

Altova MapForce Server 2021

ユーザーマニュアル

Altova MapForce Server 2021

ユーザーマニュアル

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.

公開日: 2015–2021

(C) 2015–2021 Altova GmbH

目次

1	はじめに	6
2	インストール	8
2.1	システム必要条件.....	8
2.2	プロセッサコアとライセンス.....	9
2.3	重要なパス.....	10
2.4	Linux.....	11
2.4.1	MapForce Server のインストール.....	11
2.4.2	現在インストールされている製品の確認.....	12
2.4.3	MapForce Server をアンインストールする方法.....	12
2.5	macOS.....	13
2.5.1	MapForce Server のインストール.....	13
2.5.2	MapForce Server をアンインストールする方法.....	14
2.6	Windows.....	15
2.6.1	MapForce Server のインストール.....	15
2.6.2	MapForce Server をアンインストールする方法.....	16
3	マッピングの実行	17
3.1	サーバーでの実行のためのマッピングの準備.....	19
3.2	グローバルリソース.....	24
3.3	動的な認証.....	25
4	MapForce Server コマンドライン	26
4.1	assignlicense (Windows only).....	27
4.2	exportresourcestrings.....	28
4.3	help.....	29
4.4	licenseserver.....	30

4.5	run.....	31
4.6	setdeflang.....	34
4.7	verifylicense (Windows only).....	35
4.8	version.....	36

5 MapForce Server API 37

5.1	.NET インターフェイス.....	38
5.1.1	C# サンプル.....	39
5.1.2	Visual Basic .NET サンプル.....	41
5.2	COM インターフェイス.....	45
5.2.1	C++ サンプル.....	45
5.2.2	VBScript サンプル.....	47
5.2.3	VBA サンプル.....	50
5.3	Java インターフェイス.....	53
5.3.1	Java サンプル.....	55
5.4	例: パラメーターを持つマッピングの実行.....	58
5.5	API レファレンス (COM、.NET).....	63
5.5.1	インターフェイス.....	63
5.6	API レファレンス (Java).....	80
5.6.1	クラス.....	80

6 デジタル証明書の管理 92

6.1	Windows 上で証明書ストアにアクセスする.....	95
6.2	Windows 上で証明書をエクスポートする.....	98
6.3	.Linux 上のクライアント証明書.....	104
6.4	.macOS 上のクライアント証明書.....	105
6.5	.Linux 上のサーバーの証明書を信頼する.....	106
6.6	.macOS 上のサーバーの証明書を信頼する.....	108
6.7	.Windows 上でサーバーの証明書を信頼する.....	109
6.8	.Windows 上のクライアント証明書.....	110

7 XBRL タクソノミ マネジャー 111

7.1	XBRL タクソノミ マネジャー の実行.....	115
-----	---------------------------	-----

7.2	タクソミのインストール.....	116
7.3	インストール済みのタクソミをビューする.....	119
7.4	タクソミのアンインストール.....	120
7.5	コマンドラインインターフェイス.....	121
7.5.1	help.....	122
7.5.2	info.....	123
7.5.3	intialize.....	123
7.5.4	install.....	124
7.5.5	list.....	125
7.5.6	reset.....	126
7.5.7	setdeflang.....	126
7.5.8	uninstall.....	127
7.5.9	update.....	128
7.5.10	upgrade.....	128

8	カタログファイル	130
----------	-----------------	------------

	インデックス	133
--	---------------	------------

1 はじめに

MapForce Server は、Windows、Linux、および macOS オペレーティングシステム上で作動するデータマッピング、変換のための企業ソフトウェアソリューションです。データマッピング (または、マッピング デザイン ファイル、*.mfd) は、入力、出力、および データに適用される中間の処理ステップが定義される Altova MapForce (<http://www.altova.com/ja/mapforce.html>) 内で視覚的にデザインされます。MapForce Server の役割は、MapForce Server Execution (.mfx) ファイルを MapForce と共に実行し、出力 ファイルまたはデータを生成することです。また、基礎となっているマッピングのデザインに従いデータベースを更新し、Web サービスを呼び出すこともできます。

MapForce Server は、スタンドアロンで作動し、Altova FlowForce Server の管理下で作動することもできます (<http://www.altova.com/ja/flowforce.html>)。MapForce Server と同じマシンにインストールされていると、FlowForce Server は、マッピングの実行をスケジュールに基づき実行、またはトリガーをベースにしたジョブを使用して自動的に実行することができます。更に、FlowForce Server には、マッピングの実行の前後に、ファイルとディレクトリのコピー、FTP へのファイルのアップロード、シェルコマンドの実行などの追加アクションを自動的に実行することのできる関数ライブラリが内蔵されています。

機能

- データマッピングの実行する際のサーバーレベルのパフォーマンス。
- クロスプラットフォーム MapForce Server は、Windows、Linux、または macOS オペレーティングシステム上で作動します。
- コマンドライン インターフェイス
- C++、C#、Java、VB.NET、VBScript または VBA コードから呼び出すことのできる API
- FlowForce Server とのネイティブな統合
- Altova グローバルリソースのサポート。ファイル、フォルダー、データベースへの参照の構成と、複数の Altova アプリケーションおよび複数の環境への適用に関しては次を参照してください！ [Altova グローバルリソース](#)。
- プロトコル、フォーマット、バイナリフォーマットからデータを読み込む、または、データを書き込むマッピングを実行します。
- リレーデータベース挿入を行うマッピングを実行します。

制限

- XML デジタル署名はサポートされていません
- ADO、ADO.NET、および ODBC データベース接続は、Windows 上でのみサポートされます。Linux と macOS 上では、SQLite と PostgreSQL のためのネイティブなデータベース接続を使用することができます。Linux または macOS 上で作動する他のデータベースに関しては、JDBC を使用してください。

システムの必要条件

▼ Windows

プラットフォーム更新済みの Windows 7 SP1、Windows 8、Windows 10

▼ Windows Server

プラットフォーム更新済みの Windows Server 2008 R2 SP1 または以降

Windows 版では MapForce Server を 32 ビット と 64 ビットの バージョンでご利用いただけます。

▼ Linux

- CentOS 7 または以降

- RedHat 7 まじお以降
- Debian 8 まじお以降
- Ubuntu 16.04 LTS まじお以降

次のライブラリはアプリケーションをインストール実行するために必要とされるライブラリです。下のパッケージが使用中 Linux のマシンで使用できない場合、yum (まじおは 適用できる場合、apt-get を) コマンドを実行してインストールしてください。

サーバー	CentOS, RedHat	Debian	Ubuntu
LicenseServer	krb5-libs	libgssapi-krb5-2	libgssapi-krb5-2
MapForce Server	libidn, krb5-libs	libidn11, libgssapi-krb5-2	libidn11, libgssapi-krb5-2

メモ Windows と macOS では、同じマシンに MapForce Server の複数のバージョンをインストールすることが可能です。このような場合、1度に同じ Altova LicenseServer に登録できるバージョンは一つです。このため、MapForce Server の2つの異なるバージョンは、同じマシンで同時に作動することはできません。同じ製品のバージョンの複数のインスタンスは並列に作動することができます。詳細に関しては、登録された製品にライセンスを割り当てるを参照してください。

最終更新日: 2021年02月24日

2 インストール

2.1 システム必要条件

▼ Windows

プラットフォーム更新済みのWindows 7 SP1、Windows 8、Windows 10

▼ Windows Server

プラットフォーム更新済みのWindows Server 2008 R2 SP1 以降

Windows 上ではMapForce Server を32ビットと64ビットパッケージで 사용할 수 있습니다.

▼ Linux

- CentOS 7 以降
- RedHat 7 以降
- Debian 8 以降
- Ubuntu 16.04 LTS 以降

次のライブラリはアプリケーションをインストール実行するために必要とされるライブラリです。下のパッケージが使用中 Linux のマシンで使用できない場合、yum (または 適用できる場合、apt-get) をコマンドを実行してインストールしてください。

サーバー	CentOS, RedHat	Debian	Ubuntu
LicenseServer	krb5-libs	libgssapi-krb5-2	libgssapi-krb5-2
MapForce Server	libidn, krb5-libs	libidn11, libgssapi-krb5-2	libidn11, libgssapi-krb5-2

▼ macOS

macOS 10.13 以降

2.2 プロセッサコアとライセンス

Altova サーバー製品へのライセンス供与は(ロジカルなコアの数量ではなく)製品マシン上で使用することのできる物理的なプロセッサのコアの数量をベースとしています。例えば、デュアルコアプロセッサには、クアッドコアプロセッサ4コア、ヘキサコア6コアなどの2つのコアが存在します。製品にライセンス供与されたコア数は、物理マシンは仮想マシンに関わらず、サーバーマシン上で使用することのできるコアの数量と同等、またはそれ以上である必要があります。例えば、サーバーが8コア持つ場合、8コアライセンスを購入する必要があります。コアの数量を満たすためにライセンスを組み合わせることもできます。ですから、2つの4コアライセンスを使用して、8コアライセンスの代わりに8コアを達成することもできます。

処理する容量は少ないが、CPU コアの大きなコンピューターサーバーを使用している場合、少ない数量のコアに割り当てられている仮想マシンを作成し、その数量のライセンスを購入することもできます。このようなデプロイの場合、もちろん、コンピューターのすべてのコアを使用する場合に比べこのようなデプロイの処理スピードは遅いです。

メモ 各 Altova サーバー製品ライセンスは、使用されていないライセンス供与能力が存在する場合でも、Altova サーバー製品がインストールされているマシンで、一度に1つのクライアントマシンのためにのみ使用することができます。例えば、10 コアライセンスが6 CPU コアを持つマシンのために使用されている場合、残りのライセンス許与能力の4 コアは、他のクライアントマシンのために同時に使用することはできません。

単一スレッド実行

Altova サーバー製品が単一スレッドの実行を許可すると、*単一スレッドの実行*を使用できるようになります。この場合、Altova サーバー製品ライセンスはライセンスプール内で1コアのみ使用することができます。複数のコアを持つマシンはこの1コアライセンスに割り当てることができます。このような場合、その製品を動作するマシンは単一コアで動作します。(マルチコアでは使用可能な)複数スレッドを使用できないため処理はこのために遅くなります。製品はそのマシン上で単一スレッドモードで実行されます。

マルチコアマシンに単一コアライセンスを割り当てることは、その製品のためにLicenseServer 内で「*単一スレッド実行に制限する*」チェックボックスを選択してください。

必要なコア数と見積もり

サーバーが扱うことのできるデータの量と処理回数を影響する多くの外部要素が存在します。(例えば、ハードウェア、CPU 上の現在のロード、サーバー上で動作する他のアプリケーションのメモリの割り当てなどが挙げられます)。パフォーマンスを正確に測定するために、データの量、条件、およびビジネスの環境に近い状態でアプリケーションをテストしてください。

2.3 重要なパス

インストールディレクトリ

アプリケーションは次のディレクトリにインストールされます:

Linux /opt/Altova/MapForceServer2021/bin
Mac /usr/local/Altova/MapForceServer2021/bin
Windows <ProgramFilesFolder>\Altova\MapForceServer2021\bin

実行可能ファイル

インストール後、MapForce Server 実行可能ファイルは以下のパスで見つけることができます:

Linux /opt/Altova/MapForceServer2021/bin/**mapforceserver**
Mac /usr/local/Altova/MapForceServer2021/bin/**mapforceserver**
Windows <ProgramFilesFolder>\Altova\MapForceServer2021\bin**MapForceServer.exe**

アプリケーション構成データ

アプリケーションの構成データは次のディレクトリに保管されています:

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

2.4 Linux

2.4.1 MapForce Server のインストール

システムの必要条件

▼ Linux

- CentOS 7 または以降
- RedHat 7 または以降
- Debian 8 または以降
- Ubuntu 16.04 LTS または以降

次のライブラリはアプリケーションをインストール実行するために必要とされるライブラリです。下のパッケージが使用中 Linux のマシンで使用できない場合、yum (または 適用できる場合、apt-get) を コマンドを実行してインストールしてください。

サーバー	CentOS, RedHat	Debian	Ubuntu
LicenseServer	krb5-libs	libgssapi-krb5-2	libgssapi-krb5-2
MapForce Server	libidn, krb5-libs	libidn11, libgssapi-krb5-2	libidn11, libgssapi-krb5-2

前提条件

- root ユーザーまたは sudo 特権を持つユーザーとしてインストールを行ってください。
- MapForce Server の前のバージョンが新しいバージョンがインストールされる前にアンインストールされる必要があります。
- FlowForce Server を他の Altova サーバー製品とインストールする場合は、FlowForce Server を最初にインストールすることが奨励されます。

MapForce Server のインストール

1. <https://www.altova.com/download#server> からローカルディレクトリにインストールパッケージをダウンロードします。
2. インストールパッケージをダウンロードするディレクトリに変更します。例:

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

3. MapForce Server パッケージをインストールします。

```
[Debian, Ubuntu] sudo dpkg --install mapforceserver-2021-debian.deb
[CentOS, RedHat] sudo rpm -ivh mapforceserver-2021-1.x86_64.rpm
```

メモ 現在のリリース または サービスパックバージョンに一致するために、上記のパッケージの名前を調整する必要がある場合があります。

MapForce Server にライセンスを供与する方法

1. Altova LicenseServer の最新バージョンがインストールされており、ローカルマシン、または ネットワーク上で実行されていること

- を確認してください(<https://www.altova.com/ja/licenseserver>)。
2. MapForce Server を Altova LicenseServer に登録します。このためには、作動中のコンピュータのホスト名、またはアドレスを知る必要があります。例えば、LicenseServer が 127.0.0.1 上で作動する場合、登録は以下のように行うことができます:

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

メモ 上記のコマンドは、プログラムインストールディレクトリから実行可能ファイルを呼び出していることが前提です。[重要なリンク](#)も参照してください

3. LicenseServer 管理者インターフェイスログイン、MapForce Server が実行されるマシンにライセンスを割り当てます。

ライセンスの詳細に関しては、Altova LicenseServer ドキュメントを参照してください(<https://www.altova.com/ja/documentation>)。

2.4.2 現在インストールされている製品の確認

Altova サーバー製品がインストールされているか確認するには以下のコマンドを実行します:

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

2.4.3 MapForce Server をアンインストールする方法

MapForce Server をアンインストールするには、以下のコマンドを実行します:

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

Debian と Ubuntu パッケージは “--remove” の代わりに “--purge” を使用しておこなわれるべきです。

2.5 macOS

2.5.1 MapForce Server のインストール

システムの必要条件

▼ macOS

macOS 10.13 また以降

必要条件

- 管理 (ルート) 特権を持つユーザーとしてインストールします。

FlowForce Server 統合

MapForce Server をスタンドアロンとして、または FlowForce Server インストールの一部としてインストールすることができます。To install MapForce Server を FlowForce Server の管理下でインストールするには、FlowForce インストールウィザード上で関連するオプションを選択します。FlowForce Server Advanced Edition インストーラーには MapForce Server Advanced Edition インストーラーが含まれています。FlowForce Server と MapForce Server のインストーラーは Altova ダウンロードセンターで見つけることができます (<https://www.altova.com/ja/download>)。

MapForce Server を FlowForce Server と共にインストールする場合、FlowForce Server を最初にインストールすることが奨励されます。それ以外の場合は、下記のインストール後のタスクを参照してください。

MapForce Server のインストール

- Altova Web ページから <http://www.altova.com/ja/download.html> から、ディスクイメージファイル(.dmg) をダウンロードし、開くためにクリックします。これによりインストーラーがコンピューター上で新規ドライブとして表示されます。
- パッケージ(.pkg) ファイルをダブルクリックして、インストールウィザードに従いインストールします。これらのステップは、説明を必要とせず、続行する前に、ライセンス使用許諾書に同意するステップ1が含まれています。ウィザードを完了すると、FlowForce Server アイコンがアプリケーション内に表示されます。パッケージは以下のディレクトリに `/usr/local/Altova/FlowForceServer2021/`。
- Altova License Server がインストールされていない場合、または、最新バージョンにアップグレードする場合、対応するパッケージ(.pkg) ファイルをダブルクリックし、画面の説明に従ってください。

MapForce Server へのライセンス供与

- Altova LicenseServer の最新バージョンがインストールされており、ローカルマシン、またはネットワーク上で実行されていることを確認してください (<https://www.altova.com/ja/licenseserver>)。
- MapForce Server を Altova LicenseServer に登録します。このためには、が作動中のコンピューターのホスト名、または IP アドレスを知る必要があります。例えば、LicenseServer が 127.0.0.1 上で作動する場合、登録は以下のように行うことができます:

```
sudo mapforceserver licenseserver 127.0.0.1
```

- LicenseServer 管理者インターフェイスログイン、MapForce Server が実行されるマシンにライセンスを割り当てます。

詳細に関しては、Altova LicenseServer ドキュメントを参照してください(<https://www.altova.com/ja/documentation>)。

インストール後のタスク

MapForce Server をFlowForce Server をインストールする前にインストールすると、MapForce Server とFlowForce Server をインストールした後、次のコマンドを実行してください。

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

上記のコマンドは .tool ファイルを MapForce Server の/etc ディレクトリから FlowForce Server の/tools ディレクトリにコピーします。.tool ファイルは FlowForce Server により必要とされており、MapForce Server 実行可能ファイルへのパスを含んでいます。MapForce Server のインストール前に FlowForce Server をインストールする場合、このコマンドを実行する必要はありません。

2.5.2 MapForce Server をアンインストールする方法

MapForce Server をインストールする方法:

1. アプリケーション内で MapForce Server アイコンを右クリックして、ゴミ箱に移動を選択します。MapForce Server インストールディレクトリと構成データはこのステップでは削除されません。
2. 任意で MapForce Server インストールディレクトリを削除すると、次のコマンドを実行してください。

```
sudo rm -rf /usr/local/Altova/MapForceServer2021/
```

メモ 上記のステップは /var/Altova/MapForceServer2021. から構成データを移動しません。構成データが移動されると、後に復元することはできません。

2.6 Windows

2.6.1 MapForce Server のインストール

システム必要条件

- プラットフォーム更新済みのWindows 7 SP1、Windows 8、Windows 10
- プラットフォーム更新済みのWindows Server 2008 R2 SP1 または以降

必要条件

- 管理特権を持つユーザーとしてインストールを行います。
- バージョン 20217.1 以降から MapForce Server の 32 ビットバージョンは 64 ビットバージョンにインストールすることはできません。または 64 ビットバージョンを 32 ビットバージョンにインストールすることはできません。(i) 新しいバージョンをインストールする前に古いバージョンを削除する。または(ii) 古いインストールと同じビットバージョンを新規のバージョンに更新する必要があります。

MapForce Server のインストール

MapForce Server をスタンドアロンとして、または FlowForce Server インストールの一部としてインストールすることができます。To install MapForce Server を FlowForce Server の管理下でインストールするには FlowForce インストールウィザード上で関連するオプションを選択します。FlowForce Server Advanced Edition インストーラーには MapForce Server Advanced Edition インストーラーが含まれています。FlowForce Server と MapForce Server のインストーラーは Altova ダウンロードセンターで見つけることができます (<https://www.altova.com/ja/download>)。

MapForce Server スタンドアロンをインストールする方法

1. Altova ダウンロードセンター (<https://www.altova.com/ja/download>) からインストールファイルをダウンロードして実行します。
2. 任意で、ウィザードの左下のボックスからインストール言語を選択し、「次へ」をクリックします。コマンドラインから後で言語を変更することができます。
3. 以下の内の一つを行ってください。
 - a. Altova LicenseServer がまだインストールされていない場合、デフォルトの設定を保持してください。ウィザードが最新バージョンの Altova LicenseServer をウィザードを起動しているコンピューターにインストールします。
 - b. Altova LicenseServer がまだインストールされておらず、他のコンピューターに Altova LicenseServer をインストールする場合、「使用中のコンピューターに Altova LicenseServer をインストールする」チェックボックスをクリアして、「後で登録」を選択します。この場合、Altova LicenseServer のインストールと MapForce Server の登録を個別に行う必要があります。
 - c. Altova LicenseServer が既に使用中のコンピューターにインストールされているが、インストールウィザードにより示されているバージョンより低い場合は、デフォルトの設定を保持してください。この場合、インストールウィザードが自動的に LicenseServer バージョンをダイアログボックスに表示されたバージョンにアップグレードします。既存の登録およびライセンス情報はアップグレードの後にも保存されます。
 - d. Altova LicenseServer が既に使用中のコンピューター、またはネットワークにインストールされており、ウィザードに示されているバージョンと同じ場合は、以下を行ってください。
 - i. 「使用中のコンピューターに Altova LicenseServer をインストールする」チェックボックスをクリックしてクリアします。
 - ii. 「この製品を登録する」から MapForce Server を登録する Altova LicenseServer インスタンスを選択、または「後で登録」を選択してください。LicenseServer への関連を無視し、MapForce Server のインストールを継続する場合、「後で登録」を常に選択することができます。
4. 「次へ」をクリックします。
5. ウィザードに従いインストールを完了してください。

MapForce Server へのライセンス供与

1. Altova LicenseServer の最新バージョンがインストールされており、ローカルマシン、またはネットワーク上で実行されていることを確認してください (<https://www.altova.com/ja/licenseserver>)。
2. インストール中に行っていない場合、MapForce Server を Altova LicenseServer に登録します。このためには、が作動中のコンピュータのホスト名、または、アドレスを知る必要があります。例えば LicenseServer が 127.0.0.1 上で作動する場合、登録は以下のように行うことができます:

```
MapForceServer.exe licenseserver 127.0.0.1
```

3. LicenseServer 管理者インターフェイスにログイン、MapForce Server が実行されるマシンにライセンスを割り当てます。

詳細に関しては、Altova LicenseServer ドキュメントを参照してください (<https://www.altova.com/ja/documentation>)。

2.6.2 MapForce Server をアンインストールする方法

MapForce Server をアンインストールする方法

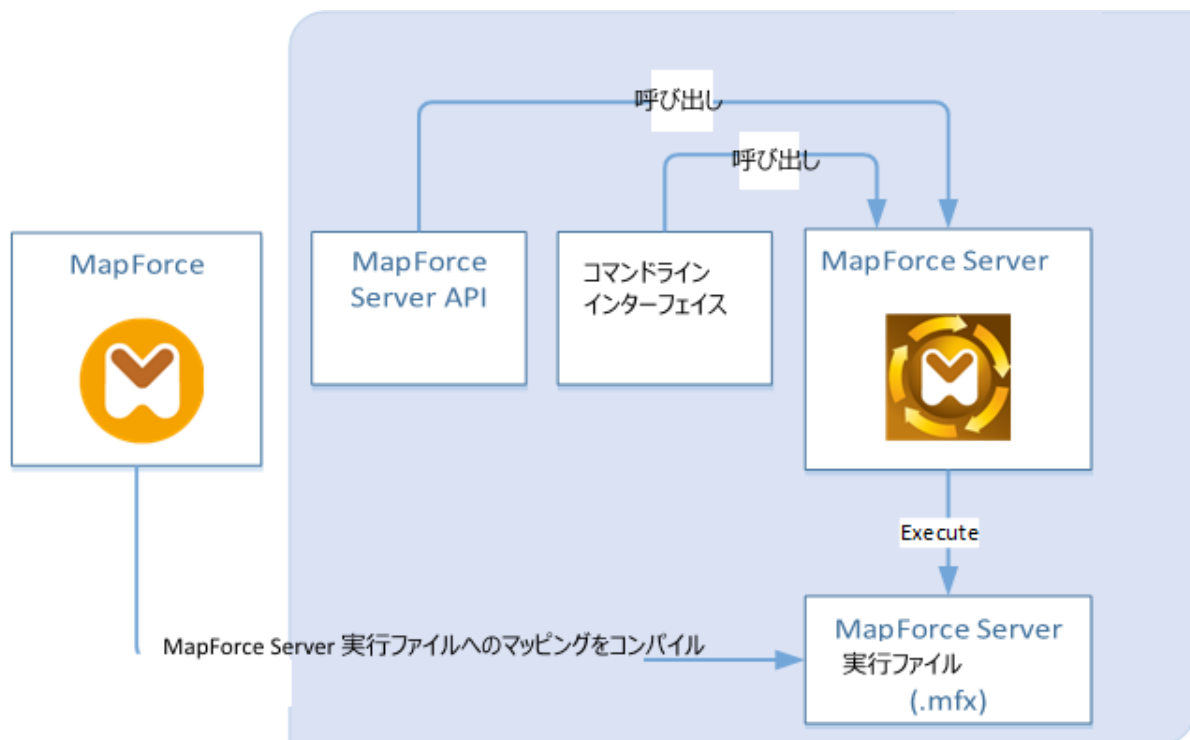
1. Windows 「開始」 ボタンを右クリックして、「設定」を選択します。
2. コントロール パネルを開きます (「コントロール パネル」を入力し、提案されたエントリをクリックします)。
3. 「プログラム」から「プログラムのアンインストール」をクリックします。
4. コントロール パネル内で、「Altova MapForce Server」を選択し、「アンインストール」をクリックします。

3 マッピングの実行

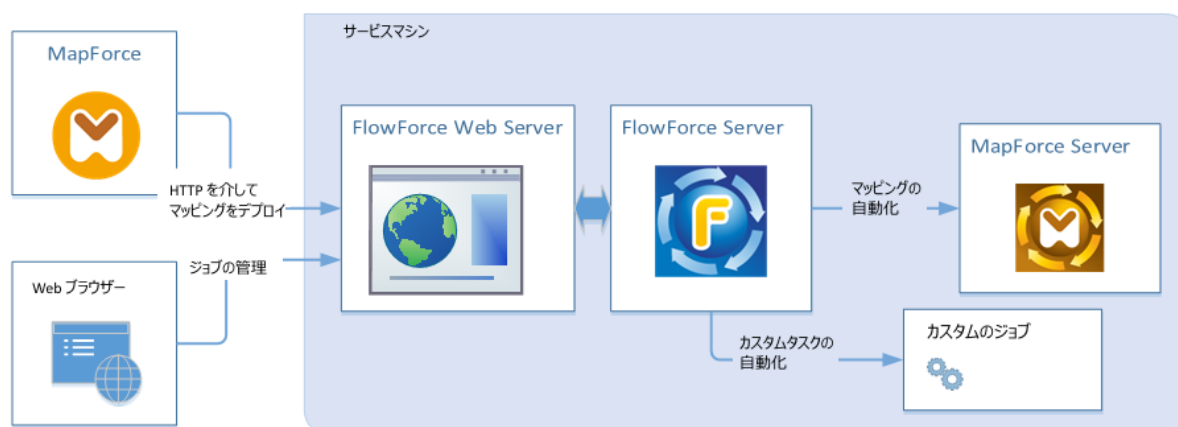
MapForce Server の役割は、Altova MapForce で作成されたデータマッピングを実行することです (<https://www.altova.com/ja/mapforce.html>)。

最初に、データマッピング (または、マッピング デザイン ファイル *.mfd) を入力、出力、データに適用される必要のあるその他の、並べ替え、カスタム関数などの中間処理ステップを定義することのできる MapForce で視覚的に作成します。マッピングの準備ができると、下の方法の1つを使用して、MapForce Server で実行します:

- MapForce が作動する Windows マシンでは、MapForce Server 実行ファイル (.mfx) にマッピングをコンパイルすることができます。 .mfx ファイルは、大まかにサーバー環境内の実行のためパッケージされたデータマッピングとして称することができます。このようなファイルを MapForce Server が実行されているサポートされている異なるプラットフォーム全体を含むオペレーティングシステムにコピーすることができます ([システムの必要条件](#) を参照してください)。On サーバー マシン上では、.mfx ファイルを MapForce Server のコマンドラインインターフェイス、または、MapForce Server API を使用して実行することができます



- MapForce が作動する Windows マシンでは、MapForce Server と FlowForce Server がインストールされているサーバー マシンにマッピングをデプロイします。サーバー マシンはオペレーティングシステムとは異なることができます ([次を参照してください](#) [システムの必要条件](#))。マッピングは、このようにデプロイされ、FlowForce Server 関数となり、予定された、またはトリガーをベースにしたジョブが作成されます。マッピングが FlowForce Server ジョブとしてマッピングが実行される場合、他のジョブのサブステップとしてチェーンされている、または、電子メールの送信、シェルスクリプトの実行など、ワークフローの一部として、または Web サービスとしても公開することができます。



このシナリオに関する詳細は、FlowForce Server ドキュメンテーション
(<http://manual.altova.com/ja/FlowForceServer>) を参照してください。

MapForce Server 実行ファイルとしてコンパイルされたマッピングを実行する

1. MapForce Enterprise および Professional でコンパイルするマッピングを開きます。
2. 「ファイル」メニューから「MapForce Server 実行ファイルにコンパイルする」をクリックし、保存先のディレクトリを開きます。
3. .mfx ファイルを保存先のディレクトリにおよびサーバーにコピーします。
4. コマンドラインインターフェイスの「run」コマンドを呼び出し (次を参照してください: コマンドラインの使用法)、および MapForce Server API の同様のメソッドを呼び出します (次を参照してください: [MapForce Server API](#))。

