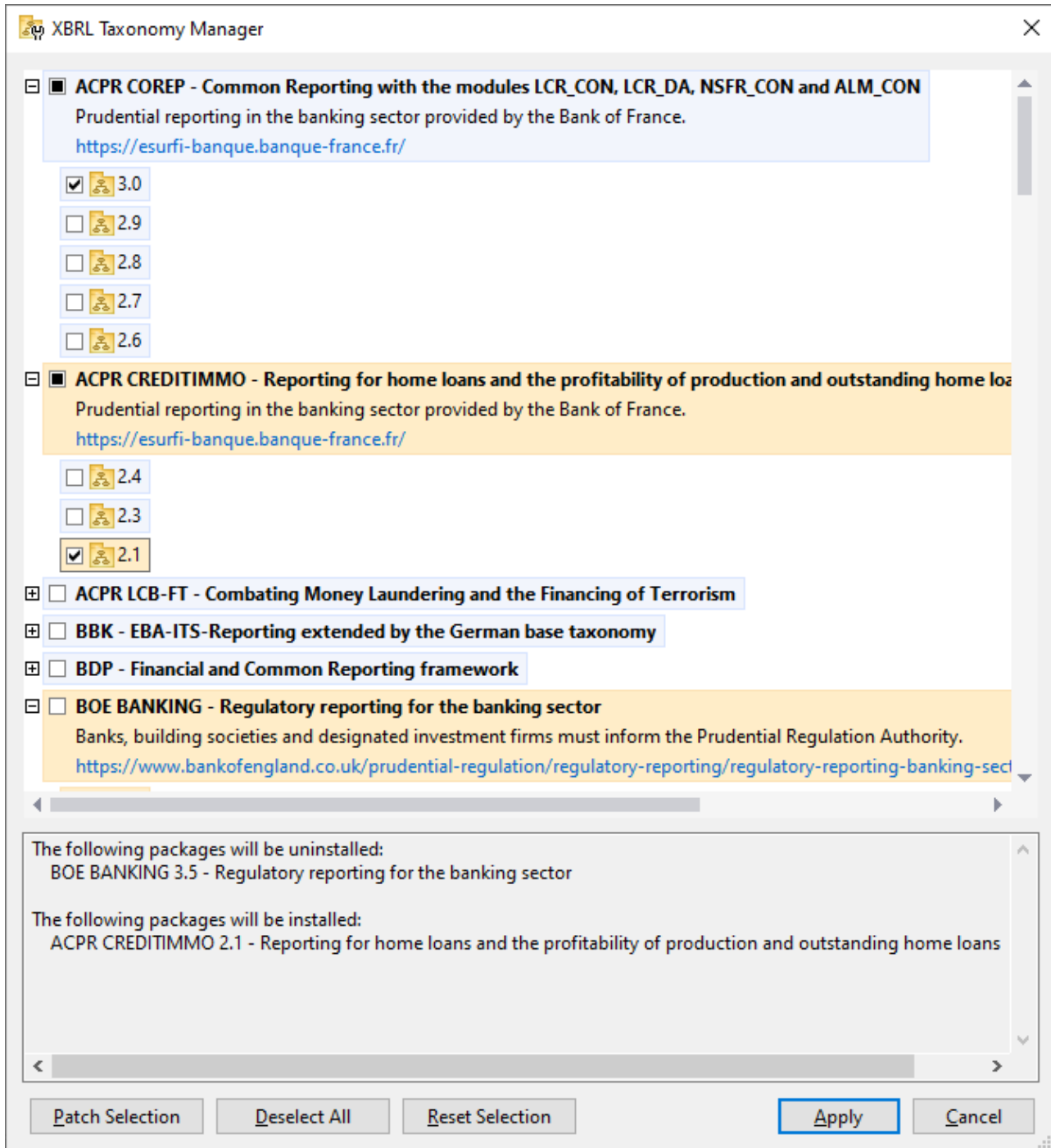
7.9.1 Exécuter le Gestionnaire de taxonomie

Interface utilisateur graphique

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

- *Durant l'installation de MapForce Server:* Vers la fin de la procédure d'installation, sélectionnez la case à cocher *Invoke Altova Taxonomy Manager* pour accéder au Gestionnaire de taxonomie XBRL directement. Ceci vous permettra d'installer les taxonomies au cours de la procédure d'installation de votre application Altova.
- Via le fichier `.altova_taxonomies` téléchargé du [Centre de téléchargement de la taxonomie XBRL d'Altova](#): Double-cliquez sur le fichier téléchargé pour exécuter la GUI de Gestionnaire de taxonomie, qui sera configurée pour installer les taxonomies que vous avez sélectionnées (le site web) pour installation.