FlowForce Server にデプロイされたマッピングを実行する

1. MapForce Enterprise および Professional でデプロイするマッピングを開きます。
2. マッピングの変換言語 (実行エンジン) が、Built-in に設定されていることを確認してください。実行エンジンを Built-in に変更する場合は、メニューコマンド「出力 | Built-In 実行エンジン」を選択、および「Built-In 実行エンジンの選択」() ツールバーボタンをクリックします。
3. 「ファイル」メニューから「FlowForce Server にデプロイする」をクリックします。
4. サーバー接続の詳細 (ホスト、ポート)、FlowForce 資格情報、FlowForce コンテナの保存先を入力します。FlowForce ジョブをブラウザ内で作成する場合は「Web ブラウザーを開いて新しいジョブを作成する」を選択します。FlowForce ジョブを後で作成することもできます (次のステップも参照してください)。
5. ブラウザーを開き、FlowForce Server Web 管理インターフェイスにログイン、マッピングをデプロイしたコンテナに移動します (前のステップを参照してください)。このステップは「新規ジョブを作成するために Web ブラウザーを開く」オプションが前のステップで選択されている場合必要ありません。
6. トリガー、パラメーター、および追加実行ステップを含む FlowForce Server ジョブを定義します (例に関しては、FlowForce Server のドキュメントを参照してください: <https://www.altova.com/ja/documentation>)。ジョブが実行のために構成されている場合、基となるマッピングの変換が実行されると、マッピングの出力が作成されます。

メモ MapForce Server がマッピングがデザインされたマシン以外のマシンで実行される場合、新しいターゲットでの実行環境で入力ファイルへのパスが調整されデータベース接続の詳細が正確であることを確認してください。例えば、マッピングセルがデータベースを呼び出し、データベースドライバーを必要とする場合、マッピングの実行に成功するためにドライバーはターゲット環境にもインストールされている必要があります。データベース接続の詳細を表示および調整する場合は、MapForce 内のデータベースコネクタを右クリックし、「プロパティ」を選択します。MapForce 内のマッピングデザインに変更が加えられると、MapForce Server 実行ファイル(.mfx) を再コンパイル、および場合により、FlowForce Server に再度デプロイすることを忘れないでください。詳細に関しては次を参照してください: [サーバー実行のためにマッピングを準備する](#)。

3.1 サーバーでの実行のためのマッピングの準備

MapForce を使用してデザインされ、レビューされるマッピングは、現在使用中のマシンと(データベースなどの)オペレーティングシステムの外部にあるリソースを参照することができます。更に、MapForce 内では、全てのマッピングのバカデフォルトでWindows スタイルの規則に従っています。また、MapForce Server が作動するサーバーでは、マッピングがデザインされたマシンと同じデータベースをサポートしない場合があります。このため、サーバー環境でマッピングを実行するには、また、ターゲットマシンがソースマシンと異なる場合は特に、若干の準備が必要とされます。

メモ ソースマシンという用語は、MapForce がインストールされているマシンを指し、ターゲットマシンは、MapForce Server、または、FlowForce Server がインストールされているマシンを指します。最もシンプルなシナリオは、ソースマシンとターゲットマシンが同じコンピューターの場合を指します。最も複雑なシナリオは、MapForce がWindows マシンで作動し、MapForce Server、または、FlowForce Server がLinux、または、macOS マシンで作動する場合を指します。

FlowForce Server にデプロイする前にMapForce 内でマッピングの検証を常に行うことが奨励されます。または、MapForce Server 実行ファイルコンパイルすることが奨励されます。

MapForce Server がスタンドアロンで(FlowForce Server 無しで)作動する場合、必要とされるライセンスは、以下のとおりです:

- ソースマシン上で、MapForce Enterprise、または、Professional Edition がマッピングをデザインし、サーバー実行ファイル(.mfx)にコンパイルするため必要とされます。。
- ターゲットマシン上で、MapForce Server、または、MapForce Server Advanced Edition がマッピングを実行するため必要とされます。

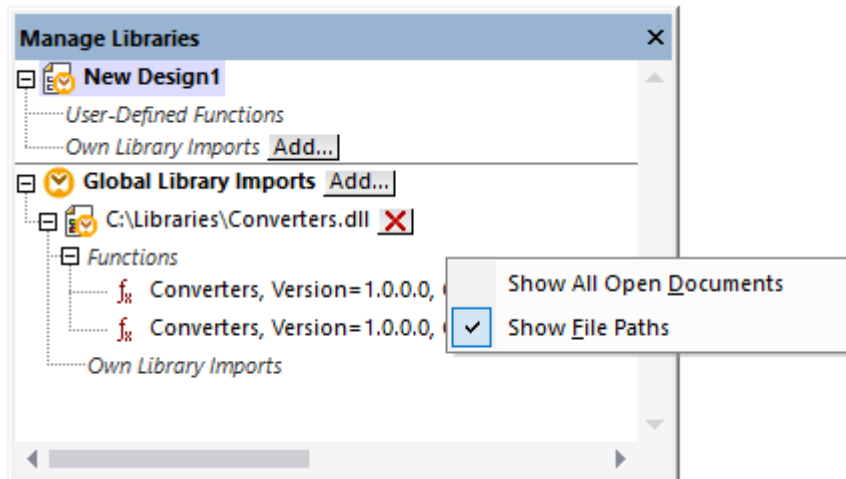
MapForce Server がFlowForce Server 管理下で作動する場合、次の条件が適用されます:

- ソースマシン上で、MapForce Enterprise、または、Professional Edition がマッピングをデザインして、ターゲットマシンにデプロイするため必要です。。
- MapForce Server とFlowForce Server がターゲットマシンにインストールされている必要があります。MapForce Server の役割は、マッピングの実行であり、FlowForce の役割はマッピングを、スケジュールまたは、オンデマンドの実行、としての実行、エラーの処理、条件付きの処理、電子メールの通知などの機能を活用することのできるジョブとして使用できるようにすることです。
- FlowForce Server は構成されているネットワークアドレスとポートで作動している必要があります。すなわち、「FlowForce Web Server」サービスが開始され、構成されており、HTTP (または、構成されている場合はHTTPS)クライアントからの接続をファイアウォールにブロックされることなく受け入れる必要があります。「FlowForce Server」サービスも開始され、指定されたアドレスとポートで作動されている必要があります。
- コンテナの1つのパーミッションが与えられているFlowForce Server ユーザーアカウントを所有している必要があります(デフォルトでは /public コンテナは認証済みのユーザーであればアクセスすることができます)。

全般的な注意点

- スタンドアロン MapForce Server で作動するターゲットマシン上のマッピングを作動するには、マッピングにより参照される全ての入力ファイルをターゲットマシンにコピーする必要があります。MapForce Server がFlowForce Server 管理化で作動する場合、ファイルを手動でコピーする必要はありません。この場合、インスタンスとスキーマファイルはターゲットマシンにデプロイされたパッケージ内に含まれます。。
- マッピングが特定のデータベースドライバを必要とするデータベースコネクタが含まれる場合、このようにドライバはターゲットマシンにもインストールされている必要があります。例えば、マッピングがMicrosoft Access データベースからデータを読み取る場合、Microsoft Access、または、Microsoft Access Runtime がターゲットマシンにもインストールされている必要があります(<https://www.microsoft.com/en-us/download/details.aspx?id=50040>)。
- マッピングを非 Windows プラットフォームにデプロイする場合、ADO、ADO.NET とODBC データベース接続は自動的にJDBC に変更されます。ネイティブ SQLite とネイティブ PostgreSQL 接続は保存され、追加の構成を必要としません。「データベース接続」も参照してください。
- (例えば、.dll、または、.class ファイルなどの)カスタム関数呼び出しをマッピングが含む場合、このような依存は、ランタイム前に既知ではないため、マッピングと共にデプロイされません。この場合、これを手動でターゲットマシンにコピーすることができます。サー

バー上の.dll または.class ファイルのパスはMapForce 内の「ライブラリの管理」ウィンドウと同じである必要があります。例



- マッピングの一部は、ワイルドカードパスを使用して複数の入力ファイルを読み取ります。この場合、入力ファイル名はランタイム前に既知ではないため、デプロイされません。マッピングの実行に成功するには、入力ファイルがターゲットマシン上に存在する必要があります。
- マッピングの出力パスにディレクトリが含まれている場合、これらのディレクトリはターゲットマシン上に存在している必要があります。それ以外の場合、マッピングを実行するとエラーが生成されます。この振る舞いは「一時的なファイルに出力を生成する」オプションが有効化されている場合既存ではないディレクトリが自動的に生成されるMapForce とは異なります。
- マッピングがクライアント証明書を使用するHTTPS 認証を必要とするWeb サービスを呼び出す場合、証明書をターゲットマシンに手動で移動する必要があります。次も参照してください：[デジタル署名の管理](#)。
- Microsoft Access とSQLite などのファイルをベースとしたデータベースにマッピングが接続される場合、データベースファイルは手動でターゲットマシンに移動、または参照することのできる共有されるディレクトリに保存される必要があります。次を参照「ファイルをベースとしたデータベース」。

パスをポータブルにする方法

サーバー上でマッピングを実行する場合、マッピングが適用することのできる変換パスを使用していること、および、サポートされるデータベース接続を使用していることを確認してください。

パスを非 Windows オペレーティングシステムへ移動可能にするためには、MapForce 内でマッピングをデザインする際に絶対パスの代わりに相対パスを使用してください。

- 希望するマッピングデザインファイル(.mfd) をWindows 上のMapForce で開きます。
- 「ファイル」メニューでマッピングの設定を選択し、チェックされている場合「生成されたコード内でパスを絶対パスにする」チェックボックスをクリアします。
- 各マッピングコンポーネントの右側にプロパティダイアログボックスを開き、(例えば、コンポーネントのタイトルバーをダブルクリックして) 全てのファイルパスを絶対パスから相対パスに変更します。「MFD ファイルに対して全てのパスを相対的に保存する」チェックボックスを選択します。便宜上、マッピングと同様全ての入力ファイルとスキーマを同じファイルにコピーし、ファイル名で参照することができます。

マッピングのデザイン時に相対および絶対パスを処理するにはMapForce ドキュメントを参照してください。

更に MapForce Server とFlowForce Server は、すべての相対パスが解決される、いわゆる「作業ディレクトリ」をサポートします。。作業ディレクトリは以下のとおりマッピングのランタイムに指定することができます。

- FlowForce Server では、ジョブの「作業ディレクトリ」パラメータを編集します。
- MapForce Server API では COM と.NET API のWorkingDirectory プロパティにより、または Java API の setWorkingDirectory メソッドにより行うことができます。

- MapForce Server コマンドライン では、作業ディレクトリはコマンドシェルの現在のディレクトリです。

データベース接続

ADO、ADO.NET、および ODBC 接続は Linux と macOS マシンではサポートされていないことに注意して下さい。このため、ターゲットマシンが Linux または macOS の場合、このような接続は、マッピングが FlowForce にデプロイされると、または、MapForce Server 実行ファイルにマッピングをコンパイルすると、JDBC に変換されます。この場合、マッピングをデプロイ、または、サーバー実行ファイルにマッピングをコンパイルする前に以下のオプションを使用することができます。

- MapForce 内では、データベースへの JDBC 接続を作成します
- MapForce 内では、データベースコンポーネントの「JDBC 固有の設定」セクション内の JDBC データベース接続の詳細を入力してください。(データベースコンポーネント設定を参照してください)。

マッピングが PostgreSQL または SQLite データベースへのネイティブな接続を使用する場合、ネイティブな接続が保存され、JDBC 変換は発生しません。マッピングが Microsoft Access と SQLite などのファイルベースのデータベースに接続される場合、追加構成が必要とされます。下記の「ファイルベースのデータベース」を参照してください。

JDBC 接続を使用してマッピングを起動することは、Java Runtime Environment または Java Development Kit がサーバーマシンにインストールされていることを必要とします。Oracle JDK または Oracle OpenJDK などのオープンソースビルドであることができます。

- JAVA_HOME 環境変数は JDK インストールディレクトリを指している必要があります。
- Windows 上では、Windows レジストリ内で検出された Java Virtual Machine ノードは JAVA_HOME 変数より高い優先順位を有します。
- JDK プラットフォーム(64ビット、32ビット)は MapForce Server を同じである必要があります。それ以外の場合は、理由があるエラーが挙げられる可能性があります。「JVM へのアクセスが不可能です。」

Linux または macOS 上での JDBC 接続のセットアップ

1. データベースベンダーにより提供される JDBC ドライバーをダウンロードして、オペレーティングシステムにインストールします。オペレーティングシステムが 32 ビットで動作する場合は 32 ビット版バージョンを、オペレーティングシステムが 64 ビットで動作する場合は 64 ビット版バージョンを選択してください。
2. 環境変数を JDBC ドライバーがインストールされている場所に設定してください。通常、CLASSPATH 変数と、必要があれば、その他複数の変数を設定します。どの特定の環境変数が構成されるかを検出するには、JDBC ドライバーに付随するドキュメントを確認してください。

メモ macOS では、システムは JDBC ライブラリが `/Library/Java/Extensions` ディレクトリにあることを必要とします。ですから、JDBC ドライバーをこの場所でアンパックすることが奨励されます。それ以外の場合、システムを構成し、JDBC ライブラリが JDBC ドライバーをインストールしたパスを検索するようになる必要があります。

macOS 上での Oracle Instant Client 接続

macOS 上で、Oracle Database Instant Client を介して Oracle データベースに接続する場合これらの命令を適用することができます。必要条件:

- Java 8.0 または、以降がインストールされている必要があります。Mac マシンが Java 8 以前の Java バージョンで動作している場合、「JDBC Thin for All Platforms」ライブラリを介して接続し、以下の命令を無視することも可能です。
- Oracle Instant Client がインストールされている必要があります。Oracle Instant Client を Oracle の公式ダウンロードページからダウンロードすることができます。Oracle ダウンロードページでは複数の Instant Client パッケージがあることに注意してください。Oracle Call Interface (OCI) サポートのあるパッケージが選択されていることを確認してください(例えば Instant Client Basic)。また、オペレーティングシステムが 32 ビットで動作する場合は 32 ビット版バージョンを、オペレーティングシステムが 64 ビットで動作する場合は 64 ビット版バージョンを選択してください。

ダウンロードした後、Oracle Instant Client をアンパックし、インストーラと共にパックされているプロパティリストファイル(.plist) を編集すると、以下の環境変数が対応するドライバへのパスをポイントします。例えば

変数	サンプル値
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
パス	\$PATH:/opt/oracle/instantclient_11_2

メモ 使用中のオペレーティングシステムで Oracle Instant Client ファイルがインストールされている場所で上記のサンプル値を編集してください。

ファイルをベースにしたデータベース

Microsoft Access と SQLite などのファイルをベースにしたデータベースは FlowForce Server にデプロイされるパッケージ内に含まれません。または、コンパイルされた MapForce Server 実行ファイルにも含まれません。このため、ソースとターゲットマシンが同じでは内場合、次のステップに従ってください。

1. MapForce 内では、マッピングを右クリックして、「生成されたコード内でパスを絶対パスにする」のチェックボックスのチェックを解除します。
2. マッピング上のデータベースコネクタを右クリックし、相対パスを使用してデータベースファイルへの接続を追加します。。マッピングデザイン(.mfd ファイル) をデータベースファイルと同じディレクトリに保存し、ファイル名別相対パスを使用してマッピングから参照することにより、パスに関連した問題を回避することができます。
3. データベースファイルをターゲットマシン内のディレクトリにコピーします（「作業ディレクトリ」と呼ぶこととします）。以下に表示されるようにマッピングを実行する際必要とされるため、このディレクトリを忘れないでください。

このようなマッピングをサーバー上で実行するには以下を行います：

- FlowForce Server コントロール下で作動する MapForce Server によりマッピングが実行される場合、FlowForce Server ジョブが前に作成された作業ディレクトリをポイントするように構成します。データベースファイルは作業ディレクトリ内に存在する必要があります。例えば「Web サービスとしてジョブを公開する」を参照してください (http://manual.altova.com/FlowForceServer/FlowForceServerAdvanced/fs_example_web_service.htm)。
- コマンドラインでスタンドアロン MapForce Server によりマッピングが実行される場合、MapForce Server 上で run コマンドを呼び出す前に、現在のディレクトリを作業ディレクトリに変更します（例えば `cd path\to\working\directory`）。
- MapForce Server API によりマッピングが実行される場合、作業ディレクトリをプログラミング的にマッピングを実行する前に設定します。これを効果的に行うために、COM と .NET API 内の MapForce Server オブジェクトのためプロパティ `WorkingDirectory` を使用することができます。Java API 内では、メソッド `setWorkingDirectory` を使用することができます。

ローカルネットワーク上の Windows マシン上にソースとターゲットマシンが存在する場合、共有されるディレクトリからデータベースファイルを読み取るためにマッピングを以下のように構成することができます：

1. ソースとターゲットマシンによりアクセスすることのできる共有されるディレクトリにデータベースファイルを保管します。
2. マッピング上のデータベースコネクタを右クリックし、絶対パスをしようしてデータベースファイルに接続を追加します。

グローバルリソース

マッピングが直接のパスまたはデータベース接続の代わりにグローバルリソースへの参照を含む場合、サーバー側でもグローバルリソースを使用することができます。MapForce Server 実行可能ファイル(.mfx) にマッピングをコンパイルする場合、グローバルリソースへの参照は変更されないため、マッピングランタイムにサーバー側で提供することができます。FlowForce Server にマッピングをデプロイする場合、リソースをサーバー上で使用するかを任意で選択することができます。

マッピング（または FlowForce Server の場合マッピング関数）の実行に成功するために、グローバルリソースとして提供する実際のファイル、フォルダー、またはデータベース接続の詳細はサーバー環境で互換性を持つ必要があります。例えば、マッピングが Linux サーバーで動作する場合、ノスのために Linux 変換を使用する必要があります。同様に、データベース接続として定義されているグローバルリソースはサーバーマシンで利用が可能である必要があります。

詳細に関しては [グローバルリソース](#) を参照してください。

XBRL タクノミ パッケージ

マッピングが FlowForce Server への XBRL タクノミ パッケージを参照する場合 MapForce はマッピングから全ての外部レファレンスを集め、現在の構成と現在インストールされているタクノミ パッケージを使用して解決します。タクノミ パッケージを指す解決された外部レファレンスが存在する場合、タクノミ パッケージはマッピングと共にデプロイされます。FlowForce Server は、そのパッケージを、デプロイ時と同様、マッピングを実行するために使用します。FlowForce Server により使用されるタクノミ パッケージを更新するには、MapForce 内で変更し、マッピングをもう一度デプロイします。

MapForce Server のルートカタログは、ターゲットマシン内でタクノミの解決方法に影響を与えます。ルートカタログは MapForce Server インストールディレクトリに対して相対的に以下のパスで見つけることができます: **etc/RootCatalog.xml**。

MapForce Server のルートカタログがこのようなカタログを含んでいない場合、または、同 URL プレフィックスのために定義されているパッケージを含まない場合、マッピングと共にデプロイされるタクノミ パッケージが使用されます。MapForce Server のルートカタログはデプロイされたタクノミよりも優先順位が高いです。

MapForce Server がスタートアップで (FlowForce Server 無しで) 動作する場合、次に用いてマッピングにより使用されるルートカタログを指定することができます:

- コマンドラインでは、オプション `-catalog` を `run` コマンドに追加することが可能です。
- MapForce Server API で、メソッド `SetOption` を呼び出し、文字列 `catalog` を最初の引数として、ルートカタログを 2 番目の引数として与えます。

マッピングがテーブルリンクベースを使用する XBRL コンポーネントである場合、タクノミ パッケージ、または、タクノミ パッケージ構成ファイルがランタイムでマッピングに提供される必要があります:

- MapForce Server コマンドラインにて、オプション `--taxonomy-package`、または、`--taxonomy-packages-config-file` を `run` コマンドに追加します。
- MapForce Server API 内で、メソッド `SetOption` を呼び出します。最初の引数は、「taxonomy-package」、または、「taxonomy-packages-config-file」である必要があります。2 番目の引数は、タクノミ パッケージ (、または、タクノミ パッケージ構成) ファイルへの実際のパスである必要がありますファイル。

3.2 グローバルリソース

Altova グローバルリソースはファイル、フォルダー、またはデータベースへのポータブルなレファレンスです。グローバルリソースとして保管されると、パスとデータベース接続の詳細は再利用できるようになり Altova アプリケーション全体で使用可能になります。例えば、複数の Altova デスクトップアプリケーション内で同じファイルを頻繁に開く必要がある場合、グローバルリソースとして定義すると便利な場合があります。このようにすると、「ファイルを開く」ダイアログボックスから対応するグローバルリソースを開くことができるため、ファイルパスを覚えておく必要がありません。この方法はファイルパスが変更されると、パスの一部を変更するのみに比べ利点があります。

グローバルリソースの一般的な使用法は、データベース接続を一度定義し、グローバルリソースをサポートする全ての Altova アプリケーションで再利用することです。例えば、MapForce マッピングがデザインされるマシンでデータベース接続を作成し、MapForce Server がマッピングを実行するマシン上で同じ接続を再利用することができます（これは、一部の例外を除き、両方のマシンで同じデータベースクライアントソフトウェアがインストールされている場合があります）。

任意で、「構成」として既知の同じグローバルリソースの複数のバージョンを作成することができます。これによりファイル、またはフォルダー（またはデータベース）を必要に応じて変更することができます。例えば、「開発」と「生産」の構成を持つデータベースリソースを作成することができます。MapForce を例に挙げると、マッピングをレビューする前に、希望する構成をマッピングランタイムコマンドラインパラメーターとして示すことにより、開発、または生産データベースからデータを簡単に抽出することができます。

次の Altova デスクトップアプリケーションからグローバルリソースを作成することができます: Altova Authentic、DatabaseSpy、MobileTogether Designer、MapForce、StyleVision、および XMLSpy。サーバー製品では、次の Altova サーバーアプリケーションでグローバルリソースを作成することができます: FlowForce Server、MapForce Server、RaptorXML Server、RaptorXML+XBRL Server。

グローバルリソースの作成に関する詳細は、MapForce ドキュメントの「Altova グローバルリソース」チャプターを参照してください。

MapForce Server 内のリソース

MapForce Server 実行可能ファイル(.mfx) にマッピングをエンコードすると、使用されるグローバルリソースレファレンスは解決されず保管されます。これはマッピングの実行に成功するためにサーバー側でこれらのレファレンスを提供する必要があることを意味します。具体的には、MapForce Server 内では、グローバルリソースを使用する .mfx ファイルを実行するために以下が必要とされています:

1. グローバルリソースが定義されているファイル。デフォルトでは、すべてのグローバルリソースは、MapForce がインストールされているマシン上の **Documents\Altova** フォルダー内の **GlobalResources.xml** というファイル内で定義されています。MapForce Server が作動するマシンでこのファイルをコピーして、必要に応じて複数のこのようなファイルを作成することができます。
2. グローバルリソース構成名。各グローバルリソースは、構成名により識別されます。デフォルト名は「デフォルト」ですが、必要に応じて追加の構成を作成することができます。「デフォルト」以外の希望する構成名を指定する必要があります。

MapForce 内では、グローバルリソースファイルパスとグローバルリソース構成名は、グラフィカルなユーザーインターフェイスから変更、および設定することができます。MapForce Server 内では、これらはマッピングランタイムに設定されます。

- コマンドラインインターフェイスを使用してマッピングを実行すると、`run` コマンドの後、オプション `--globalresourceconfig` と `--globalresourcefile` を設定します。例:

```
C:\Program Files (x86)\Altova\MapForceServer2021\bin\MapForceServer.exe run
SomeMapping.mfx --globalresourcefile="C:\Users\me\Documents\Altova\GlobalResources.xml" --globalresourceconfig="Default"
```

- MapForce Server API を使用してマッピングを実行する場合、`Run` メソッドを呼び出す前に、メソッド `SetOptions` を2回呼び出してください。最初の呼び出しは、オプションとしてグローバルリソースXMLファイルパスを提供するために必要です。2番目の呼び出しはグローバルリソース構成名を提供するために必要です。詳細に関しては [MapForce Server API](#) を参照してください。

3.3 動的な認証

MapForce 内では、ユーザー名とパスワードを入力パラメータとして受け入れるために基本的な HTTP 認証のための Web サービスを呼び出すマッピングを構成することができます。動的な認証の構成に関する詳細は、MapForce ドキュメント (<https://www.altova.com/ja/documentation>) を参照してください。

動的な認証のためにマッピングを構成する場合、対応するユーザー名とパスワードはマッピングのランタイムパラメータとして提供される必要があります。これはマッピングへ他のパラメータの型を提供することとは違いはありません。例えば、MapForce Server をコマンドラインで呼び出す場合、マッピングのための構文は下記のようなります:

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

場所:

- **<exec>** は MapForce Server 実行可能ファイルへのパスです。このパスは絶対的、または現在のディレクトリが実行可能と同じ場合、実行可能ファイル名のみを入力することができます。
- **username** と **password** は MapForce マッピング上の対応する入力パラメータの名前です。

MapForce Server API を呼び出す際、**Run** メソッドを **AddParameter** メソッドを呼び出す前に呼び出して、マッピングを認証することができます。例えば、C# では、アーカイブするコードは以下のようになります:

```
try
{
    Altova.MapForceServer.Server mfs = new Altova.MapForceServer.Server();
    mfs.AddParameter("username", "amdin");
    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);
}
```

username と **password** (**AddParameter** メソッドの最初の引数) は MapForce マッピング上の対応する入力パラメータの名前と同じである必要があります。

4 MapForce Server コマンドライン

MapForce Server をコマンドラインから作動するには、次のデフォルトの場所から実行可能ファイルを呼び出すことができます:

Linux	/opt/Altova/MapForceServer2021/bin/ mapforceserver
Mac	/usr/local/Altova/MapForceServer2021/bin/ mapforceserver
Windows	<ProgramFilesFolder>\Altova\MapForceServer2021\bin\MapForceServer.exe

CLI コマンドの使用方法和リスト

コマンドライン構文は以下の通りです:

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

- `--help` (短縮形 `--h`) は与えられたコマンドのヘルプテキストを表示します。コマンドに名前がない場合、実行可能ファイルの全てのコマンドがコマンドの短い説明と共にリストされます。
- `--version` は MapForce Server のバージョン番号を表示します。
- `<command>` は実行されるコマンドです。このセクションのサブセクション内でコマンドについて説明されています (のリストを参照してください)。
- `[options]` はコマンドのオプションです。これらのコマンドはリストされ、対応するコマンドと共に説明されています。
- `[arguments]` はコマンドの引数です。これらのコマンドはリストされ、対応するコマンドと共に説明されています。

▼ コマンドライン上の大文字と小文字の区別とスラッシュ

Windows 上の MapForceServer

Windows と Unix (Linux, Mac) 上の mapforceserver

- * 小文字 (`mapforceserver`) はすべてのプラットフォーム (Windows、Linux、および Mac) で使用することができますが、小文字 (`MapForceServer`) は Windows と Mac のみで使用することができます。
- * Linux と Mac 上ではスラッシュを使用し、Windows 上ではバックスラッシュを使用します。

CLI コマンド

使用することのできるコマンドは下にリストされており、このセクションのサブセクションで説明されています。

- [assignlicense](#): LicenseServer にライセンスを割り当て、MapForce Server にこのライセンスを割り当てます。
- [exportresourcestrings](#): XML ファイルに全てのアプリケーションリソース文字列をエクスポートします。
- [help](#): 引数 (または、引数が送信されていない場合、全てのコマンド) 内に送信されたコマンドに関する情報を表示します。
- [licenseserver](#): MapForce Server をローカルネットワーク上の LicenseServer に登録します。
- [run](#): MapForce に基づいてコンパイルされたマッピング実行ファイル (.mfx) を実行します。
- [setdeflang](#): MapForce Server のデフォルトの言語を設定します。
- [verifylicense](#): 現在の MapForce Server がライセンス供与済みかチェックし、オプションで特定のライセンスキーによりライセンス供与されているかを確認します。
- [version](#): MapForce Server のバージョンを表示します。

4.1 assignlicense (Windows only)

構文と説明

assignlicense コマンドは Windows 上のみで使用することができます (Linux または Mac システム上では使用することができません)。MapForce Server が登録されている Altova LicenseServer にライセンスファイルをアップロードし、MapForce Server にライセンスを割り当てます (licenseserver コマンドを参照してください)。ライセンスのパスを引数として取ります。コマンドによりライセンスの有効性をテストすることができます。

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

- `FILE` 引数はライセンスファイルのパスを取ります。
- `--test-only` オプションは LicenseServer にライセンスをアップロードし、ライセンスを検証しますが、MapForce Server にライセンスを割り当てません。

ライセンスに関する詳細は、LicenseServer ドキュメントを確認してください (<https://www.altova.com/manual/ja/AltovaLicenseServer/>)。

例

assignlicense コマンドのサンプル

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

- 上記の最初のコマンドは LicenseServer に指定されたライセンスをアップロードし、MapForce Server に割り当てます。
- 最後のコマンドは LicenseServer に指定されたライセンスアップロードし、MapForce Server に割り当てることなく検証します。

オプション

(使用できる場合) オプションは短いおよび長い書式でリストされています。短いおよび長い書式のためコマンドまたは2つのダッシュを使用することができます。オプションは値を取る、または、取らない場合があります。値をとる場合、以下ようになります: `--option=value`。値は引号無しで2つの場合で指定することができます: (i) 値文字列がスペースが含まれる場合、または、(ii) 引用符が必要とするオプションの説明内で明示的に指示されている場合。オプションがブール値の値を取り、ブールの値が指定されていない場合、オプションのデフォルトの値は `TRUE` です。 `--h, --help` オプションを使用してコマンドに関する情報を表示します。