Une fois que Gestionnaire de taxonomie GUI (*capture d'écran ci-dessous*) a été ouvert, les taxonomies déjà installées seront affichées telles sélectionnées. Si vous voulez installer une taxonomie supplémentaire, sélectionnez-la. Si vous voulez désinstaller une taxonomie déjà installée, désélectionnez-la. Une fois que vous avez faits vos sélections et/ou désélections, vous êtes prêts pour appliquer vos changements. Les taxonomies qui seront installées ou désinstallées seront mises en surbrillance et un message sur les modifications à venir sera posté dans le volet Messages au niveau inférieur de la fenêtre Gestionnaire de taxonomie (*voir la capture d'écran*).



Interface de ligne de commande

Vous pouvez exécuter Gestionnaire de taxonomie depuis une interface de ligne de commande en sélectionnant son fichier exécutable, `taxonomymanager.exe`.

Le fichier `taxonomymanager.exe` est situé dans le dossier suivant :

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

Vous pouvez alors utiliser toute commande de la section de référence de la commande CLI.

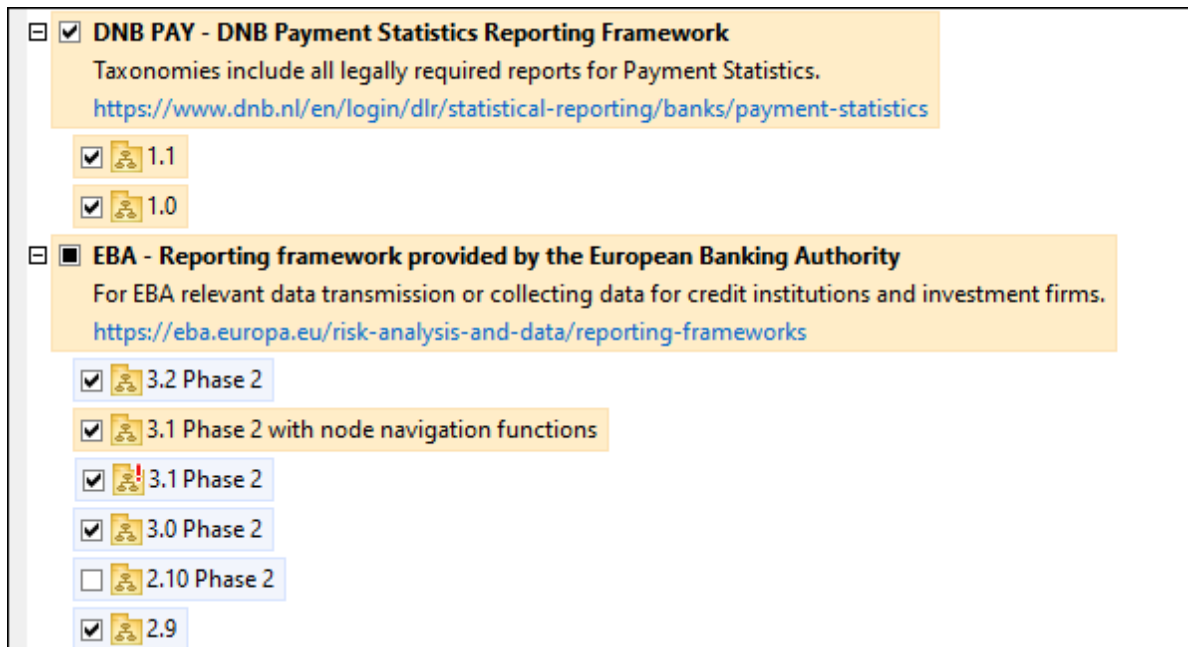
Pour afficher l'aide pour la commande, exécutez l'étape suivante :