▼ test-only [t]

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

値は `true|false` です。 `true` の場合、LicenseServer にライセンスファイルがアップロードされ、検証されますが、割り当てられることはありません。

4.2 exportresourcestrings

構文と説明

exportresourcestrings コマンドは特定の言語内のMapForce Server アプリケーションのリソース文字列を含むXML ファイルを出力します。使用することのできる言語は英語(en)、ドイツ語(de)、スペイン語(es)、フランス語(fr)、と日本語(ja)です。

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

- **LanguageCode** 引数は、出力 XML ファイル内のリソース言語の言語を与えます。これは **エクスポート言語** です。許可されているエクスポート言語は以下の通りです(それぞれの言語コードがカッコで囲まれています): 英語(en)、ドイツ語(de)、スペイン語(es)、フランス語(fr)、と日本語(ja)。
- **XMLOutputFile** 引数は出力 XML ファイルのパスと名前を指定します

ローカライゼーションの作成方法は以下のとおりです。

例

exportresourcestrings コマンドのサンプル

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

- 上のコマンドはMapForce Server のドイツ語リソース文字列を含むStrings.xml と呼ばれるファイルをc:\に作成します。

MapForce Server のローカライズされたバージョンの作成

選択された言語でMapForce Server のローカライズされたバージョンを作成することができます。5つのローカライズされたバージョン(英語、ドイツ語、スペイン語、フランス語、および日本語)はC:\Program Files (x86)\Altova\MapForceServer2021\bin フォルダ内で使用することができるため、作成する必要はありません。

ローカライズされたバージョンを以下のよう作成します:

1. **exportresourcestrings** コマンドを使用してリソース文字列を含むXML ファイルを作成します(上記のコマンド構文を参照してください)。このXML ファイル内のリソース文字列は、サポートされる以下の5つの言語のうちの1つです: コマンドと共に使用されるLanguageCode 引数に従い英語(en)、ドイツ語(de)、スペイン語(es)、フランス語(fr)、または日本語(ja)。
2. 5つのサポートされる言語の1つからのリソース文字列からターゲット言語に翻訳します。リソース文字列はXML ファイル内の <string> 要素のコンテンツです。{option} または { product} などの、中カッコ内の変数は翻訳しないでください。
3. ローカライズされたMapForce Server DLL ファイルを翻訳されたXMLファイルから生成するために、[Altova サポート](#) に連絡してください。
4. [Altova サポート](#) からローカライズされたDLLファイルを受信すると、DLL をC:\Program Files (x86)\Altova\<% APPFOLDER%\bin フォルダ内に保存します。DLL ファイルはフォームMapForceServer2021_lc.dll の名前を持ちます。名前の lc 部分は、言語コードが含まれています。例えば、MapForceServer2021_de.dll では、de の部分はドイツ語(Deutsch) のための言語コードです。
5. **setdeflang** コマンドを実行し、使用するMapForce Server アプリケーションと同様にローカライズされたDLL ファイルを設定します。setdeflang コマンドの引数のために、DLL 名の一部である言語コードを使用してください。

メモ Altova MapForce Server は5つの言語のためのサポートを提供しています: 英語、ドイツ語、スペイン語、フランス語、および日本語。これらの言語のローカライズされたバージョンを作成する必要はありません。これらの言語をデフォルトの言語として設定するには、MapForce Server のsetdeflang コマンドを使用します。

4.3 help

構文と説明

help コマンドは、ヘルプを必要とするコマンドの名前である単一の引数 (Command) を取ります。コマンドの構文、オプション、および他の関連する情報を表示します。Command 引数が指定されていない場合、説明と共に実行可能ファイルのすべてのコマンドがリストされます。

```
mapforceserver help Command
```

例

help コマンドのサンプルは **licenseserver** コマンドに関する情報を表示します。

```
mapforceserver help licenseserver
```

--help オプション

ヘルプ情報が必要とされるコマンドの **--help** オプションを使用してヘルプ情報を使用することもできます。下の2つのコマンドは同じ結果を生成します。

```
mapforceserver licenseserver --help
```

上記のコマンドは **licenseserver** コマンドの **--help** オプションを使用します。

```
mapforceserver help licenseserver
```

help コマンドは引数として **licenseserver** を取ります。

両方のコマンドは **licenseserver** コマンドに関するヘルプ情報を表示します。

4.4 licenseserver

構文と説明

licenseserver コマンドは MapForce Server を Server-Or-IP-Address 引数により指定されている Altova LicenseServer を使用して登録します。**licenseserver** コマンド実行されるのは、2つのサーバー (MapForce Server と LicenseServer) が同じネットワーク上で作動しており LicenseServer が作動している必要があります。MapForce Server を LicenseServer と登録する管理特権が必要になります。

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

- *Server-Or-IP-Address* 引数は、LicenseServer マシンの名前、または IP アドレスを取ります。

MapForce Server が LicenseServer へ登録されると、この件に関連するメッセージを受け取ります。メッセージは LicenseServer の URL も表示します。LicenseServer に移動して、MapForce Server にライセンスを割り当てることもできます。ライセンスに関する詳細は、LicenseServer ドキュメントを確認してください (<https://www.altova.com/manual/ja/AltovaLicenseServer/>)。

例

licenseserver コマンドのサンプル

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

上記のコマンドは、それぞれ `DOC.altova.com` という名前のマシンとユーザーのマシン (`localhost` と `127.0.0.1`) を Altova LicenseServer を作動するマシンとして指定します。双方の場合、指定されているマシン上の LicenseServer を使用してコマンドは MapForce Server を登録します。最後のコマンドは、コマンドを実行するためにサーバー実行可能ファイルを呼び出します。

オプション

(使用できる場合) オプションは短いおよび長い書式でリストされています。短いおよび長い書式のため、コマンドは2つのダッシュを使用することができます。オプションは値を取る、または、取らない場合があります。値をとる場合、以下ようになります: `--option=value`。値は引用符無しで2つの場合で指定することができます: (i) 値文字列にスペースが含まれる場合、または (ii) 引用符が必要とするオプションの説明内で明示的に指示されている場合。オプションがブール値の値を取り、ブールの値が指定されていない場合、オプションのデフォルトの値は `TRUE` です。 `--h`, `--help` オプションを使用してコマンドに関する情報を表示します。

▼ json [j]

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

値は `true|false` です。`true` の場合、マシン解読可能な JSON オブジェクトとして登録の結果を出力します。

4.5 run

`run` コマンドは引数として提供されている MapForce Server 実行ファイル(.mfx file) を実行します。MapForce Server 実行ファイルは MapForce を使用して作成されており、サーバー実行のためにコンパイルされたマッピングを表しています。

マッピングにより必要とされる入力ファイルは、MapForce 内でマッピングデザイン時に指定されるパスが存在することが期待されます。If MapForce Server が MapForce と同じオペレーティングシステム上で作動しない場合、マッピングにより必要とされる入力 .mfx ファイルはファイルと共にターゲットマシンにコピーされる必要があり、相対的なパスを使用して参照される必要があります。相対的、または絶対的パスを使用してマッピングを構成する詳細については、MapForce ドキュメントを参照してください (<https://www.altova.com/ja/documentation#mapforce>)。マッピングのデザイン方法に従い、他の前提条件が適用される場合があります。サーバー実行のためにマッピングを準備するを参照してください。

文字列などの単純な値をマッピングが返す場合、出力は `stdout` (標準の出力) ストリームに書き込まれます。他方、`stderr` (標準のエラー) ストリーム内で成功、またはエラーメッセージを使用することができます。成功、または失敗メッセージと共に標準出力ストリームをスクリーン上に表示することを希望しない場合、標準の出力、または標準のエラーストリーム(または両方) をファイルにリダイレクトします。`stdout` または `stderr` ストリームの双方がリダイレクトされない場合、両方とも組み合わされてスクリーン上に表示されます。

例えば、標準出力ストリームをファイルにリダイレクトする場合、次を使用します:

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

エラーストリームをファイルにリダイレクトする場合、次を使用します:

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

両方のストリームを同時にリダイレクトする場合、次を使用します:

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

ストリームのリダイレクトに関する関数の詳細に関しては、オペレーティングシステムのコマンドシェルのドキュメントを参照してください。

Syntax

Windows **MapForceServer** run [options] MfxFile

Linux **mapforceserver** run [options] MfxFile

Mac **mapforceserver** run [options] MfxFile

▼ コマンドライン上の文字種とスラッシュ

Windows 上での MapForceServer

Unix (Linux、Mac) 上での mapforceserver

* 小文字は (mapforceserver) 全てのプラットフォーム (Windows、Linux、および Mac) で使用することができますが、大文字と小文字 (MapForceServer) は、Windows および Mac のみでしか使用できません。

* Linux と Mac 上ではスラッシュを使用し、Windows 上では、バックスラッシュを使用してください。

Windows システム上では、\" 上でエドバックスラッシュと終了引用符の使用を避けてください。例 "C:\My directory\"。これら 2つの文字はコマンドラインパーサーにより二重引用符マークとして解釈されます。コマンドライン内で空白が発生し、引用符を使用する必要がある場合ダブルバックスラッシュ\\ を使用してください(例 "C:\My Directory\\")。または、スペースの使用、または結果的に空白引用符の使用を避けてください。

コマンドオプション

	--catalog	ルートカタログファイルにインストールされていないルートカタログファイルへの絶対パスを指定します。デフォルトの値はインストール済みのルートカタログファイルへの絶対パスです。 カタログファイル を参照してください。 フォーム --catalog=FILE
--cert	--certificatespath	Linux 上で作動するMapForce Server に対してこのオプションを適用することができます。マッピングにより必要とされる証明書ファイルが保管されているディレクトリへのパスを指定します。 フォーム --certificatespath=DIRECTORY デジタル証明書の管理 を参照してください。
--gc	--globalresourceconfig	マッピングがAltova グローバルリソースを使用する場合、このオプションを適用することができます(グローバルリソース を参照してください)。グローバルリソース構成(エイリアス)の名前を指定します。このオプションは--globalresourcefile オプションと共に使用される必要があります。 フォーム --gc=VALUE
--gr	--globalresourcefile	マッピングがAltova グローバルリソースを使用する場合、このオプションを適用することができます(グローバルリソース を参照してください)。グローバルリソース定義ファイルのパスを指定します。このオプションは--globalresourceconfig オプションと共に使用される必要があります。 フォーム --gr=FILE.
--l	--lang	メッセージの表示のために使用される言語です。 フォーム --lang=VALUE (en,de,ja,es, fr)
--p	--param	マッピングが入力パラメータを取るようデザインされている場合、このオプションを適用することができます。マッピング内で定義済みのパラメータに値を割り当てます。フォーム --param=ParamName:ParamValue。 --param スイッチは各パラメータの前に使用される必要があります。Use quotes if ParamName または ParamValue がスペースを含む場合引用符を使用してください。例 --p=company:"Nanonull Inc". 入力パラメータを取るマッピングに関する詳細は、MapForce ドキュメントを参照してください (https://www.altova.com/ja/documentation#mapforce)。
	--taxonomy-package	タクノミパッケージ 1.0 推奨内で説明されているとおり、追加 XBRL タクノミパッケージへの絶対パスを指定します。FILE の値はタクノミパッケージの場所を与えます。1つ以上のタクノミパッケージを指定するためオプションを複数回追加します。 フォーム --taxonomy-package=FILE
	--taxonomy-packages-config-file	XBRL タクノミパッケージをロードするために使用される TaxonomyPackagesConfig.json と呼ばれる構成ファイルへのパス

		を指定します。この構成ファイルは Altova XMLSpy、MapForce、または StyleVision のグラフィカルなユーザーインターフェイスから XBRL タクソノミパッケージを追加、削除、有効化、または無効化する都度更新されます。カスタム XBRL タクソノミパッケージを上記の製品を使用して追加することができ、ファイルは C:\Users\<username>\Documents\Altova で見つけることができます。 フォーム --taxonomy-packages-config-file=FILE
--	--	--

サンプル

このサンプルは Windows 上の MapForce Server を使用してマッピング実行ファイル(.mfx) を実行する方法について説明しています。このサンプル内で使用されているマッピングは入力ファイル「Employees.xml」を読み取り、2つの出力ファイル「PersonList.xml」と「Contacts.xml」を生成します。

まず、MapForce Server 実行可能(.mfx) ファイルを以下のように生成します:

1. MapForce を作動し、MapForce デザインファイル(.mfd ファイル)を開きます: C:\Users\<user>\Documents\Altova\MapForce2021\MapForceExamples\ChainedPersonList.mfd.
2. 「ファイル」メニューから「MapForce Server 実行ファイルにコンパイルする」をクリックします。
3. プロンプトされると、.mfx ファイルを C:\temp ディレクトリに保存します。マッピングが MapForce Server により実行される作業ディレクトリになります。

次に、コマンドラインプロンプトを開き、作業ディレクトリを C:\temp\ に変更します。

```
cd C:\temp
```

最後に ChainedPersonList.mfx を実行する次のコマンドを実行します。このサンプルでは、MapForce Server は絶対パスを使用して呼び出されています(相対的なパスを使用して呼び出す場合はシステムの PATH 環境変数に実行可能ファイルのパスを追加してください)。

```
"C:\Program Files (x86)\Altova\MapForceServer2021\bin\MapForceServer.exe" run  
ChainedPersonList.mfx
```

2つの出力ファイル(PersonList.xml と Contacts.xml) は作業ディレクトリ内に生成されます。重要な点は、マッピングが絶対パスを使用するように構成されていることです。このためにマッピングの実行に成功し、入力 Employees.xml ファイルが作業ディレクトリに存在することを必要としませんでした。Employees.xml ファイルは実際には上記の MapForce サンプルフォルダー内に存在し、絶対パスを使用して参照されています。パスが絶対、または相対パスとして扱われるかを指定するには、MapForce 内のマッピングを右クリックして、「マッピングの設定」を選択し、「生成されたコード内でパスを絶対的にする」チェックボックスを選択、またはクリアしてください。マッピング設定を変更する都度、マッピング .mfx を再コンパイルしてください。詳細に関しては、[サーバー実行のためにマッピングを準備する](#)を参照してください。

4.6 setdeflang

構文と説明

setdeflang コマンド (**sdl** と略される) は MapForce Server. のデフォルトの言語を設定します。英語 (**en**)、ドイツ語 (**de**)、スペイン語 (**es**)、フランス語 (**fr**)、および日本語 (**ja**) を使用することができます。コマンドは必須の *LanguageCode* 引数を取ります。

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

- *LanguageCode* 引数が必須で、MapForce Server のデフォルトの言語を設定します。対応する値は以下の通りです: **en**, **de**, **es**, **fr**, **ja**.
- **--h**, **--help** オプションを使用してコマンドの情報を表示します。

例

setdeflang (sdl) コマンドのサンプル

```
mapforceserver sdl de
mapforceserver setdeflang es
```

- 最初のコマンド MapForce Server のデフォルトの言語をドイツ語に設定します。
- 2番目のコマンドは MapForce Server のデフォルトの言語をスペイン語に設定します。

オプション

--h, **--help** オプションを使用してコマンドに関する情報を表示します。

4.7 verifylicense (Windows only)

構文と説明

verifylicense コマンドは、使用中の製品がライセンス供与を受けているかチェックします。更に、`--license-key` オプションにより特定のライセンスキーが製品に既に割り当て済みかをチェックします。このコマンドは Windows 上でのみサポートされています。Linux または Mac システム上ではサポートされていません。

```
mapforceserver verifylicense [options]
```

- MapForce Server に割り当てられた特定のライセンスが、`--license-key` オプションの値としてライセンスキーを与えるかをチェックします。

ライセンスに関する詳細は、LicenseServer ドキュメントを確認してください (<https://www.altova.com/manual/ja/AltovaLicenseServer/>)。

例

verifylicense コマンドのサンプル

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

- 最初のコマンドは MapForce Server がライセンス供与されているかをチェックします。
- 2 番目のコマンドは MapForce Server が `--license-key` オプションで指定されているライセンスキーと共にライセンス供与されているかをチェックします。

オプション

(使用できる場合) オプションは短いおよび長い書式でリストされています。短いおよび長い書式のため、コマンドラインでオプションを使用することができます。オプションは値を取る、または、取らない場合があります。値をとる場合、以下ようになります: `--option=value`。値は引用符無しで2つの場合で指定することができます: (i) 値文字列にスペースが含まれる場合。または、(ii) 引用符が必要とするオプションの説明内で明示的に指示されている場合。オプションがブール値の値を取り、ブールの値が指定されていない場合、オプションのデフォルトの値は **TRUE** です。 `--h`, `--help` オプションを使用してコマンドに関する情報を表示します。

▼ license-key []

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

MapForce Server がこのオプションの値としてライセンスキーと共にライセンス供与されているか確認します。

4.8 version

構文と説明

version コマンドはMapForce Server のバージョンを表示します。

```
mapforceserver version
```

例

version コマンドのサンプル

```
mapforceserver version
```

5 MapForce Server API

MapForce Server は、.NET、Java または COM をベースとしたコードからのプログラミングからのアクセスするアプリケーション プログラミング インターフェイス (API) を与えます。

各プラットフォームの説明に関しては、以下のトピックを参照してください！

- [.NET インターフェイスに関して](#)
- [COM インターフェイスについて](#)
- [Java インターフェイスについて](#)

API の技術的説明に関しては、以下のトピックを参照してください！

- [API レファレンス \(COM, .NET\)](#)
- [API レファレンス \(Java\)](#)

5.1 .NET インターフェイス

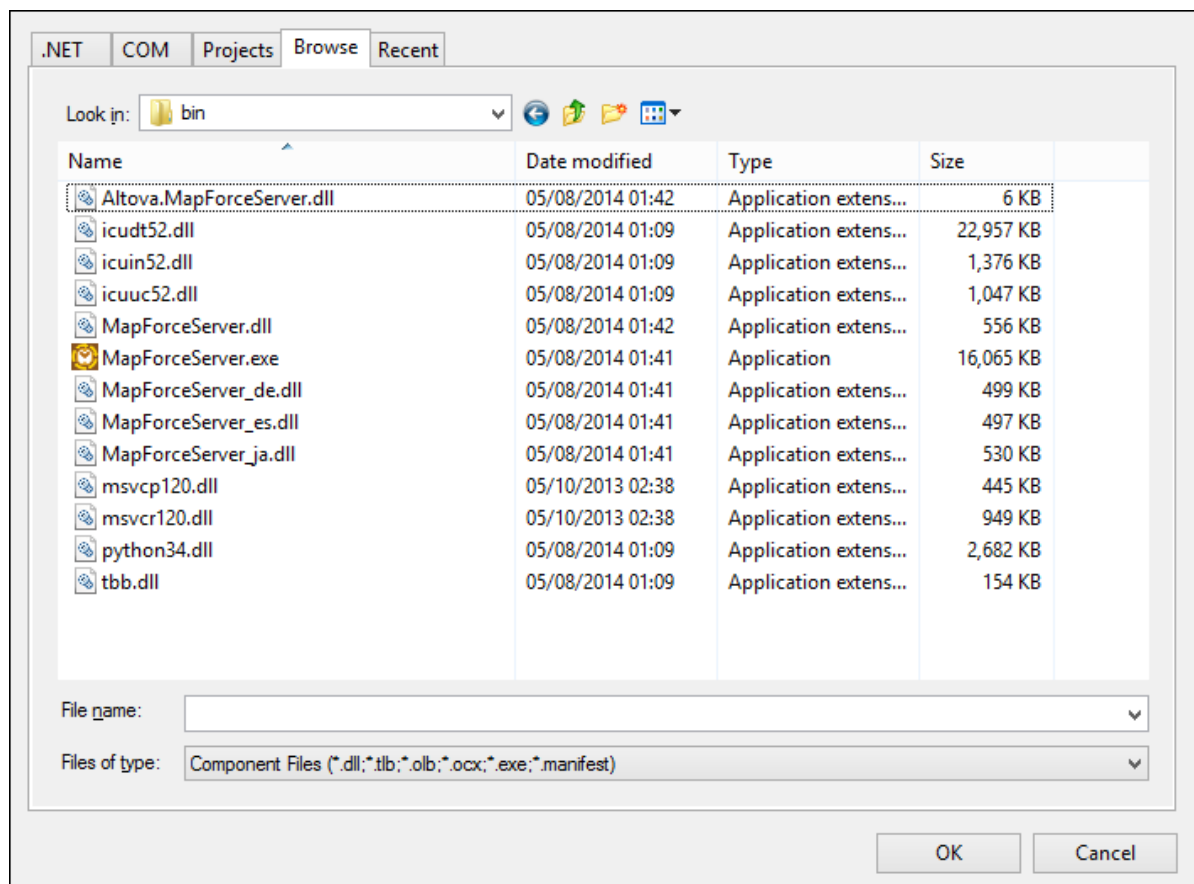
.NET インターフェイスはCOM インターフェイスの周りのラッパーとして構築されます。Altova により署名済みのプライマリアセンブリとして提供され、名前空間 `Altova.MapForceServer` を使用します。インストール中、MapForce Server はサーバーオブジェクトにより自動的に登録されるため、手動の登録は必要なくなります。

MapForce Server in your .NET プロジェクト内でを使用するには、`Altova.MapForceServer.dll` ファイルの参照を追加して示される通り追加します。`Altova.MapForceServer.dll` は MapForce Server インストールフォルダーの `bin` フォルダー内で見つけることができます。この `.dll` ファイルはグローバルアセンブリキャッシュ (GAC) は MapForce Server インストール中に自動的に追加されます。

メモ .NET Framework 4.0 の以前では GAC は `%windir%\assembly` ディレクトリ内で見つけることができました。Starting with .NET Framework 4.0 からは GAC は `%windir%\Microsoft.NET\assembly` ディレクトリ内に存在します。`%windir%` 部分はオペレーティングシステムのディレクトリを表しています。具体的には `C:\Windows`。

Visual Studio .NET プロジェクト内で MapForce Server DLL に参照を追加する方法:

1. Visual Studio 内で開かれている .NET プロジェクトから「プロジェクト | レファレンスを追加する」をクリックします。



2. ブラウザから、フォルダー: `<MapForceServer application folder>/bin` を参照して、`Altova.MapForceServer.dll` を選択し、「OK」をクリックします。

Altova.MapForceServer アセンブリの構造を Visual Studio オブジェクト ブラウザーを使用して確認することができます。オブジェクト ブラウザーを表示するには、「表示」メニューからオブジェクト ブラウザーをクリックします。

5.1.1 C# サンプル

次のサンプルは C++ コードからのマッピング マッピング実行ファイル(.mfx) の実行方法を説明しています。Windows 上ではサンプルファイルは以下のパスで見つけることができます: C:\Program Files\Altova\MapForceServer2021\etc\Examples。

必要条件

- MapForce Server サーバーがインストールされておりライセンス供与済みであること。
- 新規の Visual Studio プロジェクトを作成する場合、MapForce Server アセンブリ参照を追加します([.NET インターフェイス](#) についてを参照してください)。サンプルが MapForce Server アセンブリを既に参照しているため既存の MapForce Server API サンプルを動作している場合このステップをスキップすることができます。
- Visual Studio の「Build」メニューで、「Configuration Manager」を選択し、正確なビルドプラットフォームを設定します。例えば、**Debug | x86** (または適応できる場合 **Debug | x64**)。"Any CPU" をプラットフォームとして使用しないでください。
- MapForce Server 64 ビットがインストールされている場合、(下のサンプルのように) API を呼び出すアプリケーションが Visual Studio 内で 64 ビット プラットフォームのためにビルドされる必要があります。MapForce サーバー実行可能ファイルへのパスはコード内で適切に調節される必要があります。

サンプルソリューションは管理者の特権を必要とする「プログラムファイル」ディレクトリ内に存在します。Visual Studio を管理者として動作するか、管理者の特権を必要としない異なるフォルダーにソリューションをコピーしてください。

マッピングコードの実行

下のコードは 3 個のサーバー実行ファイル(.mfx) を実行します。下のテーブルは各 .mfx ファイルが期待する入力ファイルと、実行後に作成される出力をリストしています。

実行可能ファイル(.mfx)	入力	出力
TokenizeString.mfx	AltovaTools.xml	AltovaToolsFeatures.csv
SimpleTotal.mfx	ipo.xml	文字列
ClassifyTemperatures.mfx	Temperatures.xml	Temperatures_out.xml

Altova MapForce がインストールされている場合、更に詳しく理解するために .mfx ファイルがビルドされたオリジナルのマッピングを任意で確認することができます。これらはそれぞれ **TokenizeString1.mfd**、**SimpleTotal.mfd**、および **ClassifyTemperatures.mfd** と呼ばれます。respectively. 以下のディレクトリ内でマッピングを見つけることができます: C:\users\<user>\Altova\MapForce2021\MapForceExamples.

下サンプルは以下を行います:

- Altova.MapForceServer.Server の新規のインスタンスを作成します。これは後の作業の目的となります。
- 実行が行われる作業ディレクトリを設定します。相対パスを使用して参照する場合、入力ファイルはこのディレクトリ内に存在することが期待されます。出力ファイルはこのディレクトリ内にも作成されます(上のテーブルは参照してください)。
- これは **TokenizeString.mfx** を実行します。ファイル `Run` メソッドに対して引数として提供されます(パスは以前に設定された作業ディレクトリに対して相対的であることご注意ください)。実行に成功すると、マッピング出力を表す .csv ファイルが作業ディレクトリ内に作成されます。

- これは**SimpleTotal.mfx** を実行します。ファイルパスは作業ディレクトリに対して相対的です。このマッピングは文字列出力を生成するため**GetOutputParameter** メソッドを呼び出して文字列出力を取得します。
- これは**ClassifyTemperatures.mfx** を実行します。このマッピングは**AddParameter** メソッドを使用して提供される入力としてパラメータを期待します。

```
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);
    }
}
}
}

```

5.1.2 Visual Basic .NET サンプル

次のサンプルはVB.NET コードからのマッピング マッピング実行ファイル(.mfx) の実行方法を説明しています。Windows 上ではサンプル ファイルは以下の 場所で見つけることができます: **C:\Program Files\Altova\MapForceServer2021\etc\Examples**。

必要条件

- MapForce Server サーバーがインストールされておりライセンス供与済みであること。
- 新規の Visual Studio プロジェクトを作成する場合、MapForce Server アセンブリ参照を追加します([.NET インターフェイス](#) についてを参照してください)。サンプルが MapForce Server アセンブリを既に参照しているため既存の MapForce Server API サンプルを起動している場合このステップをスキップすることができます。
- Visual Studio の「Build」メニューで、「Configuration Manager」を選択し、正確なビルドプラットフォームを設定します。例えば、**Debug | x86** (または適応できる場合 **Debug | x64**)。"Any CPU" をプラットフォームとして使用しないでください。
- MapForce Server 64 ビットがインストールされている場合、(下のサンプルのように) API を呼び出すアプリケーションが Visual Studio. 内で 64 ビット プラットフォームのためビルドされる必要があります。MapForce サーバー実行可能ファイルへのパスはコード内で適切に調節される必要があります。

サンプルソリューションは管理者の特権を必要とする「プログラムファイル」ディレクトリ内に存在します。Visual Studio を管理者として

作動するか、管理者の特権を必要とし、異なるフォルダーにリネーションをコピーしてください。

マッピングコードの実行

下のコードは3個のサーバー実行ファイル(.mfx)を実行します。下のテーブルは各 .mfx ファイルが期待する入力ファイルと、実行後に作成される出力をリストしています。

実行可能ファイル(.mfx)	入力	出力
TokenizeString.mfx	AltovaTools.xml	AltovaToolsFeatures.csv
SimpleTotal.mfx	ipo.xml	文字列
ClassifyTemperatures.mfx	Temperatures.xml	Temperatures_out.xml

Altova MapForce がインストールされている場合、更に詳しく理解するために .mfx ファイルがエンパイルされたオリジナルのマッピングを任意で確認することができます。これらはそれぞれ **TokenizeString1.mfd**、**SimpleTotal.mfd**、および **ClassifyTemperatures.mfd** と呼ばれます。respectively. 以下のディレクトリ内でマッピングを見つけることができます: **C:\users\<user>\Altova\MapForce2021\MapForceExamples.**

下サンプルは以下を行います:

- **Altova.MapForceServer.Server** の新規のインスタンスを作成します。これは後の作業の目的となります。
- 実行が行われる作業ディレクトリを設定します。相対パスを使用して参照する場合、入力ファイルはこのディレクトリ内に存在することが期待されます。出力ファイルはこのディレクトリ内にも作成されます(上のテーブルも参照してください)。
- これは **TokenizeString.mfx** を実行します。ファイル **Run** メソッドに対して引数として提供されます(パスは以前に設定された作業ディレクトリに対して相対的であることを注意してください)。実行に成功すると、マッピング出力を表す .csv ファイルが作業ディレクトリ内に作成されます。
- これは **SimpleTotal.mfx** を実行します。ファイル **Run** メソッドに対して相対的です。このマッピングは文字列出力を生成するため **GetOutputParameter** メソッドを呼び出して文字列出力を取得します。
- これは **ClassifyTemperatures.mfx** を実行します。このマッピングは **AddParameter** メソッドを使用して提供される入力としてパラメータを期待します。

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\MapForceServer2021\bin\MapForceServer.exe"
'objMFS.ServerPath = "C:\Program
Files\Altova\MapForceServer2021\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
```

5.2 COM インターフェイス

MapForce Server は、インストール中に COM サーバーオブジェクトとして自動的に登録されます。登録が成功したか確認するには、レジストリエディターを開きます（例えば、コマンドラインに `regedit.exe` を入力します）。登録が成功している場合、レジストリは、クラス `MapForce.Server` を含んでいます。これらの 2 つのクラスは、通常の下で `HKEY_LOCAL_MACHINE\SOFTWARE` クラス検索することができます。

COM サーバーオブジェクトが登録されると、アプリケーション内から呼び出すことができ、呼び出しのためのプログラミングへのサポート持つ言語をスクリプトすることができます。MapForce Server インストール/パッケージのロケーションを変更するには、MapForce Server をアンインストール、必要なロケーションにインストールなおします。インストールプロセスにより必要な登録、および、登録の解除がこのようなことができます。

5.2.1 C++ サンプル

次のサンプルは C++ コードからのマッピング マッピング実行ファイル(.mfx) の実行方法を説明しています。Windows 上ではサンプルファイルは以下のパスで見つけることができます: `C:\Program Files\Altova\MapForceServer2021\etc\Examples`。

必要条件

下のコードを実行する前に以下の必要条件が満たされていることを確認してください！

- MapForce Server サーバーがインストールされておりライセンス供与済みであること。
- MapForce Server が COM サーバーオブジェクトとして使用できること（通常このプロセスは、自動的に MapForce Server のインストール中におこなわれます。インストールが成功したかをチェックするには、[COM インターフェイスについて](#)を参照してください）。

マッピングコードの実行

下のコードは 3 個のサーバー実行ファイル(.mfx) を実行します。下のテーブルは各 .mfx ファイルが期待する入力ファイルと、実行後に作成される出力をリストしています。

実行可能ファイル(.mfx)	入力	出力
TokenizeString.mfx	AltovaTools.xml	AltovaToolsFeatures.csv
SimpleTotal.mfx	ipo.xml	文字列
ClassifyTemperatures.mfx	Temperatures.xml	Temperatures_out.xml

Altova MapForce がインストールされている場合、更に詳しく理解するために .mfx ファイルがエンパイルされたオリジナルのマッピングを任意で確認することができます。これらはそれぞれ `TokenizeString1.mfd`、`SimpleTotal.mfd`、および `ClassifyTemperatures.mfd` と呼ばれます。respectively。以下のディレクトリ内でマッピングを見つけることができます: `C:\users\<user>\Altova\MapForce2021\MapForceExamples`。

下サンプルは以下を行います:

- `Altova.MapForceServer.Server` の新規のインスタンスを作成します。これは後の作業の目的となります。
- 実行が行われる作業ディレクトリを設定します。相対パスを使用して参照する場合、入力ファイルはこのディレクトリ内に存在することが期待されます。出力ファイルはこのディレクトリ内にも作成されます（上のテーブルも参照してください）。
- これは `TokenizeString.mfx` を実行します。ファイル `Run` メソッドに対して引数として提供されます（パスは以前に設定された作業ディレクトリに対して相対的であることを注意してください）。実行に成功すると、マッピング出力を表す .csv ファイルが

作業ディレクトリ内に作成されます。

- これは**SimpleTotal.mfx** を実行します。ファイルパスは作業ディレクトリに対して相対的です。このマッピングは文字列出力を生成するため**GetOutputParameter** メソッドを呼び出して文字列出力を取得します。
- これは**ClassifyTemperatures.mfx** を実行します。このマッピングは**AddParameter** メソッドを使用して提供される入力としてパラメータを期待します。

```
// 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\\MapForceServer2021\\bin\\
        \\MapForceServer.exe";
        // pMFS.ServerPath = "C:\\Program Files\\Altova\\MapForceServer2021\\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;
}

```

5.2.2 VBScript サンプル

次のサンプルはVBScript コードからのマッピングマッピング実行ファイル(.mfx) の実行方法を説明しています。Windows 上ではサンプルファイルは以下のパスで見つけることができます: C:\Program Files\Altova\MapForceServer2021\etc\Examples。

下のコードを実行する前に以下の必要条件が満たされていることを確認してください！

- MapForce Server サーバーがインストールされておりライセンス供与済みであること。
- MapForce Server がCOM サーバーオブジェクトとして使用できること(通常このプロセスは、自動的にMapForce Server のインストール中におこなわれます。インストールが成功したかをチェックするには [COM インターフェイスについて](#)を参照してください)。

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.")
        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\MapForceServer2021\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 direcorry
' 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 inpuz 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

```

5.2.3 VBA サンプル

Visual Basic for Applications (VBA) は、主に Microsoft Office 内のタスクの自動化のために使用されます。しかしながら、VBA から MapForce Server API を呼び出し、マッピングを実行することも可能です。次の命令は、MapForce Server と Microsoft Office 2013 のための VBA でテストされており、他の VBA 開発環境を使用すると命令が異なる場合があります。

必要条件

MapForce Server API 関数を VBA プロジェクトから呼び出す前に、次の必要条件を確認してください。

1. Microsoft Office と MapForce Server が同じマシンにインストールされている必要があります。
2. MapForce Server (32 ビット または 64 ビット) のアーキテクチャは、Microsoft Office に一致する必要があります。例えば、VBA を Microsoft Office 32 ビットで動作している場合、MapForce Server 32 ビットを使用してください。使用中の Office 製品が 64 ビットで動作しているかを確認するには、「ファイル」タブをクリックし、「アカウント」をクリックします。そして、「Excel にして」（または Word にして）をクリックします。
3. MapForce Server ライブラリは VBA プロジェクトから参照されている必要があります（下の命令を参照してください）。

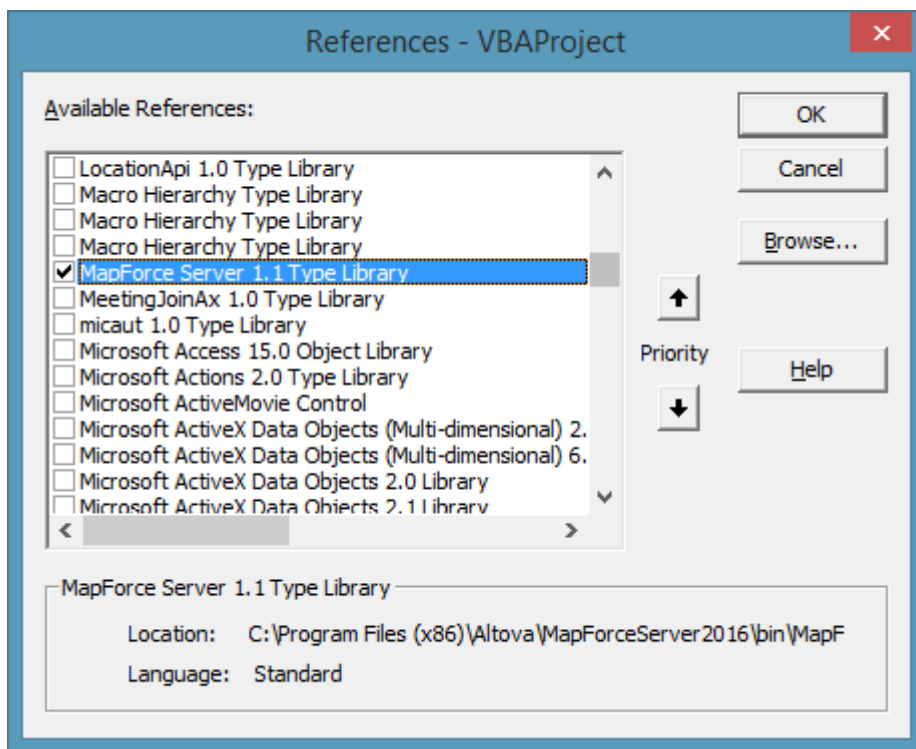
VBA プロジェクトから MapForce Server ライブラリ参照を追加する方法

1. マクロが有効化されている Microsoft Office ドキュメント (.docm、.xlsm) 内で、「Developer（開発者）」タブから「Visual Basic」をクリックします。

デフォルトでは、「Developer（開発者）」タブは Microsoft Office 内では有効化されていません。Office 2013 プログラム内で「Developer（開発者）」タブを有効化するには、リボンを右クリックして、「Customize the Ribbon

「(リボンをカスタマイズする)」をコンテキストメニューから選択します。オプション ダイアログボックスが表示されますので、「メインタブ」の下に「Developer (開発者)」チェックボックスをチェックします。

2. VBA 開発環境内の「ツール」メニューから「リファレンス」をクリックします。



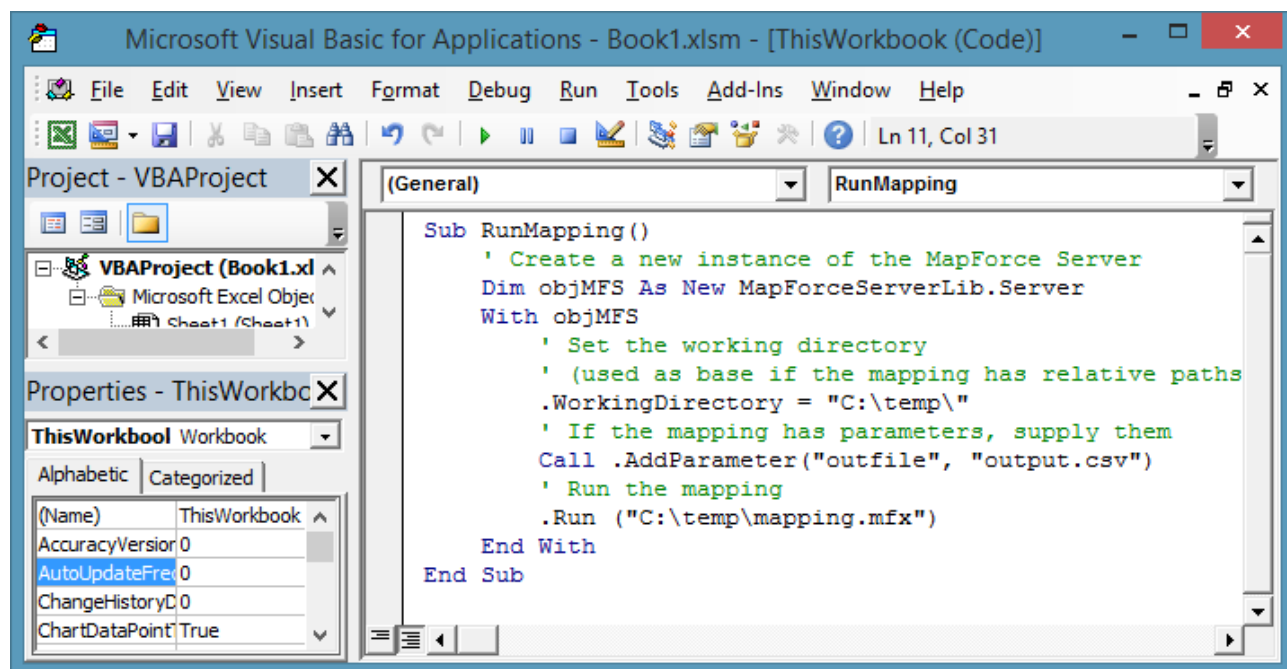
1. MapForce Server Type ライブラリを選択するために、クリックします。

MapForce Server API の呼び出し方

VBA プロジェクト内の MapForce Server ライブラリをリファレンスを追加すると、コード エディター ウィンドウに VBA コードを入力することができます。例えば、次のサンプルコードは、MapForce Server を呼び出し、値「output.csv」を持つ「outfile」と呼ばれる入力パラメータを取るマッピング実行可能ファイル(**mapping.mfx**)を実行します。

```
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
```

VBA コードをデバッグするために「F5」を押して、マッピングを実行します。



5.3 Java インターフェイス

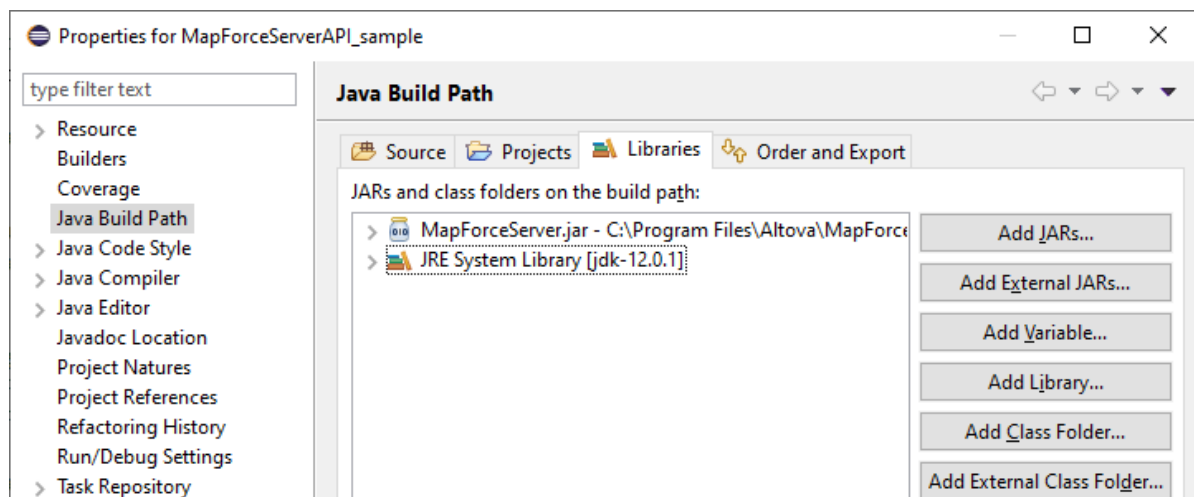
MapForce Server API から Java からアクセスするには、Java CLASSPATH に追加される必要がある

MapForceServer.jar ライブラリを参照します。You can find the **MapForceServer.jar** ファイルを MapForce Server インストールフォルダーの **bin** フォルダー内で見つけることができます。プロジェクトのセットアップに適する場合、元の場所、または他のローケーションにコピーして参照することができます。

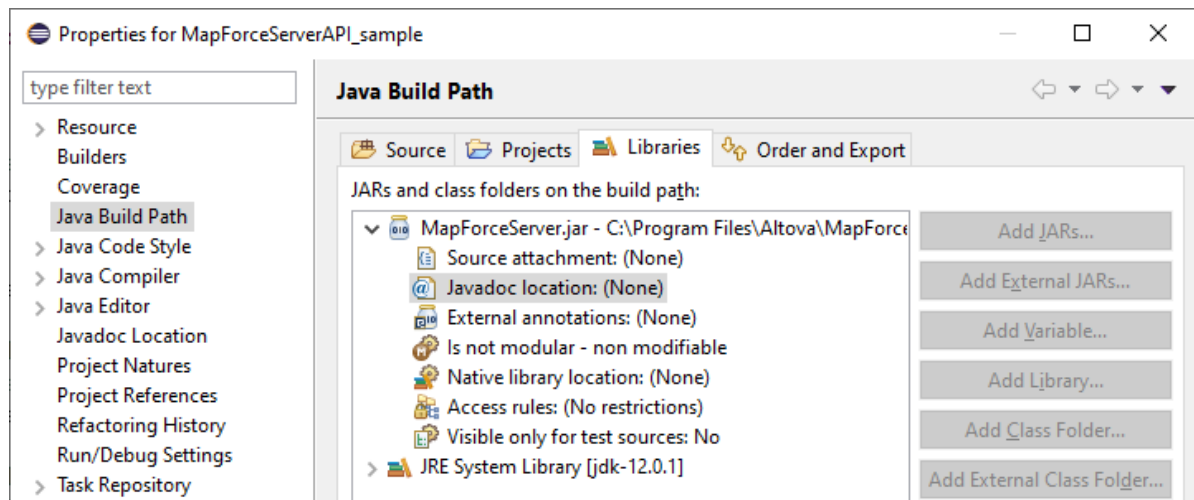
Java 開発環境として Eclipse を使用している場合、以下の通りプロジェクトのプロパティを編集することにより CLASSPATH に必要とされているライブラリを追加することができます。任意で、JavaDoc 書式のドキュメントを .jar ライブラリに添付することができます。MapForce Server インストールフォルダーの **bin** フォルダー内で JavaDoc を見つけることができます。JavaDoc ドキュメントを Eclipse 内で表示する方法を説明しています。

MapForce Server ライブラリ参照を追加する：

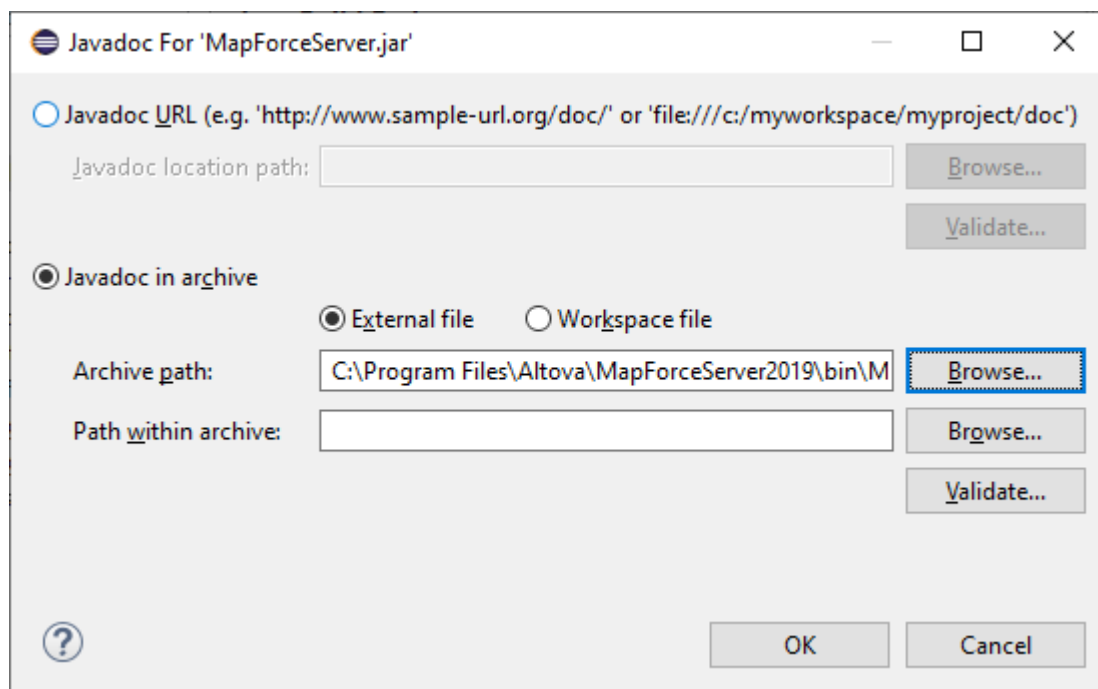
1. Eclipse 内でプロジェクトを開き、プロジェクト メニューから、プロパティをクリックします。
1. 「Java ビルドパス」をクリックします。
2. 「ライブラリ」タブから、「外部 JAR の追加」をクリックし、MapForce Server インストールフォルダー内にある **MapForceServer.jar** ファイルを参照します。



2. 任意で、JavaDoc アーカイブを追加するには、**MapForceServer.jar** レコードを展開し、**Javadoc location:** レコードをダブルクリックします。「ビルドパス上の JAR とクラスフォルダー」下の **MapForceServer.jar** を展開し、Javadoc ロケーション: (None) レコードをダブルクリックします。



3. 「アーカイブ内の Javadoc」と「外部ファイル」オプションが選択されていることを確認し、MapForce Server インストールフォルダーにある MapForceServer_JavaDoc.zip ファイルを参照してください。



4. 「OK」をクリックします。

64ビット Windows 上で作動する64ビット MapForce Server 上で、元のインストールフォルダーから参照する場合 Eclipse .classpath ファイルのサンプルは以下のようなになります(関連する行は黄色でハイライトされています):

```
<?xml version="1.0" encoding="UTF-8"?>
<classpath>
  <classpathentry kind="src" path="" />
  <classpathentry kind="lib" path="C:/Program
Files/Altova/MapForceServer2021/bin/MapForceServer.jar">
    <attributes>
```

```

<attribute name="javadoc_location" value="jar:file:/C:/Program%
20Files/Altova/MapForceServer2021/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>

```

5.3.1 Java サンプル

次のサンプルは Java コードからのマッピング マッピング実行ファイル(.mfx) の実行方法を説明しています。Windows 上では、すべてのサンプルファイルは以下のパスで見つけることができます: **C:\Program Files\Altova\MapForceServer2021\etc\Examples**。以下の通りです:

- Java プログラムにより実行される MapForce Server 実行可能ファイルである **TokenizeString.mfx**
- マッピングプログラムに入力データを提供する XML ファイルである **AltovaTools.xml**

Linux と Mac プラットフォームではサンプルはプレインストールされていません。しかし実行可能 .mfx ファイルをこのサンプルで以下のように準備することができます:

1. 希望するマッピングデザインファイル(.mfd) を Windows 上の MapForce で開きます。
2. 「ファイル」メニューでマッピングの設定を選択し、チェックされている場合「生成されたコード内でパスを絶対パスにする」チェックボックスをクリアします。
3. 各マッピングコンポーネントのためにプロパティダイアログボックスを開き、(例えば、コンポーネントのタイトルバーをダブルクリックして) 全てのファイルパスを絶対パスから相対パスに変更します。「MFD ファイルに対して全てのパスを相対的に保存する」チェックボックスを選択します。便宜上、マッピングと同様全ての入力ファイルとスキーマを同じファイルにコピーし、ファイル名で参照することができます。マッピングのデザイン中に相対と絶対パスと作業する方法に関しては MapForce ドキュメンテーションを参照してください。
4. 「ファイル」メニューから「MapForce Server 実行ファイルにコンパイル」を選択します。これは下のロードリストینگで表示されているとおり MapForce Server を使用して後に実行する .mfx ファイルを生成します。

必要条件

下のコードを実行する前に以下の必要条件が満たされていることを確認してください:

- MapForce Server サーバーがインストールされておりライセンス供与済みであること。
- Java CLASSPATH には **MapForceServer.jar** ライブラリの参照が含まれています (例えば [Java インターフェイススクリプト](#)を参照してください)。

カスタム .mfx ファイルを上記の通りに使用する場合、マッピングにより処理されるデータの種類により他の必要条件が存在する場合があります。詳細に関しては次を参照してください: [サーバー実行のためにマッピングを準備する](#)。

Windows 上では Java プロジェクトは管理者の特権を必要とする「プログラムファイル」ディレクトリ内に存在します。Java 開発環境 (例えば Eclipse) を管理者として実行、または管理者の権利を必要とすることなく異なるフォルダーにサンプルをコピーする必要があります。

Java プログラムの実行方法

下のコードリスティングは MapForce Server オブジェクトを最初に作成します。次にアプリケーションがマッピングへの入力としての役割を果たすファイルを検索する、マッピング出力ファイルが生成される個所に作業ディレクトリを設定します。上記のとおり、サンプルマッピングに入るソース XML ファイルからデータを読み取ります。ですから XML ファイルとそのスキーマが作業ディレクトリに存在することを確認してください。

`setServerPath` メソッドは MapForce Server 実行可能ファイルへのパスを設定します。例えば Ubuntu 上では `/opt/Altova/MapForceServer2021/bin/mapforceserver` です。MapForceServer.jar をデフォルトの場所から移動していない場合、`setServerPath` を省略することができます。

最後に `run` メソッドは MapForce によりコンパイルされたマッピング (.mfx ファイル) を実行します (このサンプルでは `TokenizeString.mfx`)。成功すると下のプログラムは CSV ファイルを作業ディレクトリに生成します。エラーが発生すると、プログラムは MapForce Server により生成された最後の実行メッセージを表示します。

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

            //Set a working directory - used as a base for relative paths (you may need to
            adapt the path to the installation folder)
            objMFS.setWorkingDirectory( "C:\\Program Files (x86)\\Altova\\
            \\MapForceServer2021\\etc\\Examples" );

            //Default path to the MapForce Server executable is the installation path
            (same dir with the MapForceServer.jar)
            //In case you copied the JAR file to a new location, you need to explicitly
            set the path to the .exe file
            objMFS.setServerPath( "C:\\Program Files (x86)\\Altova\\MapForceServer2021\\
            \\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", "Default" ); // "gc" can be
            used as short name for "globalresourceconfig"

            //Prepare the parameters, if your design uses parameters
            //objMFS.addParameter( "testparam1", "value 1" );

            //Run the mapping; input and output paths are stored inside the MFX file
            // NOTE Please adapt the path to the input file in order to run the sample
            if ( objMFS.run( "C:\\Program Files (x86)\\Altova\\MapForceServer2021\\etc\\
            \\Examples\\TokenizeString.mfx" ) )
                System.out.println( "Success - finished execution" );
            else
                System.out.println( objMFS.getLastExecutionMessage() );
        }
        catch ( Exception e )
        {
            e.printStackTrace();
        }
    }
}
```



```
    }  
    System.out.println( "Finished - exiting" );  
  }  
}
```

5.4 例: パラメーターを持つマッピングの実行

MapForce Server 実行ファイル(.mfx) に MapForce マッピングをエンパイルし、MapForce API から実行する方法を説明しています。サンプルは特にマッピングが入力ファイル名をパラメーターとしてとるシナリオについて説明しています。パラメーターを持たないマッピングの実行を説明するサンプルに関しては [C#](#)、[VB.NET](#)、[C++](#)、[VBScript](#)、または [Java](#) サンプルを参照してください。

必要条件

- このサンプル内では元のマッピングデザインをビューして理解するために MapForce が使用されています。MapForce は MapForce Server 実行ファイル(.mfx) マッピングをエンパイル、相対的/絶対的パスなどの設定を構成するためにも使用されています。
- このサンプルで使用されているサーバープラットフォームは Windows です。MapForce がインストールされている同じマシンまたは異なるマシンで行うことができます。Windows スタイルのパスをプラットフォームに適用できるように調整することにより Linux または Mac マシン (Java) 上でこのサンプルを実行することもできます。

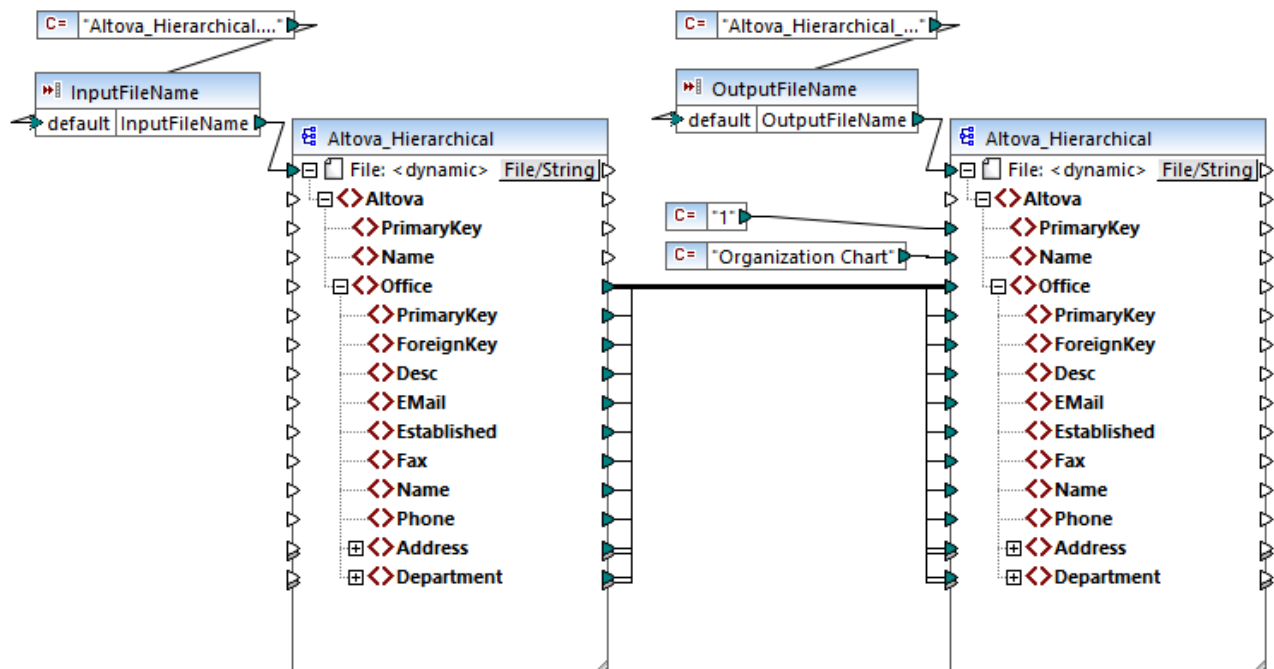
サーバー実行のためのマッピングの準備

MapForce をインストールして MapForce を最初に起動する際に以下のパスでいくつかのデモマッピングデザインを使用することができます:

C:\Users\%username%\Documents\Altova\MapForce2021\MapForce サンプル

異なるバージョンの MapForce を使用する場合上記のパスを必要に応じて変更してください。

このサンプル内で使用されているスキーマは **FileNamesAsParameters.mfd** と呼ばれます。以下で説明されている通り、ターゲット XML ファイルヘッスからデータをほぼ変更することなくコピーするこればかりやしいマッピングです。ターゲット XML ファイルの **PrimaryKey** と **Name** フィールドがマッピングからの定数により作成されます。



FileNamesAsParameters.mfd

マッピングは2つの入力パラメーターが存在します: **InputFileName** and **OutputFileName**, respectively. MapForce 内のマッピングをレビュー可能にするには、パラメーターの値は2つの定数により提供されます。以下で説明されている通り、マッピングが MapForce Server で実行される場合、パラメーターの値を自身の値と置き換えることができます。

ソースとターゲットマッピングコンポーネントの両方はマッピングからファイル名を動的に取得するために構成されています。具体的には *File*: **<Dynamic>** 入力接続はコンポーネントの一番上で使用することができます。対応するマッピングパラメーター (**InputFileName** または **OutputFileName**) からデータを読み取ることができます。MapForce 内でコンポーネントの右上の「ファイル/文字列」ボタンをクリックすることでこのオプションを設定または変更することができます。パラメーターとして提供された入力ファイル名はマッピングに入力データを提供する物理的な XML ファイルへのリファレンスである必要があります (このサンプルではマッピングデザインファイルと同じフォルダーからの **Altova_Hierarchical.xml** です)。出力ファイル名はすべての有効なファイルであることができます。例えば **Output.xml**。

MapForce Server 実行ファイル (.mfx) にマッピングをコンパイルする前に、通常マッピングの設定をレビューすることが奨励されます。マッピングの空白の部分をクリックし、コンテキストメニューから「マッピングの設定」を選択します。

この例のスキープのために、上記の通り設定を変更します。特に「生成されたコード内でパスを絶対パスにする」チェックボックスをクリアすると、マッピングにより使用されている入力または出力ファイルへの絶対パスはマッピングデザインファイル (.mfd) に相対的に解決されます。マッピングのランタイムでは MapForce Server はプログラムの作業ディレクトリ内でこれらのパスを検索します。デフォルトの作業ディレクトリは MapForce Server API にアクセスするために使用されたプラットフォームにより異なります。例えば、C# アプリケーションの場合、実行可能ファイルと同じディレクトリです。更に以下で説明されているとおり、[WorkingDirectory](#) (または適用可能な場合は [setWorkingDirectory](#) Java メソッド) という名前の API プロパティを使用して作業ディレクトリを変更することができます。

メモ マッピング設定ダイアログボックスで、.mfx ファイルのコンパイルを直接影響する唯一の設定は「生成されたコード内でパスを絶対パスにする」と「行末」です。

この例は、作業ディレクトリとして **C:\MappingExample** を使用しています。このためマッピングにより参照されている入力ファイル **Altova_Hierarchical.xml** を **C:\Users\<username>\Documents\Altova\MapForce2021\MapForceExamples** から作業ディレクトリにコピーします。

このサンプルでは、ソースとターゲットは XML ファイルで XML スキーマファイルをサーバーにコピーする必要はありません。これは派生した情報は .mfx ファイルにコンパイル中に埋め込まれるからです。マッピングが他のコンポーネント型を使用すると (例えば、データベース) 追加の必要条件が存在する場合があります。[サーバーでの実行のためにマッピングを準備する](#) も参照してください。

最後にマッピングをサーバー実行ファイルにコンパイルするには以下を行います:

- 「ファイル」メニューから「MapForce Server 実行ファイルコンパイル」をクリックしターゲットディレクトリを選択します。このサンプルではターゲットディレクトリは作業ディレクトリ **C:\MappingExample** と同じです。

次のコードリストは MapForce Server API を使用して異なる環境からマッピングを実行する方法を説明しています。下のコードリストでは、両方のパラメーターはマッピングデザイン内の同じ名前により参照されています。また、パラメーターの値はマッピング上で定義されている同じデータ型を持ちます(この場合 `string`)。

C++ からマッピングを実行する

.mfx ファイルを以下に示される C++ コードを使用して実行します。このコードリストの実行に成功するには MapForce Server が有効なライセンスを有していることを確認してください。[verifylicense](#) CLI コマンドを参照してください。

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;
}
```