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

7.9.2 Catégories de statut

Gestionnaire de taxonomie catégorise les taxonomies sous sa gestion comme suit :

- *Installer taxonomies*. Celles-ci sont affichées dans la GUI avec leurs cases à cocher sélectionnées (dans la capture d'écran ci-dessous, les versions cochées ci-de la taxonomie DNB et EBA sont des taxonomies installées). Si toutes les versions de taxonomie sont sélectionnées, alors la marque de sélection est une coche. Si au moins une version est décochée, alors la coche de sélection est un carré coloré plein. Vous pouvez décocher une taxonomie installée pour la **désinstaller**.
- *Désinstaller taxonomies disponibles*. Elles sont affichées dans la GUI avec leurs cases à cocher non sélectionnées. Vous pouvez sélectionner les taxonomies que vous souhaitez **installer**.



- Les *taxonomies pouvant être mises à niveau* sont celles qui ont été revues par leurs émetteurs depuis qu'elles ont été installées. Elles sont indiquées par leur GUI par une icône  (voir la capture d'écran ci-dessus). Vous pouvez **retoucher** la taxonomie installée avec la révision disponible.

Points à noter

- Dans la capture d'écran ci-dessus, les taxonomies DNB et quelques unes des taxonomies EBA sont cochées. Celles avec un arrière-plan bleu sont déjà installées. Celles avec un arrière-plan jaune sont désinstallées et ont été sélectionnées pour l'installation. Notez que (i) la taxonomie EBA 2.10 Phase 2 n'est pas installée et n'a pas été sélectionnée pour l'installation, (ii) la taxonomie EBA 3.1 Phase 2 a été installée, mais elle a été retouchée par son émetteur depuis qu'elle a été installée et le correctif n'a pas encore été installé.
- Lorsque vous exécutez Gestionnaire de taxonomie depuis la ligne de commande, la commande [list](#)¹⁷⁹ est utilisée avec différentes options pour recenser les différentes catégories de taxonomies :

<code>taxonomymanager.exe list</code>	Recense toutes les taxonomies installées et disponibles ; celles pouvant être mises à niveau sont également indiquées.
<code>taxonomymanager.exe list -i</code>	Recense les taxonomies installées uniquement ; celles pouvant être mises à niveau sont également indiquées
<code>taxonomymanager.exe list -u</code>	Recense les taxonomies pouvant être mises à niveau




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

7.9.3 Retoucher ou Installer une taxonomie

Retoucher une taxonomie installée

Occasionnellement, des taxonomies XBRL peuvent recevoir des patches (mises à niveau ou révisions) depuis leurs émetteurs. Lorsque Gestionnaire de taxonomie détecte que des patches sont disponibles, ceux-ci sont indiqués dans les listes de taxonomie Gestionnaire de taxonomie et vous pouvez installer des patches rapidement.

Dans la GUI

Les patches sont indiqués par l'icône . (Voir aussi la rubrique précédente sur les [catégories de statut](#)¹⁷³.) Si les patches sont disponibles, le bouton **Patch Selection** sera activé. Cliquez dessus pour sélectionner et préparer tous les patches pour installation. Dans la GUI, l'icône de chaque taxonomie sera patchée de  à , et le volet des Messages en bas du dialogue recense les patches qui doivent être appliqués. Lorsque vous êtes prêt pour installer des patches sélectionnés, cliquez sur **Appliquer**. Tous les correctifs seront appliqués ensemble. Notez que si vous décochez une taxonomie marquée pour une correction, réellement vous désinstallerez cette taxonomie.

Sur le CLI

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

1. Exécuter la commande [list -u](#)¹⁷⁹ . Cela liste des taxonomies lorsque des mises à niveau de patch sont disponibles.
2. Exécuter la commande [upgrade](#)¹⁸² pour installer les patches.

Installer une taxonomie disponible

Vous pouvez installer des taxonomies en utilisant soit la GUI Gestionnaire de taxonomie ou en envoyant Gestionnaire de taxonomie les instructions d'installation via la ligne de commande.

Note : Si la taxonomie actuelle référence d'autres taxonomies, les taxonomies référencées sont aussi installées.

Dans la GUI

Pour installer une taxonomie utilisant la GUI Gestionnaire de taxonomie GUI, sélectionnez les taxonomies que vous voulez installer et cliquez sur **Appliquer**.