C# からマッピングを実行する

.mfx ファイルを以下に示される Java コードを使用して実行します。このコードリスティングの実行に成功するには Visual Studio 内の MapForce Server DLL にリファレンスを [NET インターフェイス](#) で説明されるように追加します。MapForce Server が有効なライセンスを有していることを確認してください。 [verifylicense](#) CLI コマンドも参照してください。

C#

```
static void Main(string[] args)
{
    // 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);
}
```

Java からマッピングを実行する

.mfx ファイルを以下に示される Java コードを使用して実行します。このコードリスティングの実行に成功するには以下を確認してください。

- [Java インターフェイスについて](#)内で説明されているとおりの **MapForceServer.jar** ライブラリへのリファレンスを含む Java CLASSPATH 環境変数
- MapForce Server が有効なライセンスを有していること。 [verifylicense](#) CLI コマンドを確認してください。

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("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");  
}
```

5.5 API レファレンス (COM、.NET)

このセクションはCOM または.NET プラットフォームのために適用される MapForce Server API 要素に対する一般的なレファレンスクラスとメソッドについて説明します。

5.5.1 インターフェイス

5.5.1.1 IServer

IServer インターフェイスは、新規の MapForce サーバーオブジェクト インスタンスを作成し、MapForce Server エンジンへのアクセスを与えます。

COM プラットフォームで C++ を使用している場合、メインインターフェイスの名前は IServer です。COM プラットフォームで C# または VB.NET、¥ を使用している場合、メインインターフェイスの名前は IServer です。

プロパティ

名前	説明
APIMajorVersion	読み取り専用。 MapForce Server API のメジャーバージョンを取得します。API が他のサーバーに接続されている場合、製品バージョンとは異なる場合があります。
APIMinorVersion	読み取り専用。 MapForce Server API のマイナーバージョンを取得します。API が他のサーバーに接続されている場合、製品バージョンとは異なる場合があります。
APIServicePackVersion	読み取り専用。 MapForce Server API のサービスパックバージョンを取得します。API が他のサーバーに接続されている場合、製品バージョンとは異なる場合があります。
Is64Bit	読み取り専用。 MapForce Server エンジンが 64 ビットの実行可能ファイルの場合、true を返します。
LastExecutionMessage	読み取り専用。 最後の Run コマンド中に受け取ったメッセージを取得します。
MajorVersion	読み取り専用。 1998 から始まる製品のメジャーバージョンを取得します (例えば Altova MapForce Server 2018 のために "20" など)。
MinorVersion	読み取り専用。 製品のマイナーバージョンを取得します (例えば Altova MapForce Server 2018 r2 のために "2" など)。
ProductName	読み取り専用。 製品の名前を取得します (例えば Altova MapForce Server など)。
ProductNameAndVersion	読み取り専用。

名前	説明
	製品の完全な名前とバージョンを取得します (例えば "Altova MapForce Server 2018 r2 sp1 (x64)")。
ServerPath	MapForce Server 実行可能ファイルへのパスを取得または設定します。
ServicePackVersion	読み取り専用。 製品のサービスパックのバージョンを取得します (例えば Altova MapForce Server 2018 r2 sp1 (x64) のために "1" など)。
WorkingDirectory	実行中のジョブのための現在のディレクトリを取得、または設定します (相対的パスは、作業ディレクトリに対して評価されます)。

メソッド

名前	説明
AddCredentialProperty	現在の証明書にプロパティを追加する (例えば ユーザー名、パスワード、または両方)。最初の引数は、プロパティの名前を指定し、2番目の引数はプロパティの値を指定します。有効なプロパティ名: username, password. MapForce Server Advanced Edition ではプロパティ名 oauth:token を追加でサポートしています。 このメソッドは BeginCredential() を呼び出す後、および EndCredential() を呼び出す前に呼び出される必要があります。
AddParameter	マッピング内で定義されているパラメーターに値を割り当てます。最初の引数は、マッピング上で定義されたパラメーター名を指定します。2番目の引数はパラメーターの値を指定します。
BeginCredential	引数として与えられている新規の資格情報を作成します。このメソッドを呼び出すと、AddCredentialProperty() を使用してプロパティを追加する必要があり、EndCredential() を呼び出して最後に資格情報を閉じる必要があります。
ClearCredentialList	前に設定された資格情報のリストをクリアする。オブジェクトのライフタイムのためにすべての資格情報は有効です。前に設定された資格情報を明示的にクリアする必要がある場合、このメソッドを呼び出します。
ClearOptions	SetOption メソッドにより設定されたオプションのリストをクリアします。SetOption メソッドにより設定されたすべてのオプションはオブジェクトのライフタイムの期間有効です。前に設定されたオプションを明示的にクリアする必要がある場合、このメソッドを呼び出します。
ClearParameterList	AddParameter メソッドを使用して設定されたパラメーターのリストをクリアします。
EndCredential	BeginCredential メソッドを使用して以前に作成された資格情報オブジェクトを閉じます。
GetOutputParameter	最後の run コマンドから生成された文字列の出力を取得します。出力が生成されない場合 null を返します。この関数は、MapForce 内の出力コンポーネントの名前を識別する文字列、パラメーターを必要とします。

名前	説明
Run	MapForce Server 実行ファイル(.mfx ファイル)を実行します。成功の場合は true を返し、それ以外の場合、 false を返します。
SetOption	マッピングを実行する前のオプションを設定します。最初の引数は、オプションの名前を指定し、2番目の引数はオプションの値を指定します。このメソッドはマッピングがグローバルリソースを消費するためにデザインされている場合とても役に立ちます (Altova グローバルリソースを参照してください)。現在サポートされているオプションは以下の通りです: • globalresourcefile (or gr) - グローバルリソースファイル (このオプションが指定されている場合、グローバルリソース構成名も指定されている必要があります。次のアイテムを参照してください)。 • globalresourcefile (or gr) - グローバルリソース構成名 (このオプションが指定されている場合、グローバルリソースファイルも指定されている必要があります。次のアイテムを参照してください)。 • catalog - カスタム RootCatalog.xml ファイルへのパス (このオプションによりマッピングにより使用された URL を解決するために使用されたカスタムカタログファイルを指定します。デフォルトのカタログはプログラムインストールディレクトリのサブディレクトリ内にあります)。 • taxonomy-package - マッピングにより必要とされている場合、カスタム XBRL タクソノミパッケージへのパスです。 • taxonomy-package - マッピングにより必要とされている場合、カスタム XBRL タクソノミパッケージ構成へのパスです。 設定されたすべてのオプションはオブジェクトのライフタイムの期間有効です。同じ名前オプションを2回設定すると、前のオプションは上書きされます。明示的にすべてのオプションをクリアするために <code>ClearOptions()</code> メソッドを呼び出します。
StopServerProcess	COM オブジェクトに接続されている場合オブジェクトをリリースすることなくプロセスを明示的に停止します。COM オブジェクトがリリースされるとプロセスは暗示的に停止します。

サンプル

MapForce Server を異なる言語で開始し、実行する方法を次のサンプルは説明しています:

- [C++ サンプル](#)
- [C# サンプル](#)
- [VBScript サンプル](#)
- [VB.NET サンプル](#)

5.5.1.1.1 プロパティ

5.5.1.1.1.1 *APIMajorVersion*

MapForce Server API のメジャーバージョンを取得します。API が他のサーバーに接続されている場合、製品バージョンとは異なる場合があります。

署名

```
APIMajorVersion : Integer
```

ジェネリックな署名

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

C#

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

C++

```
ReadOnly Property APIMajorVersion As Integer
```

VB.NET

5.5.1.1.2 *APIMinorVersion*

MapForce Server API のマイナーバージョンを取得します。API が他のサーバーに接続されている場合、製品バージョンとは異なる場合があります。

署名

```
APIMinorVersion : Integer
```

ジェネリックな署名

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

C#

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

C++

```
ReadOnly Property APIMinorVersion As Integer
```

VB.NET

5.5.1.1.3 *APIServicePackVersion*

MapForce Server API のサービスパックバージョンを取得します。API が他のサーバーに接続されている場合、製品バージョンとは異なる場合があります。

署名

```
APIServicePackVersion : Integer
```

ジェネリックな署名

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

C#

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

C++

```
ReadOnly Property APIServicePackVersion As Integer
```

VB.NET

5.5.1.1.4 Is64Bit

MapForce Server エンジンが64 ビットの実行可能ファイルの場合、**true** を返します。

署名

```
Is64Bit : Boolean
```

ジェネリックな署名

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

C#

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

C++

```
ReadOnly Property Is64Bit As Boolean
```

VB.NET

5.5.1.1.1.5 *LastExecutionMessage*

最後のRunコマンド中に受け取りメッセージを取得します。

署名

```
LastExecutionMessage : String
```

ジェネリックな署名

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

C#

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

C++

```
ReadOnly Property LastExecutionMessage As String
```

VB.NET

5.5.1.1.1.6 *MajorVersion*

1998 から始まる製品のメジャーバージョンを取得します (例えば Altova MapForce Server 2018 のために "20" など)。

署名

```
MajorVersion : Integer
```

ジェネリックな署名

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

C#

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

C++

```
ReadOnly Property MajorVersion As Integer
```

VB.NET

5.5.1.1.7 *MinorVersion*

製品のマイナーバージョンを取得します (例えば Altova MapForce Server 2018 r2 のために "2" など)。

署名

```
MinorVersion : Integer
```

ジェネリックな署名

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

C#

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

C++

```
ReadOnly Property MinorVersion As Integer
```

VB.NET

5.5.1.1.8 *ProductName*

製品の名前を取得します (例えば Altova MapForce Server など)。

署名

```
ProductName : String
```

ジェネリックな署名

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

C#

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

C++

```
ReadOnly Property ProductName As String
```

VB.NET

5.5.1.1.1.9 ProductNameAndVersion

製品の完全な名前とバージョンを取得します (例えば "Altova MapForce Server 2018 r2 sp1 (x64)")。

署名

```
ProductNameAndVersion : String
```

ジェネリックな署名

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

C#

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

C++

```
ReadOnly Property ProductNameAndVersion As String
```

VB.NET

5.5.1.1.1.10 ServerPath

MapForce Server 実行可能ファイルへのパスを取得または設定します。

署名

```
ServerPath : String
```

ジェネリックな署名

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

C#

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

C++

```
Property ServerPath As String
```

VB.NET

5.5.1.1.11 ServicePackVersion

製品のサービスパックのバージョンを取得します (例えば, Altova MapForce Server 2018 r2 sp1 (x64) のために "1" など)。

署名

```
ServicePackVersion : Integer
```

ジェネリックな署名

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

C#

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

C++

```
ReadOnly Property ServicePackVersion As Integer
```

VB.NET

5.5.1.1.12 WorkingDirectory

実行中のジョブのための現在のディレクトリを取得、または、設定します (相対的パスは、作業ディレクトリに対して評価されます)。

署名

```
WorkingDirectory : String
```

ジェネリックな署名

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

C#

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

C++

```
Property WorkingDirectory As String
```

VB.NET

5.5.1.1.2 メソッド

5.5.1.1.2.1 AddCredentialProperty

現在の証明書プロパティを追加する(例えば、ユーザー名、パスワード、または両方)。最初の引数は、プロパティの名前を指定し、2番目の引数はプロパティの値を指定します。有効なプロパティ名: username, password. MapForce Server Advanced Edition ではプロパティ名 oauth:token を追加でサポートしています。

このメソッドは BeginCredential() を呼び出す後、および EndCredential() を呼び出す前に呼び出される必要があります。

署名

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

ジェネリックな署名

```
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

パラメーター

名前	型	説明
bstrName	System.String	資格情報プロパティ名を指定します。
bstrValue	System.String	設定する資格情報の値を指定します。

サンプル

次のコードリストはC# の"mycredential" という名前の資格情報を宣言する方法を表しています。資格情報名はデザイン時に MapForce 内で資格情報に与えられているものである必要があります。

```
//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();
```


5.5.1.1.2.2 AddParameter

マッピング内で定義されているパラメータに値を割り当てます。最初の引数は、マッピング上で定義されたおりにパラメータ名を指定します。2番目の引数はパラメータの値を指定します。

署名

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

ジェネリックな署名

```
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

パラメーター

名前	型	説明
bstrName	String	パラメータ名の指定
bstrValue	String	パラメータ値の指定

5.5.1.1.2.3 BeginCredential

引数として与えられている新規の資格情報を作成します。このメソッドを呼び出すと、AddCredentialProperty() を使用してプロパティを追加する必要があり、EndCredential() を呼び出して最後に資格情報を閉じる必要があります。

署名

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

ジェネリックな署名

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

C#

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

C++

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

VB.NET

パラメーター

名前	型	説明
bstrCredentialName	String	MapForce 内で表示される通り資格情報の名前を指定します。

5.5.1.1.2.4 ClearCredentialList

前に設定された資格情報のリストをクリアする。オブジェクトのライフタイムのためにすべての資格情報は有効です。前に設定された資格情報を明示的にクリアする必要がある場合、このメソッドを呼び出します。

署名

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

ジェネリックな署名

```
void ClearCredentialList()
```

C#

```
HRESULT ClearCredentialList();
```

C++

```
Sub ClearCredentialList()
```

VB.NET

5.5.1.1.2.5 ClearOptions

SetOption メソッドにより設定されたオプションのリストをクリアします。SetOption メソッドにより設定されたすべてのオプションはオブジェクトのライフタイムの期間有効です。前に設定されたオプションを明示的にクリアする必要がある場合、このメソッドを呼び出します。

署名

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

ジェネリックな署名

```
void ClearOptions()
```

C#

```
HRESULT ClearOptions();
```

C++

```
Sub ClearOptions()
```

VB.NET

5.5.1.1.2.6 *ClearParameterList*

AddParameter メソッドを使用して設定されたパラメーターのリストをクリアします。

署名

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

ジェネリックな署名

```
void ClearParameterList()
```

C#

```
HRESULT ClearParameterList();
```

C++

```
Sub ClearParameterList()
```

VB.NET

5.5.1.1.2.7 EndCredential

BeginCredential メソッドを使用して以前に作成された資格情報オブジェクトを閉じます。

署名

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

ジェネリックな署名

```
void EndCredential()
```

C#

```
HRESULT EndCredential();
```

C++

```
Sub EndCredential()
```

VB.NET

5.5.1.1.2.8 GetOutputParameter

最後の run コマンドから生成された文字列の出力を取得します。出力が生成されない場合 null を返します。この関数は、MapForce 内の出力コンポーネントの名前を識別する文字列パラメータを必要とします。

署名

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

ジェネリックな署名

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

C#

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

C++

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

VB.NET

パラメーター

名前	型	説明
bstrName	String	MapForce 内で表示される通り出力コンポーネントの名前を指定します。この名前はマッピングの各コンポーネントのタイトルバー内に表示されます(または、コンポーネントヘッダーを右クリックして、プロパティを選択します)。

5.5.1.1.2.9 Run

MapForce Server 実行ファイル(.mfx ファイル)を実行します。成功の場合は **true**を返し、それ以外の場合、**false** を返します。

署名

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

ジェネリックな署名

```
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

パラメーター

名前	型	説明
bstrMappingPath	String	MapForce Server execution (.mfx) ファイルへのパスを指定します。相対的なパスを指定すると、作業ディレクトリに対して解決されます。WorkingDirectory プロパティから作業ディレクトリを設定することができます。

5.5.1.1.2.11 *StopServerProcess*

このメソッドは 明示的に COM オブジェクトに接続されるプロセスをオブジェクトをリリースすることなく停止します。プロセスは COM オブジェクトがリリースされると明示的に停止されます。

署名

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

ジェネリックな署名

```
void StopServerProcess()
```

C#

```
Sub StopServerProcess()
```

VB.NET

5.6 API レファレンス (Java)

このセクションはJava プラットフォームのために適用されるMapForce Server API 要素に対する一般的なレファレンス(クラスとメソッド)について説明します。

5.6.1 クラス

5.6.1.1 MapForceServer

`MapForceServer` クラスは、新規のMapForce サーバーオブジェクト インスタンスを作成し、MapForce Server エンジンへのアクセスを与えます。

メソッド

名前	説明
<code>addCredentialPropertiesFromMap</code>	資格情報プロパティマップから現在の資格情報プロパティを追加します。資格情報プロパティマップ(<code>property_name</code> , <code>property_value</code>)を引数として取りま す。このメソッドは <code>beginCredential()</code> を呼び出す後、および <code>endCredential()</code> を呼び出す前に呼び出される必要があります。メソ ッドを呼び出す代わりに、 <code>AddCredentialProperty()</code> を呼び出すこともで きます。
<code>addCredentialProperty</code>	現在の証明書プロパティを追加する(例えば、ユーザー名、パスワード、または 両方)。最初の引数は、プロパティの名前を指定し、2番目の引数はプロパ ティの値を指定します。有効なプロパティ名: <code>username</code> , <code>password</code> 。MapForce Server Advanced Edition ではプロパティ名 <code>oauth:token</code> を追加でサポ ートしています。 このメソッドは <code>beginCredential()</code> を呼び出す後、および <code>endCredential()</code> を呼び出す前に呼び出される必要があります。このメ ソッドを呼び出す代わりに <code>addCredentialPropertiesFromMap()</code> を 呼び出すこともできます。
<code>addParameter</code>	マッピング内で定義されているパラメーターに値を割り当てます。
<code>beginCredential</code>	引数として与えられている新規の資格情報を作成します。このメソッドを呼び出す 場合、 <code>addCredentialProperty()</code> or <code>addCredentialPropertiesFromMap()</code> を使用してプロパティを追加す る必要があります。最後に <code>endCredential()</code> を呼び出して資格情報を閉じま す。
<code>clearCredentialList</code>	前に設定された資格情報のリストをクリアする。オブジェクトのライフタイムのため すべての資格情報は有効です。前に設定された資格情報を明示的にクリアする 必要がある場合、このメソッドを呼び出します。
<code>clearOptions</code>	<code>setOption()</code> メソッドにより設定されたオプションのリストをクリアします。 <code>setOption</code> メソッドにより設定されたすべてのオプションはオブジェクトのライフ タイムの期間有効です。前に設定されたオプションを明示的にクリアする必要 がある場合、このメソッドを呼び出します。

名前	説明
clearParameterList	<code>addParameter</code> メソッドを使用して設定されたパラメータのリストをクリアします。
endCredential	<code>beginCredential</code> メソッドを使用して以前に作成された資格情報オブジェクトを閉じます。
getAPIMajorVersion	MapForce Server API のメジャーバージョンを取得します。API が他のサーバーに接続されている場合、製品バージョンとは異なる場合があります。
getAPIMinorVersion	MapForce Server API のマイナーバージョンを取得します。API が他のサーバーに接続されている場合、製品バージョンとは異なる場合があります。
getAPIServicePackVersion	MapForce Server API のサービスパックバージョンを取得します。API が他のサーバーに接続されている場合、製品バージョンとは異なる場合があります。
getLastExecutionMessage	最後の <code>run</code> コマンド中に受け取ったメッセージを取得します。
getMajorVersion	1998 から始まる製品のメジャーバージョンを取得します (例えば Altova MapForce Server 2018 のために "20" など)。
getMinorVersion	製品のマイナーバージョンを取得します (例えば Altova MapForce Server 2018 r2 のために "2" など)。
getOutputParameter	最後の <code>run</code> コマンドから生成された文字列の出力を取得します。出力が生成されない場合 <code>null</code> を返します。この関数は、MapForce 内の出力コンポーネントの名前を識別する文字列パラメータを必要とします。
getProductName	製品の名前を取得します (例えば Altova MapForce Server など)。
getProductNameAndVersion	製品の完全な名前とバージョンを取得します (例えば "Altova MapForce Server 2018 r2 sp1 (x64)")。
getServicePackVersion	製品のサービスパックのバージョンを取得します (例えば Altova MapForce Server 2018 r2 sp1 (x64) のために "1" など)。
is64bit	MapForce Server エンジンが 64 ビットの実行可能ファイルの場合、 <code>true</code> を返します。
run	MapForce Server 実行ファイル (.mfx ファイル) を実行します。成功の場合は <code>true</code> を返し、それ以外の場合、 <code>false</code> を返します。
setOption	マッピングを実行する前のオプションを設定します。最初の引数は、オプションの名前を指定し、2 番目の引数はオプションの値を指定します。このメソッドはマッピングがグローバルリソースを消費するためデザインされている場合とても役に立ちます (Altova グローバルリソース を参照してください)。現在サポートされているオプションは以下の通りです: globalresourcefile (or gr) - グローバルリソースファイル (このオプションが指定されている場合、グローバルリソース構成名も指定されている必要があります。次のアイテムを参照してください)。 globalresourcefile (or gr) - グローバルリソース構成名 (このオプションが指定されている場合、グローバルリソースファイルも指定されている必要があります。次のアイテムを参照してください)。

名前	説明
	• catalog – カスタム RootCatalog.xml ファイルへのパスのオプションによりマッピングにより使用された URL を解決するために使用されるカスタムカタログファイルを指定します。デフォルトのカタログはプログラムインストールディレクトリのサブディレクトリにあります。 • taxonomy-package – マッピングにより必要とされている場合、カスタム XBRL タクノミパッケージへのパスです。 • taxonomy-package – マッピングにより必要とされている場合、カスタム XBRL タクノミパッケージ構成へのパスです。 設定されたすべてのオプションはオブジェクトのライフタイムの期間有効です。同じ名前でオプションを2回設定すると、前のオプションは上書きされます。明示的にすべてのオプションをクリアするために <code>clearOptions()</code> メソッドを呼び出します。
setServerPath	MapForce Server 実行可能ファイルへのパスを設定します
setWorkingDirectory	実行中のジョブのために現在のディレクトリを設定します (相対的パスは、作業ディレクトリに対して評価されます)。

サンプル

Java 内の MapForceServer の新規のインスタンスの作成のサンプルに関しては、[Java サンプル](#)を参照してください。

5.6.1.1.1 メソッド

5.6.1.1.1.1 *addCredentialPropertiesFromMap*

資格情報プロパティマップから現在の資格情報プロパティを追加します。資格情報プロパティマップ (`property_name`, `property_value`) を引数として取ります。このメソッドは `beginCredential()` を呼び出す後、および `endCredential()` を呼び出す前に呼び出される必要があります。メソッドを呼び出す代わりに `AddCredentialProperty()` を呼び出すこともできます。

署名

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

ジェネリックな署名

パラメーター

名前	型	説明
arg0	Map	資格情報プロパティの名前とその値を提供するマップ

サンプル

次のコードリストは `addCredentialPropertiesFromMap` メソッドを使用して "mycredential" と呼ばれる資格情報を現在のコンテキストに追加することを示しています。

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

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();
```

5.6.1.1.1.2 addCredentialProperty

現在の証明書にプロパティを追加する(例えば、ユーザー名、パスワード、または両方)。最初の引数は、プロパティの名前を指定し、2番目の引数はプロパティの値を指定します。有効なプロパティ名: username, password. MapForce Server Advanced Edition ではプロパティ名 oauth:token を追加でサポートしています。

このメソッドは `beginCredential()` を呼び出す後、および `endCredential()` を呼び出す前に呼び出される必要があります。このメソッドを呼び出す代わりに `addCredentialPropertiesFromMap()` を呼び出すこともできます。

署名

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

ジェネリックな署名

パラメーター

名前	型	説明
arg0	String	資格情報プロパティの名前(例えば "username" または "password")。
arg1	String	資格情報プロパティの値

サンプル

次のコードリストは "mycredential" という名前の資格情報を宣言する方法を表しています。資格情報名はデザイン時に MapForce 内で資格情報に与えられているものである必要があります。

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

objMFS.beginCredential("mycredential");
```

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

5.6.1.1.1.3 addParameter

マッピング内で定義されているパラメータに値を割り当てます。

署名

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

ジェネリックな署名

パラメーター

名前	型	説明
arg0	String	パラメータ名の指定
arg1	String	パラメータ値の指定

5.6.1.1.1.4 beginCredential

引数として与えられている新規の資格情報を作成します。このメソッドを呼び出す場合、addCredentialProperty() or addCredentialPropertiesFromMap() を使用してプロパティを追加する必要があり、最後にendCredential() を呼び出して資格情報を閉じます。

署名

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

ジェネリックな署名

パラメーター

名前	型	説明
arg0	String	MapForce 内で表示される通り資格情報の名前

5.6.1.1.1.5 clearCredentialList

前に設定された資格情報のリストをクリアする。オブジェクトのライフタイムのためにすべての資格情報は有効です。前に設定された資格情報を明示的にクリアする必要がある場合、このメソッドを呼び出します。

署名

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

ジェネリックな署名

5.6.1.1.1.6 *clearOptions*

`setOption()` メソッドにより設定されたオプションのリストをクリアします。`setOption` メソッドにより設定されたすべてのオプションはオブジェクトのライフタイムの期間有効です。前に設定されたオプションを明示的にクリアする必要がある場合、このメソッドを呼び出します。