Vous pouvez aussi sélectionner les taxonomies que vous voulez installer sur le [site web d'Altova](#) et générer un fichier téléchargeable `.altova_taxonomies`. Lorsque vous double-cliquez sur ce fichier, il ouvrira Gestionnaire de taxonomie avec les taxonomies que vous vouliez présélectionner. La seule chose qui vous reste à faire, c'est cliquer sur **Appliquer**.

Sur le CLI

Pour installer des taxonomies via la ligne de commande, exécutez la commande [installer](#)¹⁷⁹ :

```
taxonomymanager.exe install [options] Taxonomie+
```

où `Taxonomie` est la taxonomie (ou les taxonomies) que vous voulez installer ou un fichier `.altova_taxonomies`. Une taxonomie est référencée par un identifiant de format `<name>-<version>`. (Les identifiants de taxonomies sont affichés quand vous exécutez la commande [list](#)¹⁷⁹.) Vous pouvez saisir autant de taxonomies que vous le souhaitez. Pour plus de détails, voir la description de la commande [install](#)¹⁷⁹.

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

Installer une taxonomie requise

Lorsque vous exécutez une commande activée par XBRL dans MapForce Server, et que MapForce Server découvre qu'une taxonomie dont elle a besoin pour exécuter la commande n'est pas présente ou est incomplète, Gestionnaire de taxonomie sera affiché l'information sur la taxonomie manquante. Vous pouvez ensuite installer directement toute taxonomie manquante via Gestionnaire de taxonomie.

Dans le Gestionnaire de taxonomie GUI, vous pouvez consulter toutes les taxonomies précédemment installées à tout moment en exécutant Gestionnaire de taxonomie de **Outils | Gestionnaire de taxonomie**.

7.9.4 Désinstaller une taxonomie, Réinitialiser

Désinstaller une taxonomie

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

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

Dans la GUI

Pour désinstaller les taxonomies utilisant la GUI Gestionnaire de taxonomie, effacez leurs cases à cocher et cliquez sur **Appliquer**. Les taxonomies sélectionnées et leurs taxonomies référencées seront désinstallées.

Pour désinstaller les taxonomies, cliquez sur **Désélectionner tout** et cliquez sur **Appliquer**.

Sur le CLI

Pour désinstaller des taxonomies via la ligne de commande, exécutez la commande `désinstaller` :

```
taxonomymanager.exe uninstall [options] Taxonomy+
```

où chaque argument `Taxonomie` est la taxonomie que vous voulez désinstaller ou un fichier `.altova_taxonomies`. Une taxonomie est spécifiée par un identifiant qui a un format `<name>-<version>`. (Les identifiants de taxonomies sont affichés quand vous exécutez la commande `list`¹⁷⁹.) Vous pouvez saisir autant de taxonomies que vous le souhaitez. Pour plus de détails, voir la description de la commande `désinstaller`¹⁸¹.

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

Réinitialiser Gestionnaire de taxonomie

Vous pouvez réinitialiser Gestionnaire de taxonomie.

- Dans la GUI, cliquez sur **Reset Selection**. Ceci réinitialise la GUI pour montrer quelles taxonomies sont actuellement installées. Toute sélection ou désélection que l'utilisateur a effectué dans la session actuelle sera annulée.
- Dans la CLI, exécutez la commande `reset`¹⁸⁰. Ceci supprime toutes les taxonomies installées et le répertoire de mise sous cache.

Une fois avoir exécuté cette commande, vous devrez exécuter la commande `initialize`¹⁷⁸, pour pouvoir recréer le répertoire de mise sous cache. En alternative, exécutez la commande `reset`¹⁸⁰ avec l'option `-i`.

Notez que `reset -i`¹⁸⁰ restaure l'installation originale du produit, il est recommandé d'exécuter la commande `update`¹⁸² après avoir réalisé la réinitialisation. En alternative, exécutez la commande `reset`¹⁸⁰ avec les options `-i` and `-u`.

7.9.5 Interface de ligne de commande (CLI)

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

<i>Windows</i>	C:\ProgramData\Altova\SharedBetweenVersions\TaxonomyManager.exe
----------------	---

Linux	/opt/Altova/MapForceServer2024/bin/taxonomymanager
macOS	/usr/local/Altova/MapForceServer2024/bin/taxonomymanager

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

Syntaxe de ligne de commande

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

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

Dans l'extrait ci-dessus, la barre verticale `|` sépare un ensemble d'items mutuellement exclusifs. Les crochets `[]` indiquent des items optionnels. De manière générale, vous pouvez saisir le chemin d'exécutable suivi soit par les options `--h`, `--help`, ou `--version` ou par une commande. Chaque commande peut contenir des options et des arguments. La liste des commandes est décrite dans les sections suivantes.

7.9.5.1 help

Cette commande propose une aide contextuelle pour les commandes liées à l'exécutable Gestionnaire de taxonomie.

Syntaxe

```
<exec> help [command]
```

Où `[command]` est un argument optionnel qui spécifie un nom de commande valide.

Veillez noter les points suivants :

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

Exemple

La commande suivante affiche une aide concernant la commande `list` :

```
./taxonomymanager help list
```

7.9.5.2 info

Cette commande affiche des informations détaillées pour chacune des taxonomies fournies en tant qu'argument de `Taxonomie`. Cette information inclut le titre, la version, description, l'éditeur et toute taxonomie soumise et toute taxonomie dépendante, et mentionne si la taxonomie a été installée ou non.

Syntaxe

```
<exec> info [options] Taxonomy+
```

- L'argument `Taxonomie` est le nom de la taxonomie ou une partie du nom de la taxonomie. (Pour afficher une ID de pack de taxonomie et des informations détaillées sur son statut d'installation, vous devriez utiliser la commande [list](#) ¹⁷⁹.)
- Utiliser `<exec> info -h` pour afficher l'aide de la commande.

Exemple

La commande suivante affiche des informations détaillées concernant les taxonomies `eba-2.1.0` et `us-gaap-2020.0` :

```
taxonomymanager info eba-2.1.0 us-gaap-2020.0
```

7.9.5.3 initialize

Cette commande initialise l'environnement Gestionnaire de taxonomie. Elle crée un répertoire de cache où les informations concernant toutes les taxonomies sont stockées localement. L'initialisation est réalisée automatiquement la première fois qu'une application activée par XBRL est installée. Vous n'aurez pas besoin d'exécuter cette commande dans des circonstances normales, mais vous devrez l'exécuter généralement après la commande `reset`.

Syntaxe

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

Options

La commande `initialize` accepte les options suivantes :

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

Exemple

La commande suivante initialise Gestionnaire de taxonomie:

```
taxonomymanager initialize
```

7.9.5.4 install

Cette commande installe une ou plusieurs taxonomies.

Syntaxe

```
<exec> install [options] Taxonomy+
```

Pour installer de multiples taxonomies, ajoutez l'argument **Taxonomy** de nombreuses fois.

L'argument **Taxonomy** est l'un des suivants :

- Un identifiant de taxonomie (avoir un format de `<name>-<version>`, par exemple : `eba-2.10`). Pour trouver les identifiants de taxonomie que vous voulez, exécutez la commande [list](#)¹⁷⁹. Vous pouvez aussi utiliser des identifiants abrégés s'ils sont uniques, par exemple `eba`. Si vous utilisez un identifiant abrégé, alors la dernière version de cette taxonomie sera installée.
- Le chemin vers un fichier `.altova_taxonomies` téléchargé depuis le site web d'Altova. Pour information sur ces fichiers, voir [Introduction à TaxonomyManager : Comment cela fonctionne-t-il ?](#)¹⁶⁷.

Options

La commande `install` accepte les options suivantes :

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

Exemple

La commande suivante installe les dernières taxonomies `eba` (European Banking Authority) et `us-gaap` (US Generally Accepted Accounting Principles) :

```
taxonomymanager install eba us-gaap
```

7.9.5.5 list

Cette commande recense les taxonomies sous la gestion de Gestionnaire de taxonomie. La liste affiche comme suit

- Toutes les taxonomies disponibles
- Les taxonomies contenant le string dans leur nom soumis comme argument **Taxonomy**
- Seules les taxonomies installées
- Seules les taxonomies qui peuvent être mises à niveau

Syntaxe

```
<exec> list | ls [options] Taxonomy?
```

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

Options

La commande **list** accepte les options suivantes :

<code>--installed, --i</code>	Uniquement la liste des taxonomies installées. Le réglage par défaut est faux .
<code>--upgradeable, --u</code>	Recenser uniquement les taxonomies lorsque des mises à niveau (patches) sont disponibles. Le réglage par défaut est faux .
<code>--help, --h</code>	Afficher l'aide pour la commande.