署名

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

ジェネリックな署名

5.6.1.1.1.7 *clearParameterList*

`addParameter` メソッドを使用して設定されたパラメーターのリストをクリアします。

署名

```
clearParameterList() -> void
```

ジェネリックな署名

5.6.1.1.1.8 *endCredential*

`beginCredential` メソッドを使用して以前に作成された資格情報オブジェクトを閉じます。

署名

```
endCredential() -> void
```

ジェネリックな署名

5.6.1.1.1.9 *getAPIMajorVersion*

MapForce Server API のメジャーバージョンを取得します。API が他のサーバーに接続されている場合、製品バージョンとは異なる場合があります。

署名

```
getAPIMajorVersion() -> int
```

ジェネリックな署名

5.6.1.1.10 *getAPIMinorVersion*

MapForce Server API のマイナーバージョンを取得します。API が他のサーバーに接続されている場合、製品バージョンとは異なる場合があります。

署名

```
getAPIMinorVersion() -> int
```

ジェネリックな署名

5.6.1.1.11 *getAPIServicePackVersion*

MapForce Server API のサービスパックバージョンを取得します。API が他のサーバーに接続されている場合、製品バージョンとは異なる場合があります。

署名

```
getAPIServicePackVersion() -> int
```

ジェネリックな署名

5.6.1.1.12 *getLastExecutionMessage*

最後の run コマンド中に受け取ったメッセージを取得します。

署名

```
getLastExecutionMessage() -> java.lang.String
```

ジェネリックな署名

5.6.1.1.13 *getMajorVersion*

1998 から始まる製品のメジャーバージョンを取得します (例えば, Altova MapForce Server 2018 のため "20" など)。

署名

```
getMajorVersion() -> int
```

ジェネリックな署名

5.6.1.1.14 *getMinorVersion*

製品のマイナーバージョンを取得します (例えば, Altova MapForce Server 2018 r2 のため "2" など)。

署名

```
getMinorVersion() -> int
```

ジェネリックな署名

5.6.1.1.15 *getOutputParameter*

最後の run コマンドから生成された文字列の出力を取得します。出力が生成されない場合 null を返します。この関数は、MapForce 内の出力コンポーネントの名前を識別する文字列/パラメータを必要とします。

署名

```
getOutputParameter(arg0:String) -> java.lang.String
```

ジェネリックな署名

パラメーター

名前	型	説明
arg0	String	MapForce 内で表示される通り出力コンポーネントの名前を指定します。この名前はマッピングの各コンポーネントのタイトルバー内に表示されます (または、コンポーネントヘッダーを右クリックして、プロパティを選択します)。

5.6.1.1.16 *getProductName*

製品の名前を取得します (例えば, Altova MapForce Server など)。

署名

```
getProductName() -> java.lang.String
```

ジェネリックな署名

5.6.1.1.17 *getProductNameAndVersion*

製品の完全な名前とバージョンを取得します (例えば, "Altova MapForce Server 2018 r2 sp1 (x64)")。

署名

```
getProductNameAndVersion() -> java.lang.String
```

ジェネリックな署名

5.6.1.1.18 *getServicePackVersion*

製品のサービスパックのバージョンを取得します (例えば, Altova MapForce Server 2018 r2 sp1 (x64)のために "1" など)。

署名

```
getServicePackVersion() -> int
```

ジェネリックな署名

5.6.1.1.19 *is64bit*

MapForce Server エンジンが64 ビットの実行可能ファイルの場合、**true** を返します。

署名

```
is64bit() -> boolean
```

ジェネリックな署名

5.6.1.1.1.20 run

MapForce Server 実行ファイル(.mfx ファイル)を実行します。成功の場合は、**true**を返し、それ以外の場合、**false** を返します。

署名

```
run(arg0:String) -> boolean
```

ジェネリックな署名

パラメーター

名前	型	説明
arg0	String	MapForce Server Execution ファイル (.mfx ファイル)へのパスを指定します。相対的なパスを指定すると、作業ディレクトリに対して解決されます。 setWorkingDirectory メソッドから呼び出すことにより作業ディレクトリを設定することができます。

5.6.1.1.1.21 setOption

マッピングを実行する前のオプションを設定します。最初の引数は、オプションの名前を指定し、2番目の引数はオプションの値を指定します。このメソッドはマッピングがグローバルリソースを消費するためにデザインされている場合とても役に立ちます ([Altova グローバルリソース](#)を参照してください)。現在サポートされているオプションは以下の通りです:

- **globalresourcefile (or gr)** – グローバルリソースファイルパス(このオプションが指定されている場合、グローバルリソース構成名も指定されている必要があります。次のアイテムを参照してください)。
- **globalresourcefile (or gr)** – グローバルリソース構成名(このオプションが指定されている場合、グローバルリソースファイルパスも指定されている必要があります。次のアイテムを参照してください)。
- **catalog** – カスタムRootCatalog.xml ファイルへのパス(このオプションによりマッピングにより使用されたURL を解決するために使用されたカスタムカタログファイルを指定します。デフォルトのカタログはプログラムインストールディレクトリのサブディレクトリにあります)。
- **taxonomy-package** – マッピングにより必要とされている場合、カスタムXBRL タクノミパッケージへのパスです。
- **taxonomy-package** – マッピングにより必要とされている場合、カスタムXBRL タクノミパッケージ構成へのパスです。

設定されたすべてのオプションはオブジェクトのライフタイムの期間有効です。同じ名前オプションを2回設定すると、前のオプションは上書きされます。明示的にすべてのオプションをクリアするためにclearOptions() メソッドを呼び出します。

署名

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

ジェネリックな署名

パラメーター

名前	型	説明
arg0	String	設定するオプション名を指定します。
arg1	String	設定するオプションの値を指定します。

5.6.1.1.1.22 *setServerPath*

MapForce Server 実行可能ファイルへのパスを設定します

署名

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

ジェネリックな署名

パラメーター

名前	型	説明
arg0	String	MapForce Server 実行可能ファイルへのパスを設定します

5.6.1.1.1.23 *setWorkingDirectory*

実行中のジョブのために現在のディレクトリを設定します (相対的パスは、作業ディレクトリに対して評価されます)。

署名

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

ジェネリックな署名

パラメーター

名前	型	説明
arg0	String	作業ディレクトリへのパスを指定します。

5.6.1.2 MapForceServerException

MapForceServerException クラスは、MapForceServer クラスによりスローされたプログラムのなアクセスを与えます。

```
public class MapForceServerException extends Exception
```

6 デジタル証明書の管理

デジタル証明書の管理は、クライアントコンピュータとWeb サーバー間のデータのやり取りの重要な一部です。マッピングはMapForce によりWindows 上でのみだけでなく、Linux またはmacOS サーバー上でMapForce Server (スタンドアロン、または FlowForce Server 実行)により実行することができるため、このセクションは多種のプラットフォーム上でのHTTPS 証明書の管理について説明しています。

安全なHyperText Transport Protocol (HTTPS) のコンテキストでは、サーバーとクライアント間の証明書を区別することが重要です。

サーバーの証明書

サーバーの証明書は、サーバーをMapForce などのクライアントアプリケーションの信頼されているエンティティとして識別します。サーバーの証明書は、商業証明機関により電子的に署名され手いる可能性があります。所属機関により自身で署名される場合もあります。この場合、MapForce 内でマッピングをデザイン中に、次の設定を指定することができます。

- サーバーの証明書がチェックされるか否か。
- 証明書の名前とホストの名前の間に不一致が検出された場合、リクエストが継続されるか否か。

MapForce のHTTP セキュリティ設定 ダイアログボックスで以下の設定を使用することができます。サーバーの証明書のチェックを選択する場合、以下に注意してください。

- 信頼される証明機関により証明書が署名されるWeb サーバーを呼び出す場合、使用中のオペレーティングシステムはサーバーの証明書を信頼するように構成されている場合が多く、追加の構成は必要ありません。
- 自身により署名される証明書を与えるWeb サーバーを呼び出す場合、(例えば 所属機関内のローカルネットワークサーバー) オペレーティングシステムを構成し、証明書を信頼するように構成する必要があります。

多くの場合、ブラウザのアドレスバー内にWeb サービスのURL を入力し、オペレーティングシステムとWeb サーバー間の信頼のレベルをチェックすることができます。サーバーが信頼されていない場合、または オペレーティングシステムがサーバーを信頼するように構成されていない場合、ブラウザは「この接続は信頼されていません」「このWeb サイトのセキュリティ証明書に問題があります」などのメッセージを表示します。ブラウザがオペレーティングシステム以外の証明書 データベースを使用する場合、(例えば Ubuntu やFirefox を使用など) ブラウザを使用してWeb サーバーの信頼レベルをチェックすることはできません。

Windows 上では、サーバーへの信頼をブラウザの命令に従い、または システムの信頼された証明機関ストアにインストールすることにより信頼を構築することができます(次を参照してください)！Windows 上でサーバーの証明書を信頼する[Windows 上でサーバーの証明書を信頼する](#)。macOS 上では、Safari 内で同様の操作を行うことができます(次を参照してください)！macOS 上のサーバーの証明書を信頼する[macOS 上のサーバーの証明書を信頼する](#)。Linux に適用可能な命令に関しては、次を参照してください！Linux 上のサーバーの証明書を信頼する[Linux 上のサーバーの証明書を信頼する](#)。

クライアントの証明書

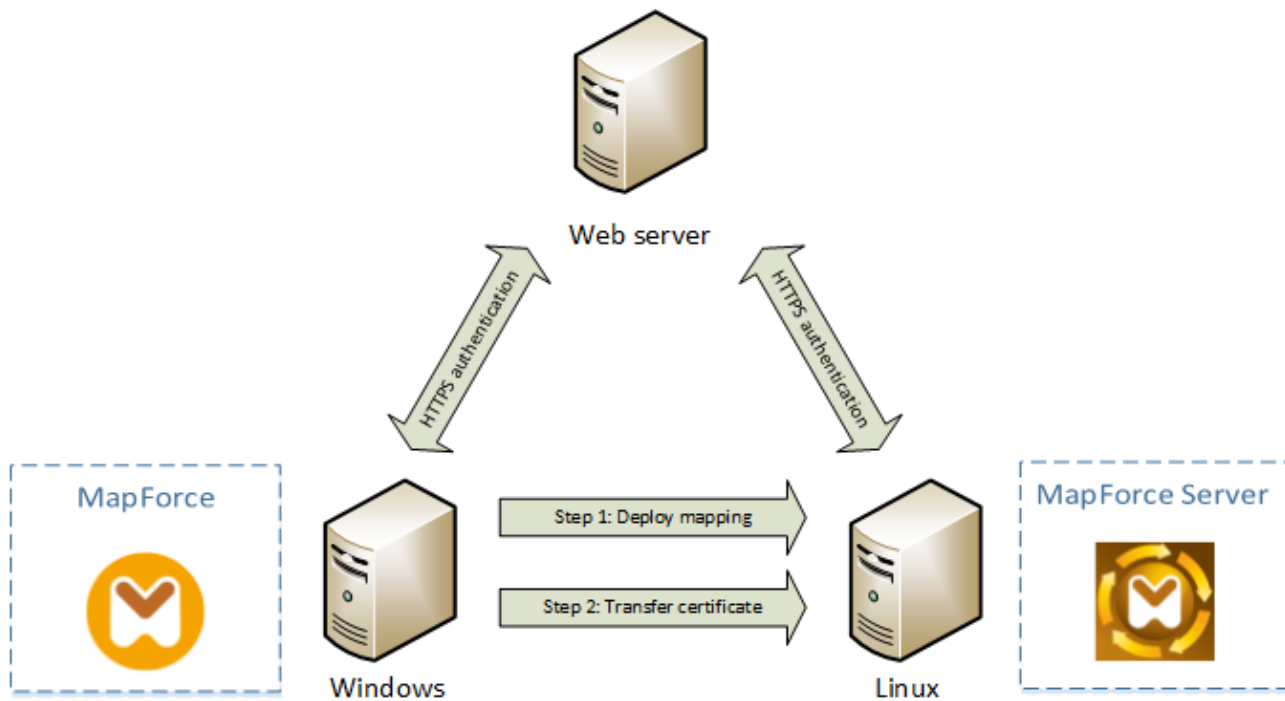
サーバーの証明書はサーバーを信頼されるエンティティとして識別するために使用されますが、クライアントの証明書は主にWeb サーバーに対して呼び出しを認証するために使用されます。クライアントの証明書を必要とするWeb サーバーを呼び出す場合、クライアント 構成命令のためのWeb サーバーの管理者に連絡をする必要があります。IIS (Internet Information Services) を例にとると、Web サーバーはHTTPS とクライアントの証明書を処理するために以下の方法により構成されます。

- HTTPS を必要とし、クライアントの証明書を無視します。
- HTTPS を必要とし、クライアントの証明書を受け入れます。
- HTTPS を必要とし、クライアントの証明書を必要とします。

Web サービスリクエストの成功、または失敗は、Web サーバーとクライアントとアプリケーションの構成により異なります。例えば Web サーバーがクライアントの証明書を必要とするように構成されている場合、呼び出しに成功するとは、呼び出しアプリケーションは有効なクライアントの証明書を表示する必要があります。

MapForce のパースペクティブからは、HTTPS を使用する Web サービス呼び出しを含むマッピングも同様に適用されます。特に、このようなマッピングの実行に成功するには、Web サーバーはクライアントの証明書を受け入れる、または、必要とするように構成され、マッピングが実行されるオペレーティングシステムは、Web サーバーへの正確なクライアントの証明書を提供します。

下のダイアグラムは、MapForce 内で仕様がされているクライアントの証明書が、MapForce Server を作動している Linux サーバーに転送されるシナリオを表しています。ターゲット オペレーティングシステムに証明書が転送されると、MapForce Server は証明書を使用して自身を Web サーバーに対して認証し、マッピングを実行することができます。



クライアントの証明書を持つマッピングを他のコンピューターにデプロイする方法

Web サービス呼び出し内の HTTPS 認証に関しては、MapForce は Secure Sockets Layer (SSL) プロトコルの後続である Transport Layer Security (TLS) を HTTP に加え、使用することができます。クライアント実装、または、サーバーが TLS をサポートしない場合、SSL のフィードバックが起こる可能性があります。

複数のプラットフォーム上で、クライアントの証明書認証を使用した Web 呼び出しをサポートするには、MapForce (と MapForce Server) は各プラットフォームの証明書の管理実装に依存し、証明書の管理がオペレーティングシステムの基となるスコープにより証明書が管理されることを保証します。各オペレーティングシステムは、証明書の管理のために異なるサポートを下のテーブルに示されるようにサポートします。

プラットフォーム	証明書の管理と実装
Windows	Windows 上では、証明書 スナッピンを使用して証明書を管理することができます (次を参照してください) Windows 上で証明書ストアにアクセスする。 TLS サポートは SChannel として既知の、安全なチャネル/ライブラリを介して使用することができます。
Linux	Linux 上では、OpenSSL (openssl) コマンドラインツールとライブラリを使用して、証明書を管理することができます。MapForce Server がインストールされている Linux マシン上で

プラットフォーム	証明書の管理と実装
	OpenSSL サポートが使用できない場合、証明書を管理する前に、ダウンロードしてインストールする必要があります。 OpenSSL ライブラリを介して TLS サポートを使用することができます (https://www.openssl.org/)。
macOS	macOS 上では「検索 > アプリケーション > ユーティリティ」にあるキーチェーンアクセスマネージャーを使用して証明書を管理することができます。 TLS サポートは、オペレーティングシステムネイティブな安全なトランスポートライブラリにより提供されます。

MapForce からの呼び出しを想定する既に成功している同じ Web サービスを使用する場合、Windows オペレーティングシステム上でマッピングを実行すると、異なる証明書の構成は通常必要ありません (Windows 上でマッピングの実行に成功するための条件に関しては、次を参照してください: Windows 上のクライアント証明書 [Windows 上のクライアント証明書](#))。しかしながら、Windows コンピューター上で MapForce を使用して、マッピングをデザインし、(異なるオペレーティングシステムで作動する)他のコンピューターにデプロイするには、クライアントの証明書を保管しない、または、デプロイされたパッケージにコピーします。Web サービスの呼び出しとマッピングの実行の成功に関しては、クライアントの証明書がターゲット オペレーティングシステム上に存在する必要があります。

Windows システムから証明書を他の Windows ベースのコンピューターに転送する場合、必要とする証明書を (秘密キーを使用して) ソースシステムからエクスポートします。(次を参照してください: Windows 上で証明書をエクスポートする [Windows 上で証明書をエクスポートする](#))。ターゲット オペレーティングシステム上の現在のユーザー「個人用」ストアに同じ証明書をインポートします (次を参照してください: Windows 上のクライアント証明書 [Windows 上のクライアント証明書](#))。

Linux と macOS プラットフォームにクライアントの証明書を転送する方法に関しては、次を参照してください: [Linux 上のクライアント証明書](#) と [macOS 上のクライアント証明書](#)。

6.1 Windows 上で証明書ストアにアクセスする

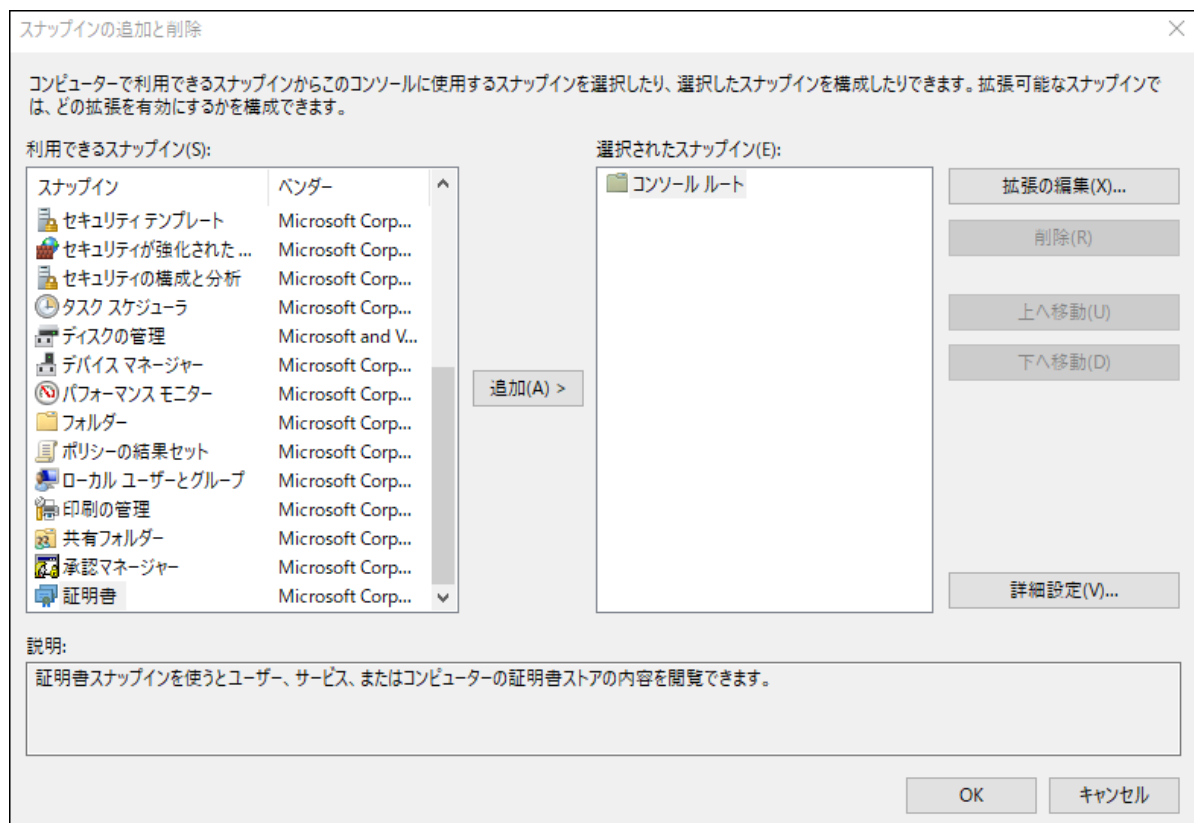
Windows 上では Microsoft 管理コンソール(MMC) スナップイン、または Internet Explorer から証明書を管理することができます。

(現在 Windows を使用するユーザーのための)証明書 スナップインを開く方法：

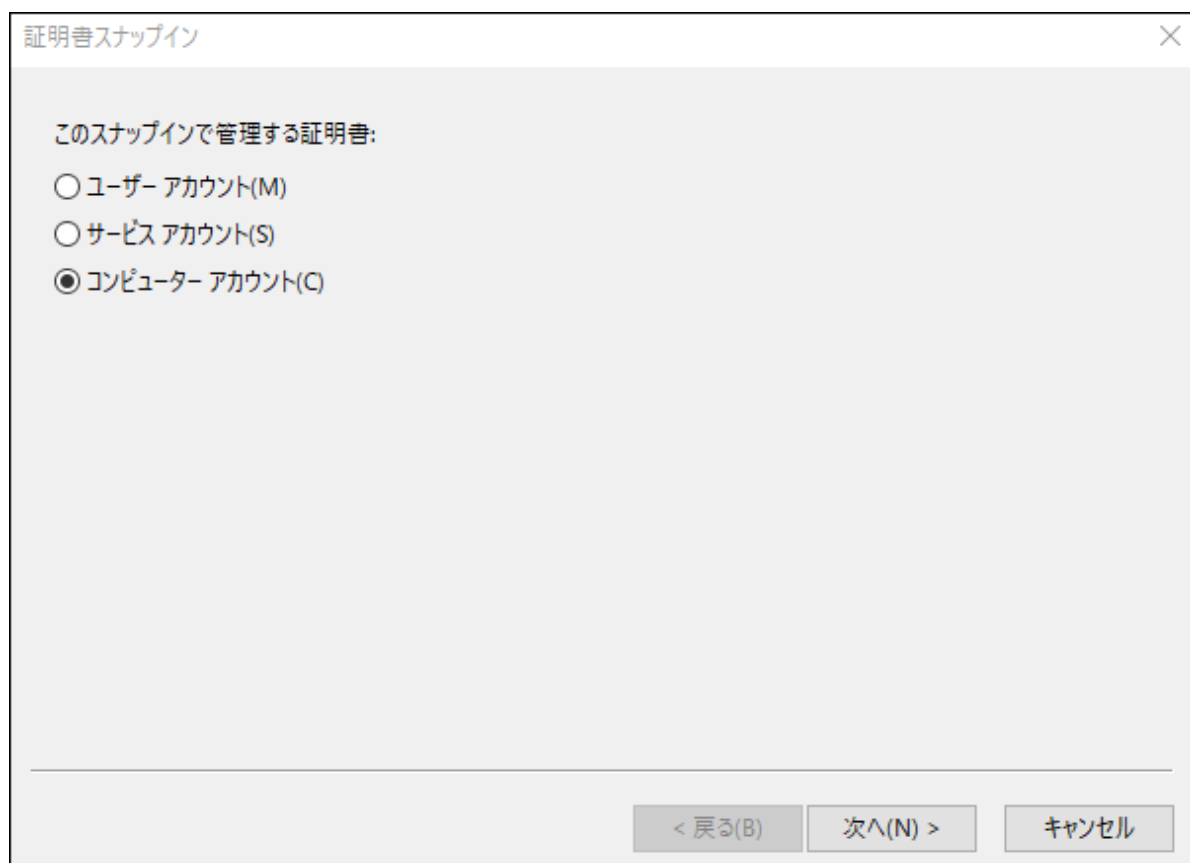
- `certmgr.msc` をコマンドラインで実行します。

(コンピュータのアカウントのために)証明書スナップインを開く方法：

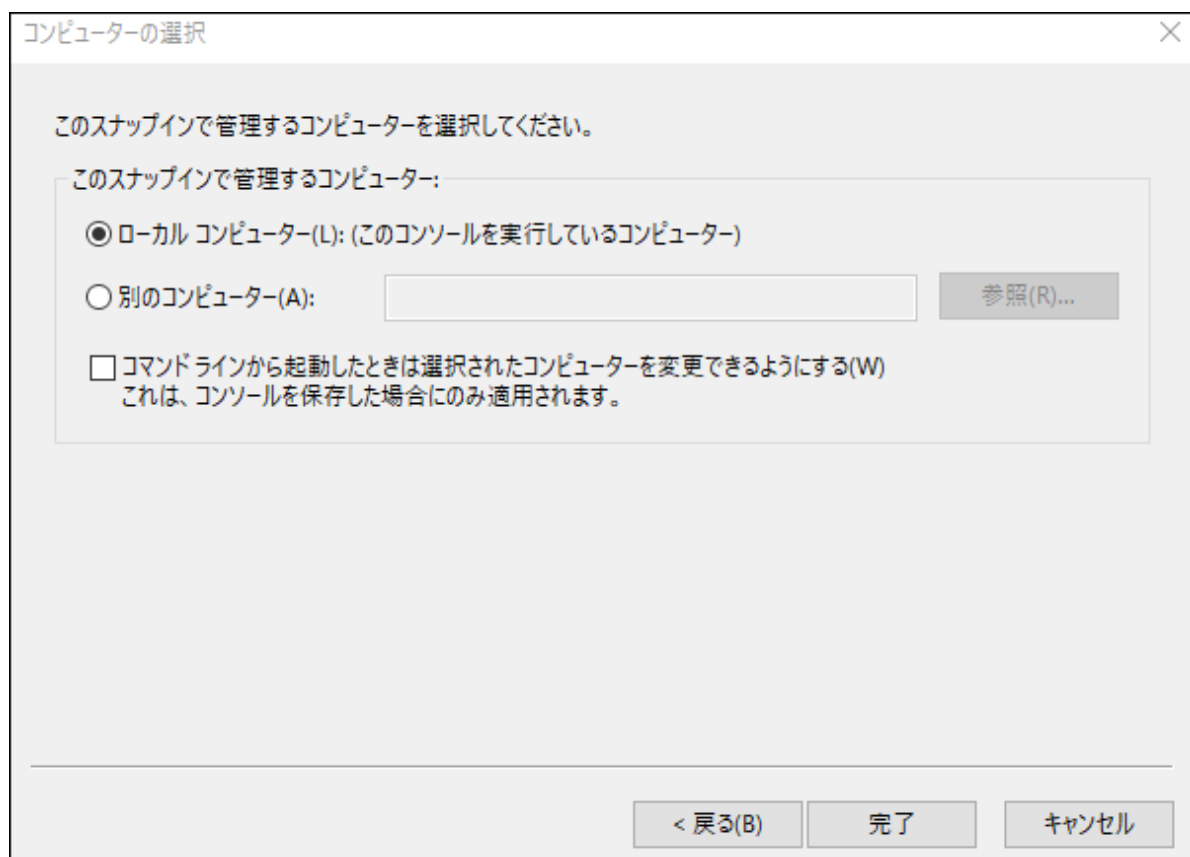
1. `mmc` をコマンドラインで実行します。
2. 「ファイル」メニューから「スナップインの追加/削除」をクリックします。
3. 「証明書」をクリックして、「追加」をクリックします。



4. 「コンピュータのアカウント」をクリックして、「次へ」をクリックします。



5. 「ローカルコンピューター」をクリックして、「完了」をクリックします。



6.2 Windows 上で証明書をエクスポートする

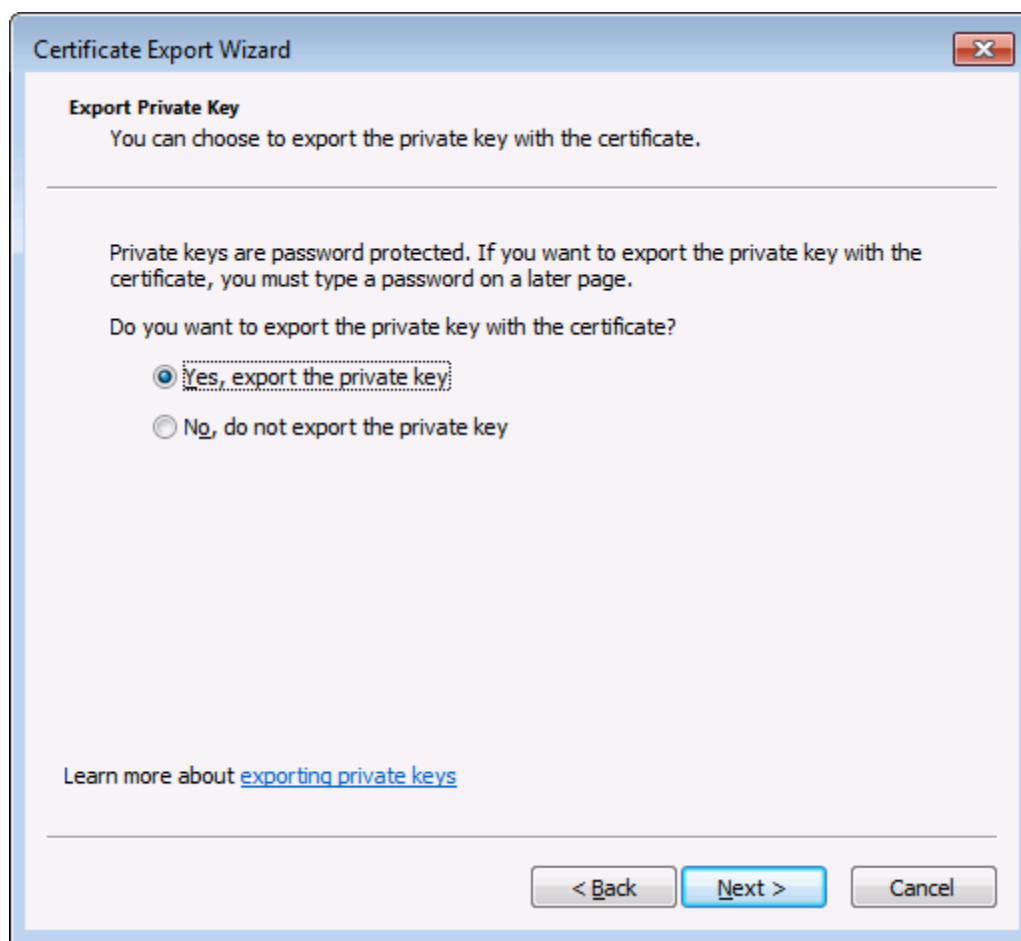
HTTPS を使用して Web サービスを呼び出し、MapForce Server または FlowForce Server を作動する macOS または Linux サーバーにデPLOYされるマッピングは、Windows 上でマッピングをデザインしテストする。同じクライアントの証明書が非 Windows オペレーティングシステムで使用可能である必要があります。このようなマッピングを MapForce Server を作動する非 Windows オペレーティングシステムで実行するには、Windows から必要とする秘密キーを持つ証明書をインポートし、ターゲット オペレーティングシステムにエクスポートします。

Windows から秘密キーを持つ証明書をエクスポートする方法:

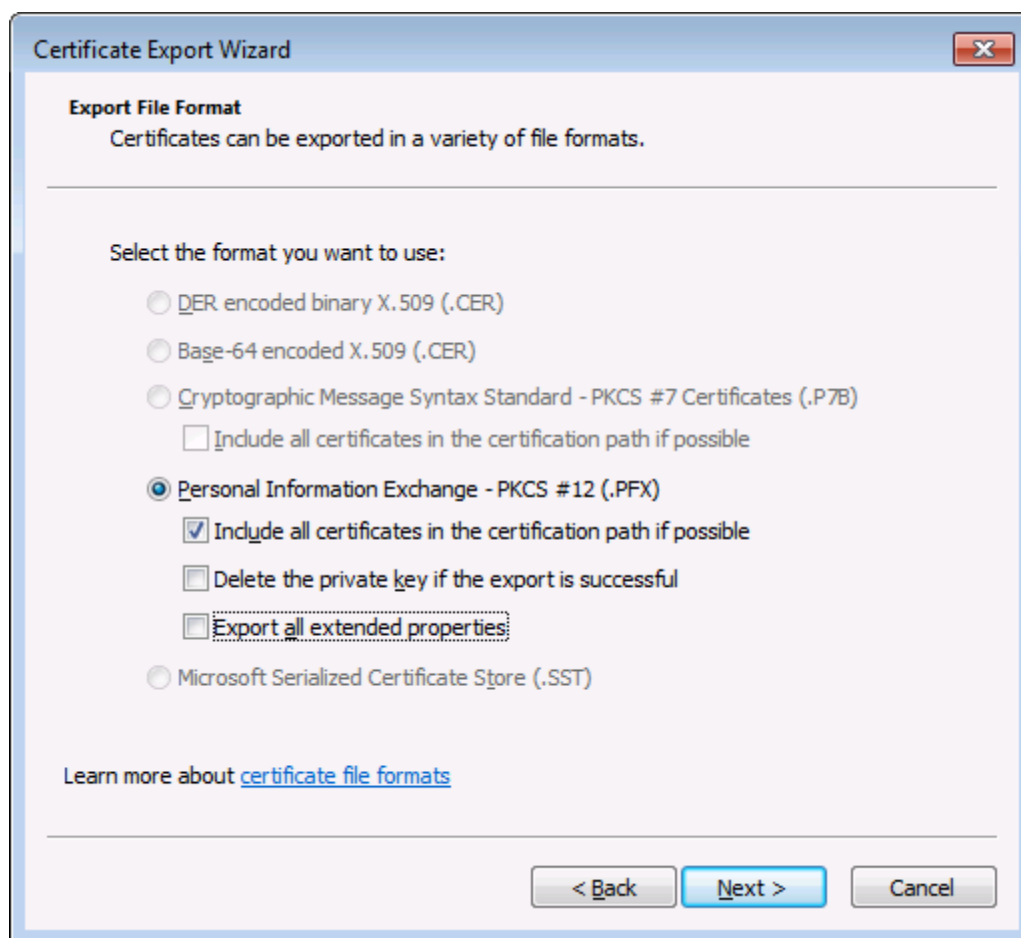
1. Windows 上で、証明書 スナップインを開きます(次を参照してください) Windows 上で証明書ストアにアクセスする[Windows 上で証明書ストアにアクセスする](#)。
2. エクスポートする証明書を右クリックし、「すべてのタスク」をポイントし、「エクスポート」をクリックします。
3. 「次へ」をクリックします。



4. Windows から秘密キーと共にエクスポートするオプションを選択して、「次へ」をクリックします。

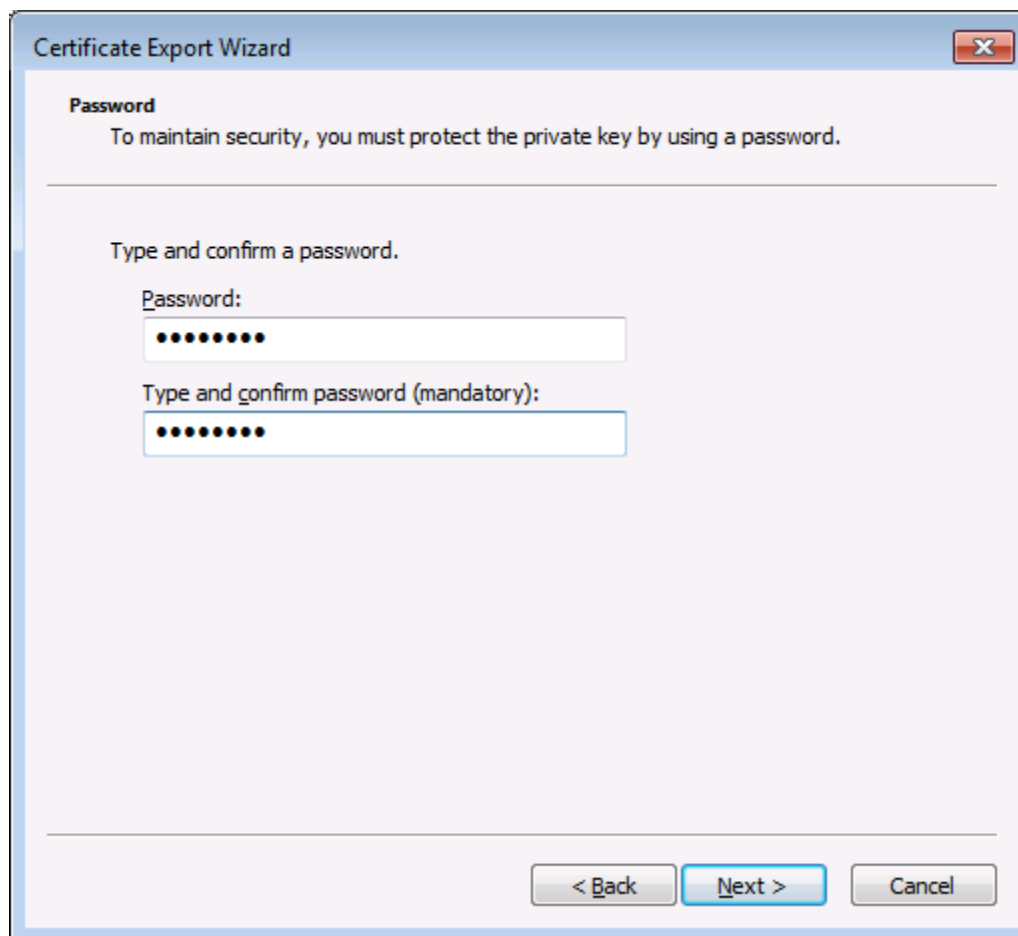


5. 個人情報のエクスチェンジ - PKCS #12 (.pfx) ファイル書式を選択し、「次へ」をクリックします。

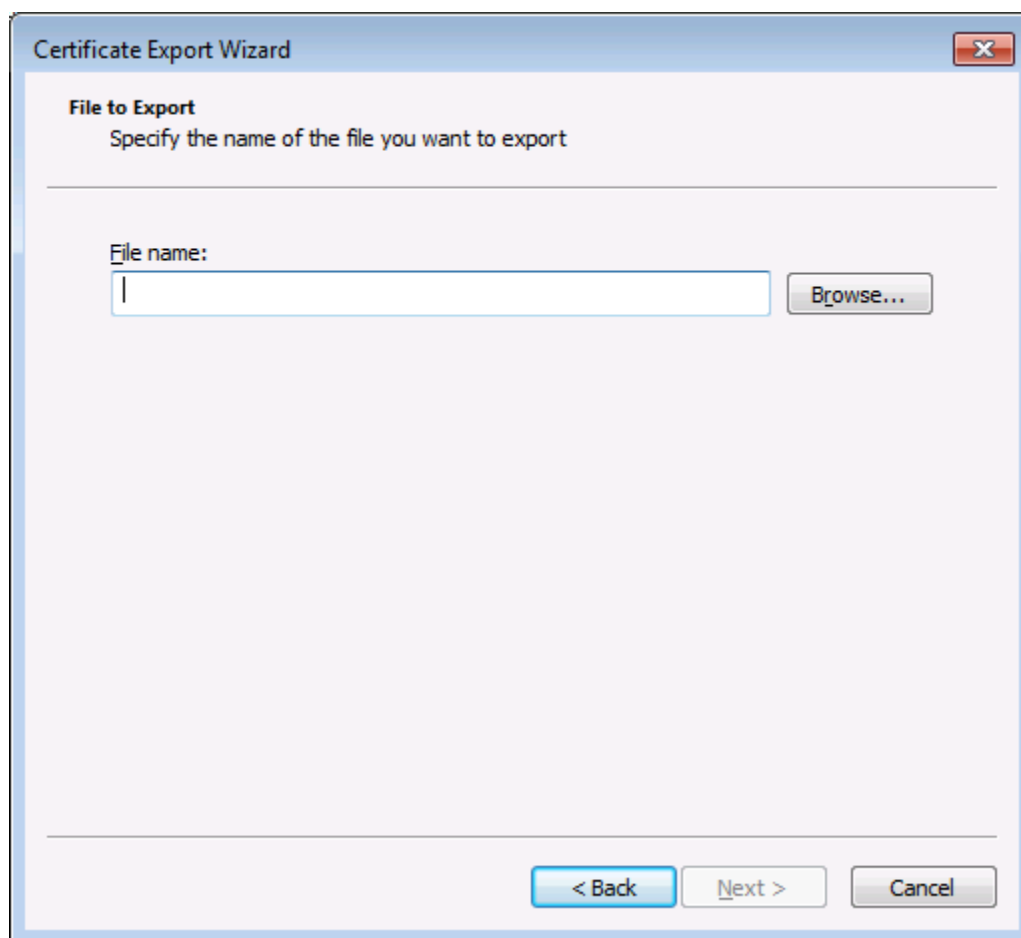


メモ エクスポート後証明書を使用できなくなるため、「エクスポートに成功すると秘密キーを削除する」が選択されていないことを確認してください。

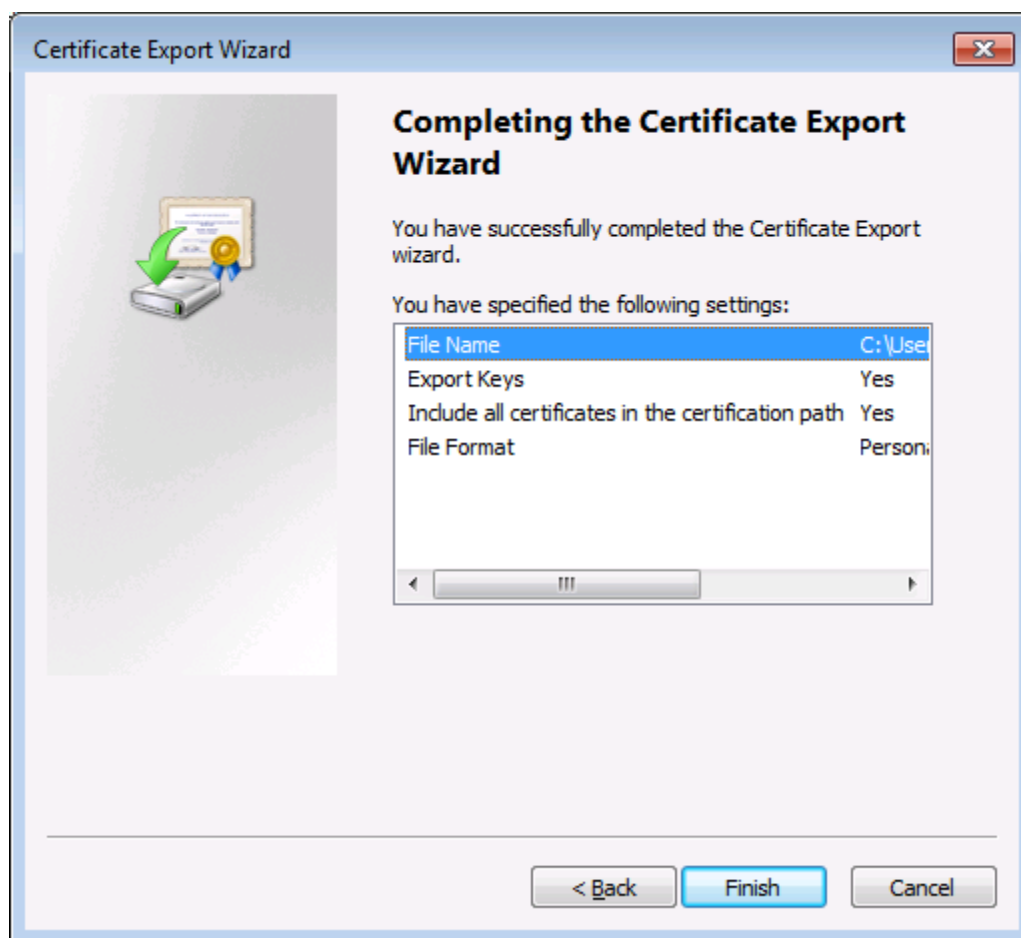
6. パスワードを入力し、「次へ」をクリックします。このパスワードは、証明書がターゲット オペレーティングシステムにコピーされると必要になります。



7. ファイルをエクスポートする場所を参照し、「次へ」をクリックします。



8. 「完了」をクリックします。



6.3 Linux 上のクライアント証明書

クライアントの証明書を介して HTTPS を使用してマッピングが Web サービス認証を含む場合、MapForce Server を作動する Linux マシンにマッピングを以下のステップでデプロイします:

1. MapForce 内で Web サービスを呼び出すマッピングを開きます。
2. Web サービスのエポネートのヘッダーをダブルクリックします。エポネートの設定 ダイアログボックスが表示されます。
3. HTTP セキュリティ設定の横の「編集」をクリックします。
4. HTTP セキュリティ設定 ダイアログボックス内で、「クライアントの証明書」をクリックして、Windows 上の「現在のユーザー 個人用」ストアから必要とする証明書を選択します。
5. マッピングを保存し、マッピング実行ファイルにコンパイル、または FlowForce Server にデプロイします。
6. Web サービス呼び出しにより必要とされるクライアントの証明書をターゲット オペレーティングシステムに移動します。証明書が秘密キーを確認し、「拡張キー使用法」プロパティでクライアントの認証が目的として含まれます。

Linux にクライアントの証明書を転送する方法:

1. Windows から *Personal Information Exchange - PKCS #12 (.pfx)* ファイル書式内の秘密キーを持つクライアントの証明書をエクスポートします(次を参照してください) Windows 上で証明書をエクスポートする [Windows 上で証明書をエクスポートする](#)。
2. Linux マシンに証明書ファイルをコピーします。
3. コマンドを使用して、.pfx ファイルを .pem に変換します:

```
openssl pkcs12 -in cert.pfx -out "John Doe.pem" -nodes
```

このコマンドは、秘密キーを暗号化することなく .pfx ファイルを解析し、.pem ファイルを出力します。暗号化された秘密キーはパスワードをプロンプトし、サーバー実行ではサポートされていません。

マッピングの実行

クライアントの証明書として .pem ファイルを使用するために MapForce Server に命令するには、マッピングを実行する際に `--certificatespath` パラメータを設定します。`--certificatespath` パラメータはディレクトリのパスを定義し現在のマッピングにより必要とされる証明書に保管されます。例えば、証明書ファイルパスが `/home/John/John Doe.pem` の場合、`--certificatespath` は `/home/John` に設定される必要があります。

デフォルトでは `--certificatespath` パラメータが与えられないと、MapForce Server は現在のユーザーのディレクトリ `$HOME/.config/altova/certificates` 内の証明書を検索します。

マッピングの実行に成功すると、証明書ファイルが .pem 拡張子を持つことが期待され、ファイル名は証明書のスペースを含む共通名 (CN) (例えば、**John Doe.pem**) に一致する必要があります。CN にスラッシュ (/) が含まれる場合、アンダースコア (_) 文字で置き換えられる必要があります。

FlowForce Server ジョブとしてマッピングを実行する場合、証明書ファイルを `$HOME/.config/altova/certificates` ディレクトリにコピーします。ジョブを実行する際に、FlowForce Server はマッピングにより必要とされる証明書ファイルを検索するためにこのディレクトリを使用します。

セキュリティ上の理由から、機密な情報が含まれているため、証明書ファイルを他のユーザーから読み取れないことを確認してください。

6.4 macOS 上のクライアント証明書

HTTPS クライアントの証明書を使用する Web サービス 認証が含まれる場合、MapForce Server を作動する macOS にこのようなマッピングをデプロイするために次のステップを行ってください！

1. MapForce 内で Web サービスを呼び出すマッピングを開きます。
2. Web サービスのエポネートのヘッダーをダブルクリックします。エポネートの設定 ダイアログボックスが表示されます。
3. HTTP セキュリティ設定の横の「編集」をクリックします。
4. HTTP セキュリティ設定 ダイアログボックス内で、クライアントの証明書ををクリックして、必要とする証明書を選択します。
5. 証明書名がサーバーのホスト名に一致しない場合、「証明書とリクエスト間の一致しない名前を許可する」を選択します。保存して、ターゲット オペレーティングシステムにマッピングをデプロイします。
6. (次を参照してください！MapForce マッピングのデプロイ)。
7. Web サービス呼び出しにより必要とされるクライアントの証明書をターゲット オペレーティングシステムに移動します。証明書が秘密キーを持っていることを確認すると、証明書の「拡張キー使用法」プロパティがクライアントの認証が目的として含まれます。

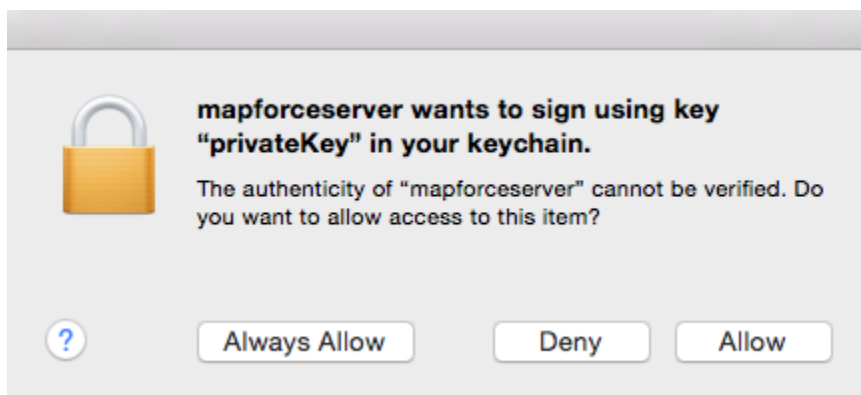
macOS にクライアントの証明書を移動させる方法：

1. Windows から秘密キーを持つクライアントの証明書を Personal Information Exchange - PKCS #12 (.pfx) ファイル書式でエクスポートします (次を参照してください！Windows 上で証明書をエクスポートする[Windows 上で証明書をエクスポートする](#))。pfx ファイルを macOS にコピーします。
2. おざりされていない場合、オペレーティングシステムがサーバーの証明書を信頼していることを確認してください (次を参照してください！Mac OS 上のサーバーの証明書を信頼する[Mac OS 上のサーバーの証明書を信頼する](#))。
3. 「検索 > アプリケーション > ユーティリティ」からキーチェーンを実行します。
4. 「ファイル」メニューから「アイテムのインポート」をクリックします。
5. ステップ1でWindows からエクスポートされたクライアントの証明書を参照し、目的のキーチェーンを選択します。
6. 「開く」をクリックして、暗号化された証明書とパスワードを入力します。

マッピングの実行

MapForce Server `run` コマンドを使用してマッピングを実行する準備が整いました。以下の点に注意してください！

- SSH を使用してマッピングを実行した場合、`security unlock-keychain` コマンドを使用して、キーチェーンを最初二案ロックします。
- macOS のグラフィカルなユーザーインターフェイスを使用してマッピングを実行した場合、MapForce Server サーバーにキーチェーンへのアクセスを許可するようにプロンプトされると、「許可」をクリックします。



6.5 Linux 上のサーバーの証明書を信頼する

Linux 上では信頼される証明書をシステムの証明書ストアに以下のようにインポートすることができます。

サーバーの証明書の信頼性が確実の場合、次のステップをおこなってください。

Debian と Ubuntu では、以下のステップに従います：

1. Web サーバーの証明書ファイルを次のディレクトリにコピーします。

```
sudo cp /home/downloads/server_cert.crt /usr/local/share/ca-certificates/
```

2. 証明書を以下のように更新します

```
sudo update-ca-certificates
```

CentOS では、以下のステップに従います：

1. ca-certificates パッケージをインストールします：

```
yum install ca-certificates
```

2. 動的な証明機関構成機能を有効化します：

```
update-ca-trust enable
```

3. サーバーの証明書を次のディレクトリにコピーします：

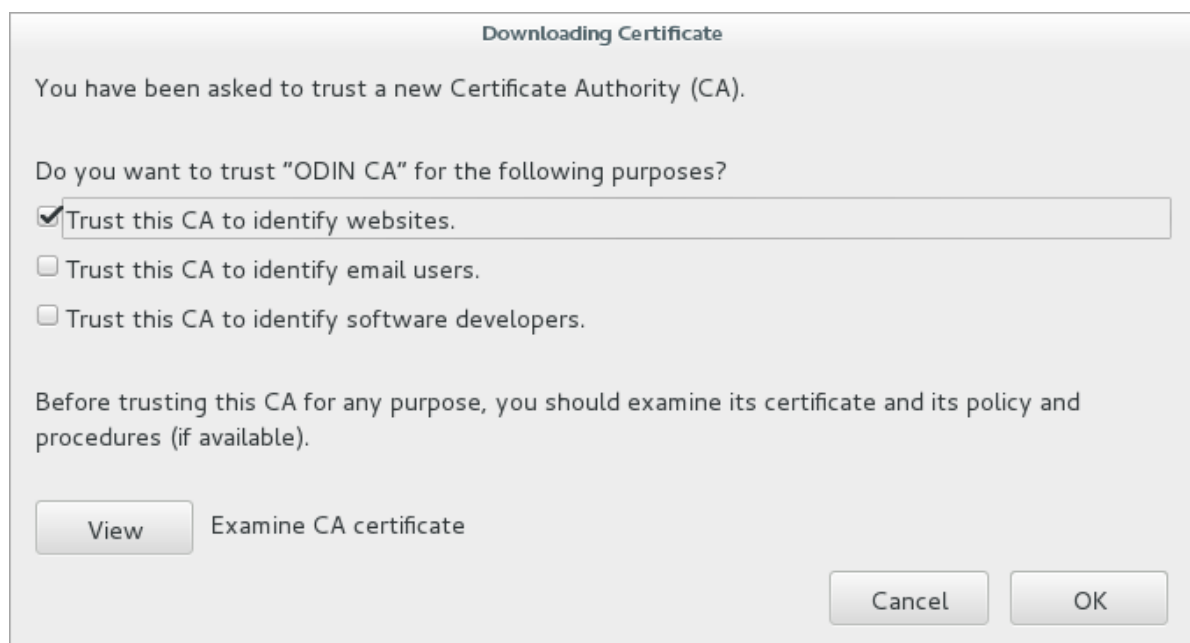
```
cp server_cert.crt /etc/pki/ca-trust/source/anchors/
```

4. コマンドを使用します：

```
update-ca-trust extract
```

ブラウザからのみサーバーにアクセスする必要がある場合、各ブラウザストアに証明書をインポートするのみで十分です。各ブラウザにより正確な手続きは異なります。例えば、Firefox 59.0.2 では、以下を行うことができます。

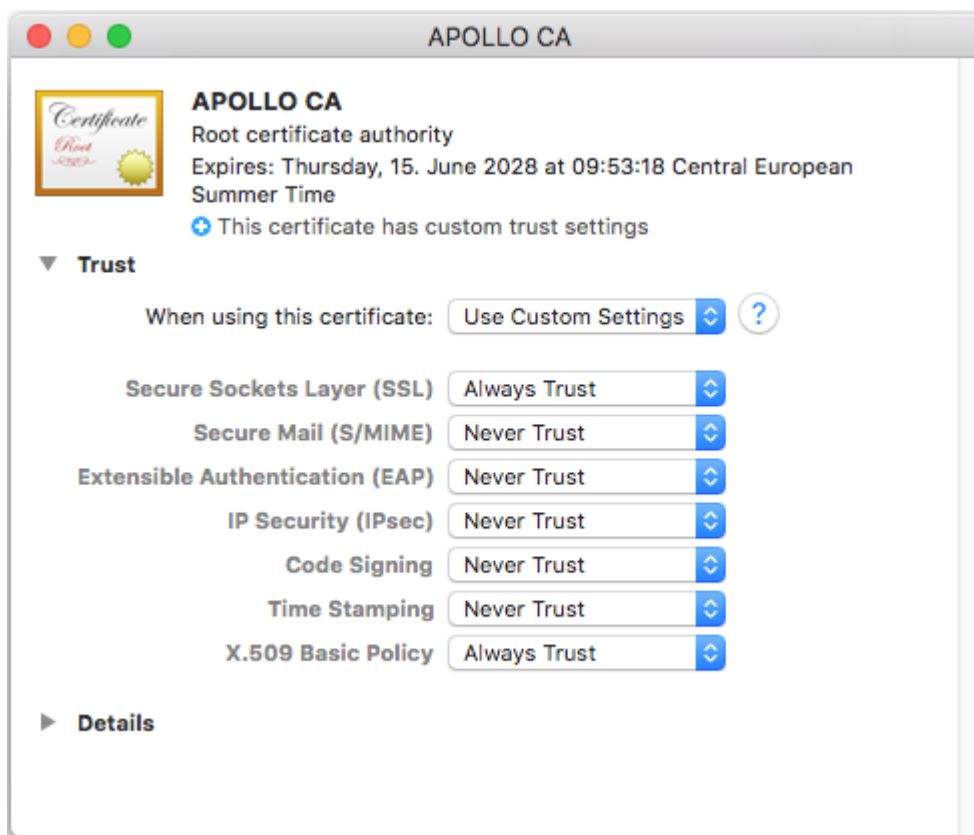
1. 「オプション | プライバシーとセキュリティ」から、「証明書を表示」をクリックします。
2. 「認証局証明書」タブで、「インポート」をクリックし、以前で作成されているルート証明書ファイルを参照します。
3. プロンプトされる「この証明書をウェブサイトの識別に使用する」を選択します。



6.6 macOS 上のサーバーの証明書を信頼する

macOS では、信頼される証明書を Keychain Access に以下のようにインポートすることができます。

1. Keychain Access を実行します。
2. 「システム」をクリックして、「証明書」をクリックします。
3. 「ファイル」メニューから「アイテムのインポート」をクリックします。
4. 信用される証明書を参照し、「開く」をクリックします。
5. Keychain Access パスワードをプロンプトされると入力し、「キーチェーンの変更」をクリックします。
6. 証明書をダブルクリックし、「信頼」セクションを拡張し、「常に信頼する」を選択します。