Exemples

- Pour exécuter toutes les taxonomies disponibles, exécutez : `taxonomymanager list`
- Pour recenser les taxonomies installées, exécutez : `taxonomymanager list -i`
- Pour recenser tous les taxonomies qui contiennent soit "eba", soit "us-gaap" dans leur nom, exécutez `taxonomymanager list eba us-gaap` :

7.9.5.6 reset

Cette commande supprime toutes les taxonomies installées et le répertoire de mise sous cache. Vous réinitialiserez complètement votre environnement de taxonomie. Une fois avoir exécuté cette commande, vous devrez exécuter la commande [initialize](#)¹⁷⁸, pour pouvoir recréer le répertoire de mise sous cache. En alternative, exécuter la commande `reset` avec l'option `-i`. Puisque `reset -i` restaure l'installation originale du produit, nous vous recommandons que vous exécutiez la commande [update](#)¹⁸² après avoir réalisé la réinitialisation et l'initialisation. En alternative, exécutez la commande `reset` avec les options `-i`¹⁷⁸ et `-u`¹⁷⁸.

Syntaxe

```
<exec> reset [options]
```

Options

La commande **reset** accepte les options suivantes :

<code>--init, --i</code>	Initialiser Gestionnaire de taxonomie après le reset. Le réglage par défaut est faux .
<code>--update, --u</code>	Mettre à jour la liste de taxonomies disponibles dans le cache. Le réglage par défaut est faux .

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

Exemples

- Pour réinitialiser Gestionnaire de taxonomie, exécuter : `taxonomymanager reset`
- Pour réinitialiser Gestionnaire de taxonomie et l'initialiser, exécutez : `taxonomymanager reset -i`
- Pour réinitialiser Gestionnaire de taxonomie, initialiser-la et mettez à jour sa liste de taxonomie, exécutez : `taxonomymanager reset -i-u`

7.9.5.7 uninstall

Cette commande désinstalle une ou plusieurs taxonomies. Par défaut, toute taxonomie référencée par la taxonomie actuelle sera également désinstallée. Pour désinstaller uniquement la taxonomie actuelle et garder les taxonomies référencées, définir l'option `--k`.

Syntaxe

`<exec> désinstaller [options] Taxonomy+`

Pour désinstaller de multiples taxonomies, ajoutez l'argument `Taxonomy` de nombreuses fois.

L'argument `Taxonomy` est l'un des suivants :

- Un identifiant de taxonomie (avoir un format de `<name>-<version>`, par exemple : `eba-2.10`). Pour trouver les identifiants de taxonomie qui sont installées, exécutez la commande `list -i`¹⁷⁹. Vous pouvez aussi utiliser un nom de taxonomie abrégé s'il est unique, par exemple `eba`. Si vous utilisez un nom abrégé, alors toutes les taxonomies qui contiennent une abréviation dans leur nom seront désinstallées.
- Le chemin vers un fichier `.altova_taxonomies` téléchargé depuis le site web d'Altova. Pour information sur ces fichiers, voir [Introduction à TaxonomyManager : Comment cela fonctionne-t-il ?](#)¹⁶⁷.

Options

La commande `désinstaller` accepte les options suivantes :

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

Exemple

La commande suivante désinstalle les taxonomies `eba-2.10` et `us-gaap-2020.0` et leurs dépendances :

```
taxonomymanager uninstall eba-2.10 us-gaap-2020.0
```

La commande suivante désinstalle la taxonomie `eba-2.10` mais pas les taxonomies qu'elle référence :

```
taxonomymanager uninstall --k eba-2.10
```

7.9.5.8 update

Cette commande requête la listes des taxonomies disponibles depuis le magasin en ligne et met à jour le répertoire de mise sous cache local. Vous devriez exécuter cette commande sauf si vous avez réalisé un [reset](#)¹⁸⁰ et [initialize](#)¹⁷⁸.

Syntaxe

```
<exec> update [options]
```

Options

La commande **mise à jour** accepte les options suivantes :

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

Exemple

La commande suivante met à jour le cache local avec la liste des dernières taxonomies :

```
taxonomymanager update
```

7.9.5.9 upgrade

Cette commande met à niveau toutes les taxonomies éligibles qui peuvent être mises à niveau à la dernière version *patch* disponible. Vous pouvez identifier des taxonomies à mettre à niveau en exécutant la commande [list -u](#)¹⁷⁹.

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

Syntaxe

```
<exec> upgrade [options]
```

Options

La commande `mise à niveau` accepte les options suivantes :

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

8 Fichiers catalogues

Si vous utilisez le mécanisme de catalogue d'Altova, vous pouvez trouver les fichiers de catalogue pertinents dans le dossier `etc` du dossier d'application MapForce Server. Pour des informations détaillées, voir la [section Catalogues du manuel de MapForce d'Altova](#).

Vous pouvez créer `CustomCatalog.xml` depuis le fichier modèle `CustomCatalog_template.xml`. Assurez-vous de renommer le modèle de fichier en `CustomCatalog.xml` étant donné que ce dernier fichier sera le fichier utilisé dans le mécanisme de catalogue (pas le modèle de fichier).

Veillez noter les points suivants :

- Lors d'une nouvelle installation de la même version majeure (mêmes versions ou différentes versions mineures), le modèle de fichier sera remplacé par un nouveau modèle de fichier, mais `CustomCatalog.xml` ne sera pas touché.
- Toutefois, si vous installez une nouvelle version majeure au-dessus de la version majeure précédente, alors le dossier de la version majeure précédente sera supprimé - ensemble avec son `CustomCatalog.xml`. Donc, si vous voulez continuer à utiliser `CustomCatalog.xml`, assurez-vous d'enregistrer `CustomCatalog.xml` depuis le dossier de la version majeure à un endroit sûr. Une fois que la nouvelle version majeure a été installée, vous pouvez copier `CustomCatalog.xml` que vous avez enregistré dans le dossier `etc` de la nouvelle version majeure et l'éditer là tel que requis.

Index

A

ADO, 6, 43

ADO.NET, 6, 43

Altova ServiceController, 13

assignlicense,

as CLI command, 75

Attribuer une licence à MapForce Server sur Linux, 21

Attribuer une licence à MapForce Server sur macOS, 26

Attribuer une licence à MapForce Server sur Windows, 15

C

C++,

exécuter les mappages avec, 96

exemple, 96

Certificats numériques,

dans les mappages MapForce, 148

de confiance sur Linux, 151

de confiance sur Mac, 152

de confiance sur Windows, 153

exporter depuis Windows, 155

gestion sur Windows, 154

transférer sur Linux, 161

transferring to Mac, 163

CLI commands,

assignlicense, 75

exportresourcestrings, 76

help, 78

licenseserver, 79

setdeflang, 84

verifylicense, 85, 86

Cœurs de serveur,

exigences, 31

commandes CLI,

run, 80

Configuration,

sur Linux, 17

sur macOS, 23

sur Windows, 9

Configuration de MapForce Server, 8

D

Démarrer LicenseServer sur Linux, 20

Démarrer LicenseServer sur macOS, 25

Démarrer LicenseServer sur Windows, 13

Démarrer MapForce Server sur Linux, 20

Démarrer MapForce Server sur macOS, 25

Démarrer MapForce Server sur Windows, 13

Désinstallation, 9

Désinstaller, 9

E

Enregistrer MapForce Server avec LicenseServer sur Linux, 20

Enregistrer MapForce Server avec LicenseServer sur macOS, 26

Enregistrer MapForce Server avec LicenseServer sur Windows, 14

exécuter,

en tant que commande CLI, 80

Exemple C#,

exécuter les mappages avec, 89

exemple, 89

exportresourcestrings,

as CLI command, 76

F

Fenêtres,

certificats de serveur de confiance sur, 153

FlowForce Server, 6, 43

G

Gérer la licence de MapForce Server, 8

Gestionnaire de taxonomie,

aperçu CLI, 176

aperçu de, 167

Commande Aide CLI, 177

Gestionnaire de taxonomie,

- Commande de mise à jour CLI, 182
- Commande de mise à niveau CLI, 182
- Commande désinstallation CLI, 181
- Commande Info CLI, 178
- Commande Initialiser CLI, 178
- Commande Installer CLI, 179
- Commande Liste CLI, 179
- Commande Réinitialiser CLI, 180
- comment exécuter, 171
- corriger une taxonomie, 174
- désinstaller une taxonomie, 175
- installation d'une taxonomie, 174
- mettre à niveau une taxonomie, 174
- recenser les taxonomies par statut dans, 173
- réinitialiser, 175
- statut de taxonomies dans, 173

H**help,**

- as CLI command, 78

HTTPS,

- appeler des services Web via, 148

I**installation de LicenseServer sur Linux, 19****Installation de MapForce Server, 8****Installation sur Linux, 17****Installation sur Windows, 9****Installer LicenseServer sur macOS, 24****Installer LicenseServer sur Windows, 12****Installer sur macOS, 23****Installer sur Windows Server Core, 10****Interface de ligne de commande,**

- comment utiliser, 72

J**Java, 43**

- ajouter MapForce Server à CLASSPATH, 105
- Appeler MapForce Server de, 109

- exemple, 109

L**Licence MapForce Server sur Linux, 20****Licence MapForce Server sur macOS, 25****Licence MapForce Server sur Windows, 13****Licence pour MapForce Server,**

- attribuer à Linux, 21
- Attribuer à Windows, 15
- Attribuer sur macOS, 26

licenseserver,

- as CLI command, 79

Linux,

- certificats de serveur de confiance sur, 151
- consulter les produits Altova installés actuellement, 34
- désinstallerMapForce Server de, 34
- exécuter des mappages avec des appels de service Web via HTTPS, 161
- info de prise en charge, 6
- installation sur, 17
- installing MapForce Server on, 33
- Licence MapForce Server sur, 20
- support info, 30
- transférer des certificats de client sur, 161

M**Mac,**

- certificats de serveur de confiance sur, 152
- exécuter des mappages avec des appels de service Web via HTTPS, 163
- transférer des certificats de client sur, 163

macOS,

- désinstallerMapForce Server de, 36
- info de prise en charge, 6
- installation sur, 23
- installing MapForce Server on, 35
- Licence MapForce Server sur, 25
- support info, 30

MapForce Server,

- migrer vers un nouvel appareil, 29

MapForce Server API,

- exemple C#, 89
- exemple C++, 96

MapForce Server API,

- Exemple Java, 109
- exemple VB.NET, 92
- exemple VBA, 102
- exemple VBScript, 99
- introduction, 87
- Pour .NET, 88
- pour COM, 96
- pour Java, 105

Mappages,

- exécuter avec C#, 89
- exécuter avec C++, 96
- exécuter avec Java, 109
- exécuter avec VB.NET, 92
- exécuter avec VBA, 102
- exécuter dans la ligne de commande, 80
- préparer pour l'exécution, 43

Mappings,

- exécuter avec VBScript, 99

Migrer MapForce Server vers un nouvel appareil, 29**Mise à niveau de MapForce Server sur Windows, 28****O****ODBC, 6, 43****Optimisation Join,**

- fonctionnement, 51

R**Ressources Globales,**

- définition, 49
- utiliser dans des mappages, 49

S**Services Web,**

- appeler via HTTPS, 148

setdeflang,

- as CLI command, 84

V**VB.NET,**

- exécuter des mappages avec, 92
- exemple, 92

VBA,

- exécuter des mappages avec, 102
- exemple, 102

VBScript,

- exécuter des mappages avec, 99
- exemple, 99

verifylicense,

- as CLI command, 85

version,

- as CLI command, 86

Versions de LicenseServer, 12, 19, 24**W****Windows,**

- désinstaller MapForce Server de, 39
- exécuter les mappages avec des appels de service Web via HTTPS, 165
- info de prise en charge, 6
- installation sur, 9
- installing MapForce Server on, 37
- Licence MapForce Server sur, 13
- mise à niveau de MapForce Server sur, 28
- support info, 30

X**XML Catalogs,**

- configuring, 184
- how it works, 184