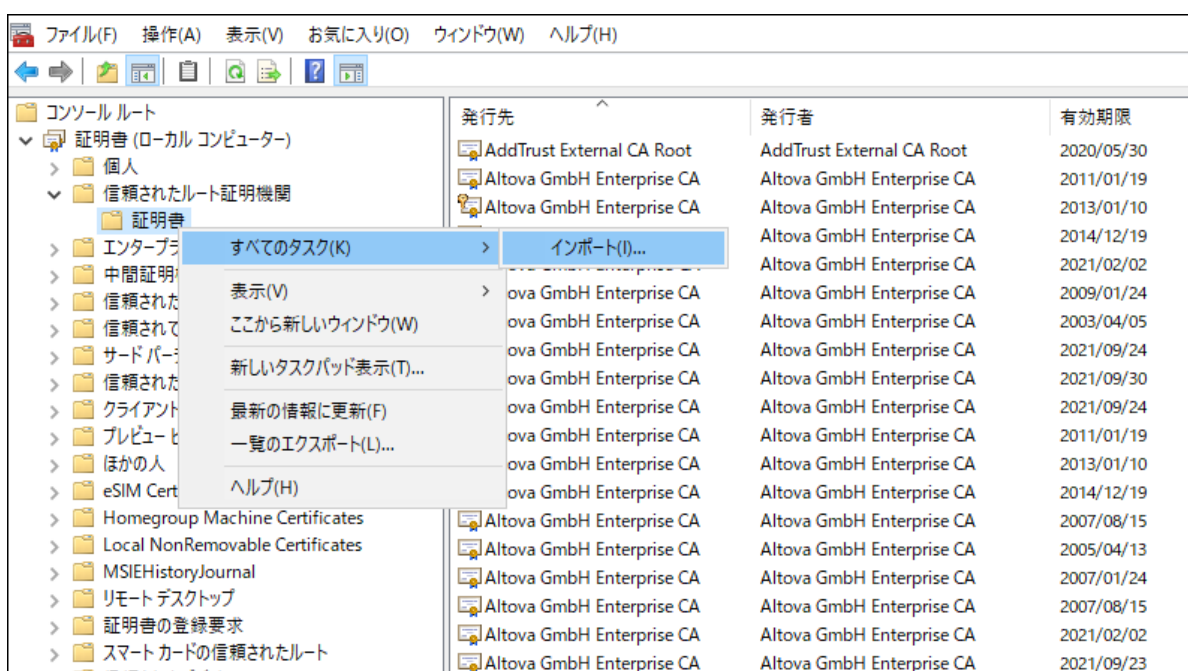
6.7 Windows 上でサーバーの証明書を信頼する

Windows では、信頼された証明書をシステム証明書ストアに以下のようにインポートすることができます。

1. Windows 証明書ストアをコンピュータアカウントのために開きます、[Windows 証明書ストアへのアクセス](#)を参照してください。

Web サーバー証明書の認証を確認するだけの場合、次のステップを実行します。

2. 「信頼されたルート証明書機関」で「証明書」を右クリックし、「すべてのタスク | インポート」を選択し、証明書のインポートウィザードに従ってください。

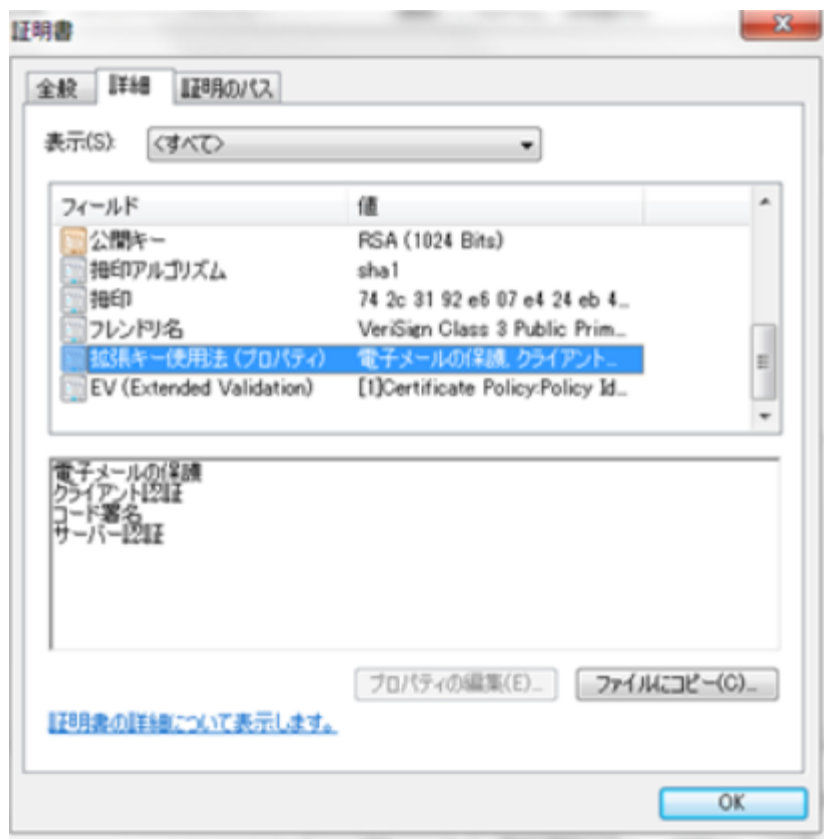


詳細に関しては [https://technet.microsoft.com/en-us/library/cc754489\(v=ws.11\).aspx](https://technet.microsoft.com/en-us/library/cc754489(v=ws.11).aspx) を参照してください。

6.8 Windows 上のクライアント証明書

クライアントの証明書を必要とするマッピングを Windows 上で実行する場合、マッピングの実行に成功するための条件は以下の通りです:

- クライアントの証明書が「現在のユーザー個人用」証明書ストア内に存在する必要がある場合があります（マイストアとして称されることもあります）。ストアに証明書が存在する場合は、証明書インポートウィザードを使用してインポートされる必要があります。命令に関しては、次を参照してください: [https://technet.microsoft.com/en-us/library/cc754489\(v=ws.11\).aspx](https://technet.microsoft.com/en-us/library/cc754489(v=ws.11).aspx)。
- 証明書は秘密キーを持つ必要があります。
- 証明書の「拡張キー使用法」プロパティが「クライアント認証」を目的として含む必要があります。



MapForce の現在のバージョンでは、MapForce により使用することのできるライブラリの制限により、Windows はマッピングが実行されると、必要とする証明書を自動的に証明書ストアから選択します。「現在のユーザー個人用」証明書ストアをフィルターすると、マッピングの実行は成功し、サーバーは適切な証明書を検索します。HTTPS 認証（および、証明書の選択操作）は、Windows により管理され、MapForce または MapForce Server によりコントロールされていないことに注意してください。一部の状況、複数の証明書が「現在のユーザー個人用」ストア内に存在し、適切でない証明書がオペレーティングシステムにより選択される場合があります。結果、マッピングの実行に失敗する場合があります。「現在のユーザー個人用」ストア内で使用することのできる証明書の数を制限することにより、このような状況を回避することができます。

7 XBRL タクソノミ マネジャー

XBRL タクソノミ マネジャーはMapForce Server を含むAltova XBRL-有効化されているアプリケーション全体で使用するためにXBRL タクソノミをインストールし管理する中央化された方法を提供するツールです。Windows ではXBRL タクソノミ マネジャーには視覚的なユーザーインターフェイスがあり、コマンドラインでも使用可能です。Linux とMac* 上ではコマンドラインでのみツールを使用することができます。

* Linux と macOS オペレーティングシステムをこれらのオペレーティングシステム上でXBRL タクソノミ マネジャーをMapForce Server、StyleVision Server、またはRaptorXML+XBRL Server などのAltova クロスプラットフォームサーバーと共に実行している場合に適用可能です。

XBRL タクソノミ マネジャーには以下の機能を搭載しています：

- 使用中のコンピュータにインストールされているXBRL タクソノミをビューして、新規のバージョンをダウンロード可能を確認してください。
- Altova 製品のリースサイクルに関わらずXBRL タクソノミの新しいバージョンをダウンロードします。全てのタクソノミはXBRL タクソノミ マネジャーによりアクセス可能なオンラインベースのパッケージストレージ上でAltova により管理されており、使用可能になるとすぐにダウンロードすることが可能です。
- タクソノミの複数のバージョン(または、必要な場合全てのバージョン)をインストールまたはアンインストールする
- 単一のXBRL タクソノミは「パッケージ」を表しており、他のタクソノミ上に依存関係が存在する場合があります。特定のタクソノミをインストールまたはアンインストールすることを選択すると、依存関係を持つタクソノミが検知され、自動的にインストールまたはアンインストールされます。グラフィカルなユーザーインターフェイス(または適用できる場合コマンドライン)は依存が追加または削除されることを通知します。
- XBRL タクソノミ マネジャーにより管理されるXBRL タクソノミはインスタンス、またはスキーマドキュメントのURI レファレンスがインターネットからファイルを抽出することとは異なり、ローカルファイルから解決されることを有効化するXML カタログ メカニズムから利益を得ることができます。通常パフォーマンス上の理由からポート URI からのスキーマのレスポンスが実質的ではない場合、大きなXBRL タクソノミの場合これはとても重要です。

XBRL タクソノミ マネジャーはXBRL タクソノミの管理者が[Altova XBRL-有効化されたアプリケーション](#)の一つを使用していることを必要とします。欧州銀行監督局レポーティングフレームワークタクソノミ、US-GAAP 金融レポーティングタクソノミ、他の多種の国別、またはドメイン固有のXBRL タクソノミが含まれます。完全なリストを確認するにはXBRL タクソノミ マネジャーを実行またはlist コマンドをコマンドラインで実行します。または[インストール済みパッケージをビュー](#)を参照してください。

カスタムXBRL タクソノミ

XBRL タクソノミ マネジャーに含まれるカスタムXBRL タクソノミと作業する場合、これを以下のよう有効化します：

- Altova XBRL-有効化済みのデスクトップアプリケーションからツール | オプション メニューコマンドを実行し、XBRL | タクソノミ パッケージ 設定ページに移動します。設定ページからカスタムXBRL タクソノミの.zip パッケージを参照してください
- サポートするコマンドまたはAPI メソッドを実行する場合 Altova XBRL-有効化済みサーバーアプリケーションから taxonomy-package または--taxonomy-package-config-file オプションを提供してください。例えば、これらのオプションはvalxbrl またはvalxbrltaxonomy などのRaptorXML+XBRL Server 内のXBRL 検証コマンドまたはMapForce Server 内のrun コマンド(または関連するAPI メソッド) によりサポートされています。

Altova XBRL-有効化済みアプリケーション

以下のAltova アプリケーションはXBRL-有効化済みで、XBRL タクソノミ マネジャーにより提供される機能によりサポートを受けています：

- Excel のためのAltova XBRL アドイン(EBA, Solvency II)
- MapForce Enterprise Edition
- MapForce Server

- MapForce Server Advanced Edition
- RaptorXML+XBRL Server
- StyleVision Server
- StyleVision Enterprise Edition
- XMLSpy Enterprise Edition

XBRL タクソノミ マネジャーを使用したXBRL タクソノミ 内での変更は同じコンピュータにインストールされている場合、上にリストされたすべてのアプリケーションで反映されます。

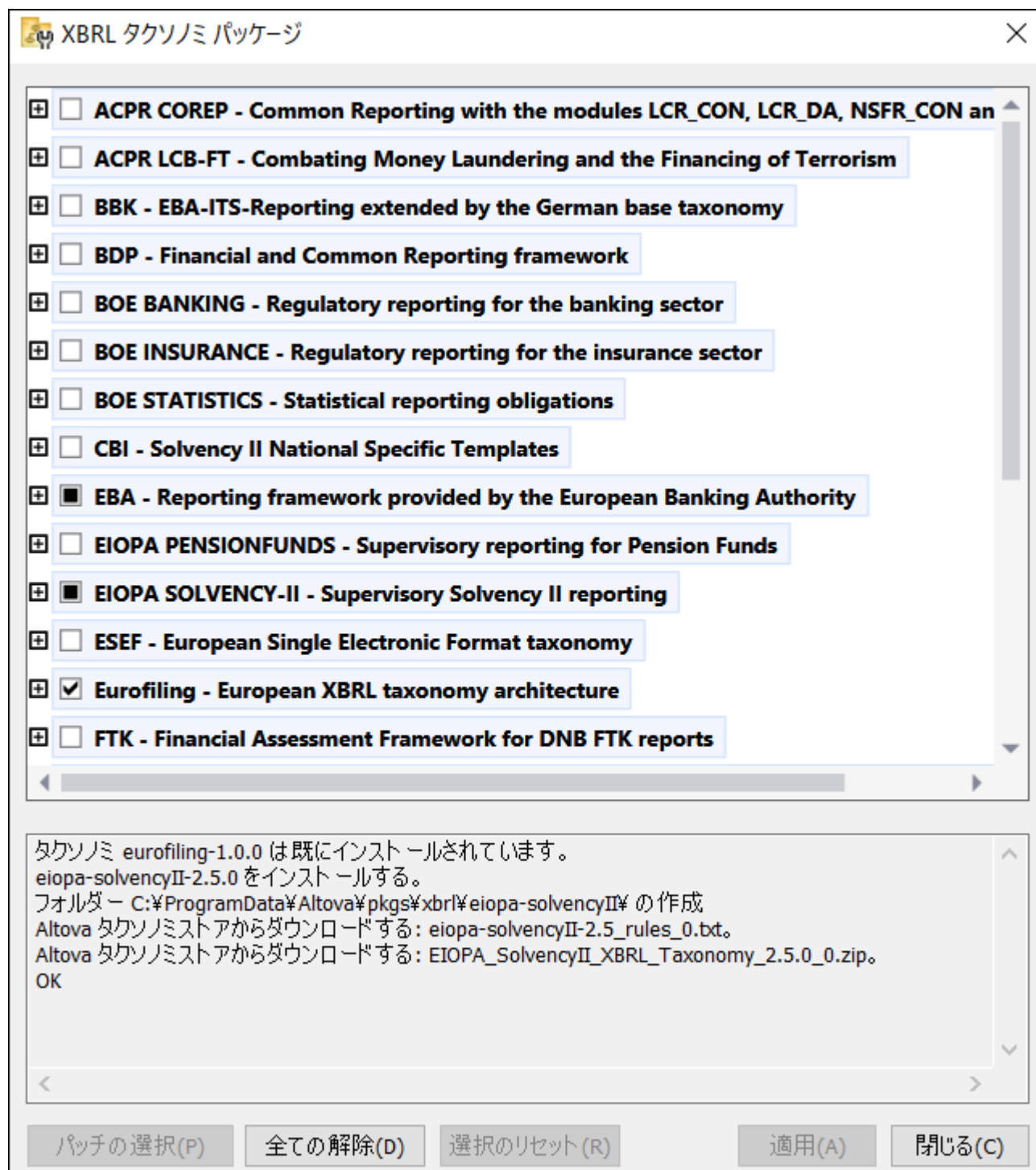
インストール

Altova XBRL-有効化済みアプリケーション、または **Altova Mission Kit Enterprise Edition** をインストールするとXBRL タクソノミ マネジャーは自動的にインストールされます。同様に、Altova XBRL-有効化済みアプリケーション、または **Altova Mission Kit Enterprise Edition** をコンピュータからアンインストールすると自動的にアンインストールされます。

しくみ

Altova は製品内で使用されている全てのXBRL タクソノミを保管するオンラインタクソノミストレージを管理しています。このタクソノミストレージは定期的に更新されます。例えば、発行機関により関連するタクソノミの新規のバージョンがリリースされた後などに更新されます。

XBRL タクソノミ マネジャーをグラフィカルなユーザーインターフェイスで実行すると、タクソノミをインストール、更新またはアンインストールすることができるダイアログボックス内に使用可能な最新のタクソノミの情報が表示されます。コマンドラインインターフェイスで同じアクションを実行することができます。Linux と macOS 上では、コマンドラインインターフェイスはXBRL タクソノミ マネジャー とのインタラクションのみです。



Altova Web サイト (<https://www.altova.com/download>) からダウンロードされた running.altova_taxonomies ファイルを実行してタクソノミをインストールすることができます。更にXBRL タクソノミをロードするアクションを取る場合 MapForce Server は不足するXBRL タクソノミをインストールするようプロンプトします。

タクソノミのインストール方法に関わらず、インストール済みのタクソノミに関する情報はコンピュータ内で中央化された場所でトラックされてい

ます。これはローカルキャッシュディレクトリとして知られています。ローカルキャッシュディレクトリにはオンデマンドでインストールされる実際のタクソノミファイル(を除き) Altova のパッケージに関する情報が含まれています。ローカルキャッシュディレクトリは以下のパスで見つけることができます:

<i>Linux</i>	/var/opt/Altova/pkg
<i>macOS</i>	/var/Altova/pkg
<i>Windows</i>	C:\ProgramData\Altova\pkg

ローカルコンピュータにオンラインストレージの最新の状態を作成するためローカルキャッシュディレクトリは時々自動的に更新されます。更に具体的には、キャッシュは以下のように更新されます:

- XBRL タクソノミマネジャーを **実行** する場合。
- 同じカレンダーの日付で MapForce Server を実行する場合。
- MapForce Server が既に実行中の場合、キャッシュディレクトリは24 時間ごとに更新されます。
- コマンドラインインターフェイスから更新を実行しローカルキャッシュをオンラインストレージからオンデマンドで更新することができます。

タクソノミをインストールまたはアンインストールすると、ローカルキャッシュディレクトリは使用可能な情報、およびインストールされているタクソノミの情報、およびタクソノミファイル自身と共に自動的に更新されます。

ローカルキャッシュディレクトリはインストールまたはアンインストールするタクソノミをベースに自動的に管理されます。手動で変更または削除されてはいけません。XBRL タクソノミ マネジャーを元の状態に戻す必要がある場合、コマンドラインインターフェイスの `reset` コマンドを実行してから `initialize` コマンドを実行します。または `reset` コマンドを `-i` オプションと共に実行してください。

7.1 XBRL タクソノミ マネジャー の実行

XBRL タクソノミ マネジャーを下にリストされるアプローチで適用可能な場合実行することができます。

グラフィカルなユーザーインターフェイス (Windows のみ)

MapForce Server が Windows 上で作動する場合 XBRL タクソノミ マネジャーを以下のように作動することができます:

- Altova Web サイトからダウンロードされた `.altova_taxonomies` 拡張子を持つファイルを (ダブルクリックして) 実行します。
- Windows コントロールパネルから Altova タクソノミマネジャー エントリを右クリックして、変更 または アンインストール をコンテキストメニューから選択します。

更に MapForce Server のインストール後チェックボックス Altova タクソノミマネジャーの呼び出し がインストールウィザードの最後のページで使用可能になります。

コマンドライン インターフェイス (Windows)

XBRL タクソノミ マネジャーをコマンドインターフェイスから実行する方法

1. コマンドプロンプトを開きディレクトリを `C:\ProgramData\Altova\SharedBetweenVersions` に変更します。
2. コマンドラインでヘルプを表示するには、以下を実行します:

```
TaxonomyManager.exe --help
```

コマンドライン インターフェイス (Linux, macOS)

XBRL タクソノミ マネジャーをコマンドインターフェイスから実行する方法

1. ターミナルウィンドウを開き `%INSTALLDIR%` がプログラムのインストールディレクトリである `%INSTALLDIR%/bin` にディレクトリを変更します。 [重要な注](#)を参照します。
2. コマンドラインでヘルプを表示するには、以下を実行します:

```
sudo ./taxonomymanager --help
```

7.2 タクソノミのインストール

タクソノミを下にリストされるアプローチで適用可能な場合インストールすることができます。

以下の点に注意してください：

- XBRL タクソノミ マネジャー からタクソノミをインストールまたはアンインストールすると同じコンピューター上のすべてのユーザーアカウントに影響を及ぼします。
- XBRL タクソノミ マネジャー からタクソノミをインストールまたはアンインストールすると同じコンピューター上にインストールされている Altova XBRL-有効化済みのすべてのアプリケーションに影響を与えます。
- 現在のタクソノミは他のタクソノミ上に依存関係があり、依存するタクソノミもインストールされます（または適用可能な場合アンインストールされます）。

グラフィカルなユーザーインターフェイス (Windows のみ)

タクソノミをインストールする方法

1. **実行** する。
2. インストールするタクソノミまたはタクソノミバージョンの横のチェックボックスを選択し [適用] をクリックします。

または、**altova_taxonomies** 拡張子を持つファイルを Altova Web (<https://www.altova.com/taxonomy-manager>) サイトからダウンロードし、**altova_taxonomies** ファイルを実行するためにダブルクリックします。**altova_taxonomies** ファイルを実行する際に XBRL タクソノミ マネジャー が開かれます。

コマンドライン インターフェイス (Windows)

タクソノミをインストールするには、以下を実行します：

```
TaxonomyManager.exe install FILTER...
```

FILTER が以下を意味する箇所：

1. フォーマット **<name>-<version>** 内のタクソノミ識別子の例: **eba-2.10**、**us-gaap-2020.0**。使用可能なすべてのタクソノミ識別子をビューするには **list** コマンドを実行します。
2. Altova Web サイトからダウンロードされた **altova_taxonomies** ファイル。

コマンドライン インターフェイス (Linux, macOS)

タクソノミをインストールするには、以下を実行します：

```
sudo ./taxonomymanager install FILTER...
```

FILTER が以下を意味する箇所：

1. フォーマット **<name>-<version>** 内のタクソノミ識別子の例: **eba-2.10**、**us-gaap-2020.0**。使用可能なすべてのタクソノミ識別子をビューするには **list** コマンドを実行します。
2. Altova Web サイトからダウンロードされた **altova_taxonomies** ファイル。

オンデマンドでタクソノミをインストールする

MapForce Server が特定の XBRL タクソノミを検知すると使用中のコンピューターにインストールされる必要があります。オンデマンドでタ

タクソノミをインストールしようとしてプロンプトされる場合があります。例えば、XBRL タクソノミのロードに必要なコマンドを実行し、XBRL タクソノミが現在インストールされていない場合でも、コマンドはキャンセルされ、しかし類似したエラーメッセージが表示されます:

```
Unable to load a schema with target namespace [...] from [...].
Details:
I/O Error: File [...] is part of missing taxonomy [...].Call 'taxonomymanager install
[...]'.

```

不足するタクソノミをインストールには `install` コマンドをエラーメッセージにより提案された通り実行します。`list` コマンド を実行して以前にインストールされたすべてのタクソノミを常に確認することができます。

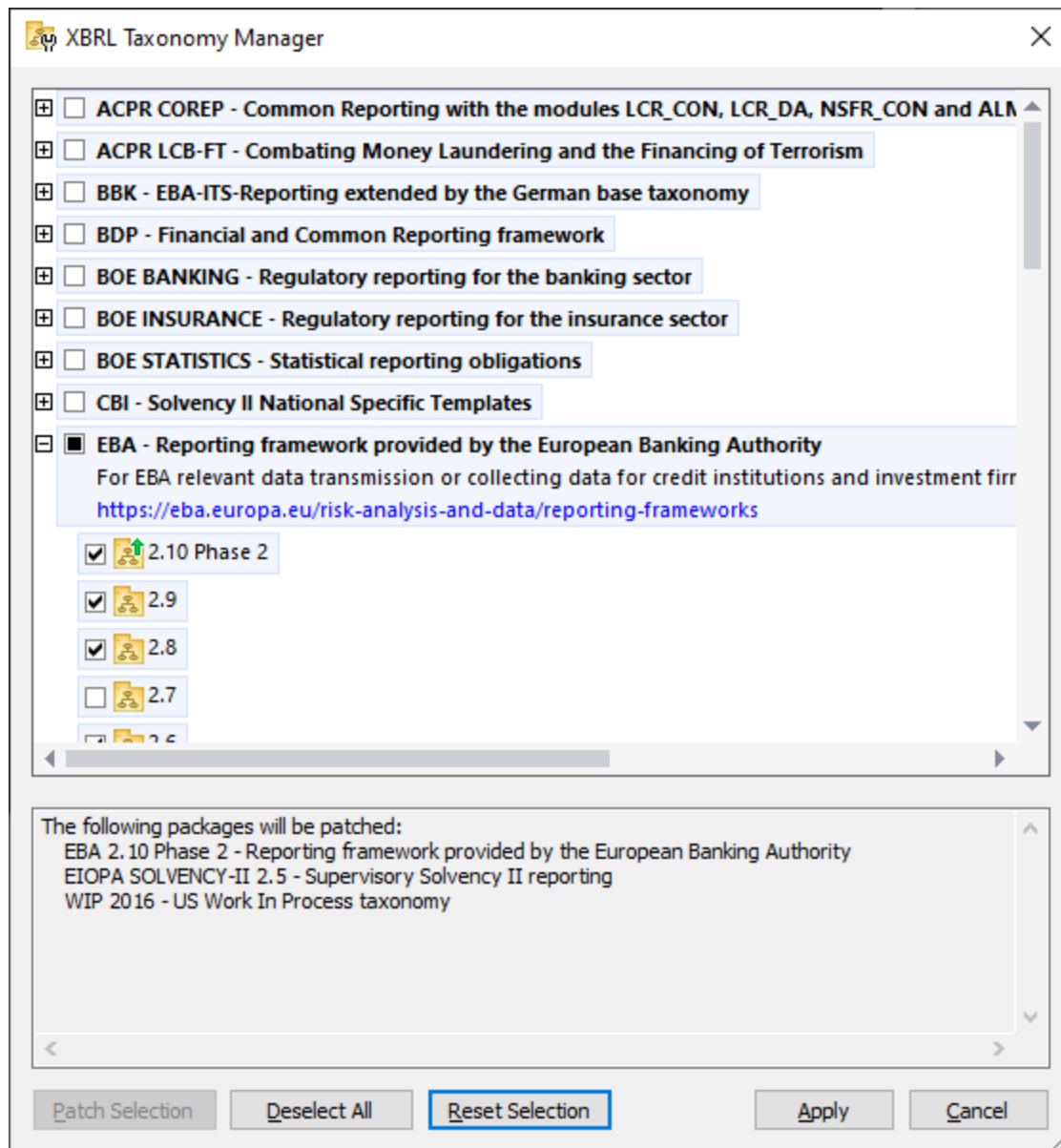
パッチの適用

不定期にXBRL タクソノミは発行元からパッチを受け取ります。XBRL タクソノミマネジャーがパッチが使用可能であることを検知すると、以下が表示されます:

- Windows の視覚的なユーザーインターフェイスを介してXBRL タクソノミマネジャーを使用すると、対応するXBRL タクソノミは  アイコンと共に表示されます。
- コマンドラインまたはLinux/macOS システムを使用すると、`list -u` コマンドと共に実行可能ファイルを実行する際にリストされているパッチを持つXBRL タクソノミがリストされます。

Windows 上でパッチを適用する方法:

1. [パッチの選択] ボタンをクリックします。  から  への変更を修飾する各 XBRL タクソノミのアイコンとダイアログボックスは適用されるパッチについて通知します。例:



メモ パッチの選択 ボタンは現在インストールされているXBRL タクソノミのため、パッチが使用可能な箇所でのみ有効化されています。

2. [適用]をクリックします。

コマンドラインインターフェイスでパッチを適用する方法:

1. `list -u` コマンドを実行します。パッチ更新が使用可能なタクソノミをリストします。
2. `upgrade` コマンドを実行して、パッチをインストールします。

更に詳しい情報に関しては [コマンドラインインターフェイス](#)。

7.3 インストール済みのタクソノミをビューする

現在インストールされているタクソノミを下にリストされるアプローチで適用可能な場合ビューすることができます。

グラフィカルなユーザーインターフェイス (Windows のみ)

視覚的なユーザーインターフェイスからインストール済みのすべてのタクソノミをビューするにはXBRL タクソノミ マネジャーを [実行](#) します。タクソノミ (または、タクソノミのバージョン) の横の選択済みのチェックボックスはそのタクソノミがインストール済みであることを示しています。

コマンドライン インターフェイス (Windows)

コマンドラインインターフェイスからすべての使用可能なタクソノミをビューするには、以下を実行します:

```
TaxonomyManager.exe list
```

インストール済みのタクソノミのみをビューするには、以下を実行します:

```
TaxonomyManager.exe list -i
```

新規のバージョンが使用可能なタクソノミのみをビューするには、以下を実行します:

```
TaxonomyManager.exe list -u
```

コマンドライン インターフェイス (Linux, macOS)

使用可能な全てのタクソノミをビューするには、以下を実行します:

```
sudo ./taxonomymanager list
```

インストール済みのタクソノミのみをビューするには、以下を実行します:

```
sudo ./taxonomymanager list -i
```

新規のバージョンが使用可能なタクソノミのみをビューするには、以下を実行します:

```
sudo ./taxonomymanager list -u
```

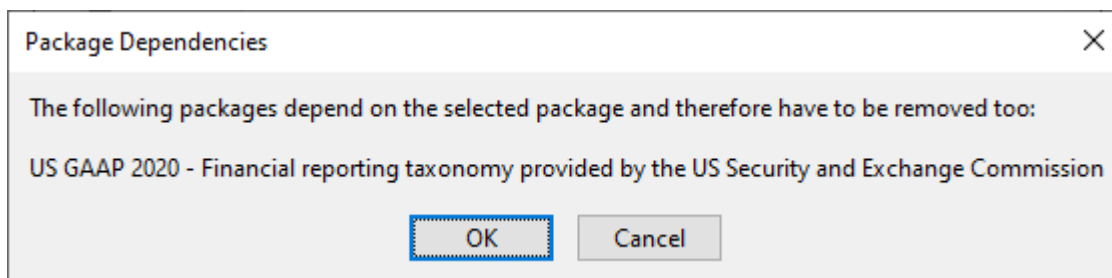
7.4 タクソノミのアンインストール

タクソノミを下にリストされるアプローチで適用可能な場合アンインストールすることができます。

グラフィカルなユーザーインターフェイス (Windows のみ)

タクソノミをアンインストールする方法

1. **実行** する。
2. アンインストールするタクソノミまたはタクソノミバージョンの横のチェックボックスのチェックを解除します。選択されたタクソノミが他のタクソノミパッケージに依存する場合、依存関係も削除されることを通知するダイアログボックスが開かれます。例:



3. [適用] をクリックしてタクソノミとその依存関係を削除します。

コマンドライン インターフェイス (Windows)

タクソノミをアンインストールするには、以下を実行します:

```
TaxonomyManager.exe uninstall FILTER...
```

FILTER が以下を意味する箇所:

1. フォーマット `<name>-<version>` 内のタクソノミ識別子の例: **eba-2.10**、**us-gaap-2020.0**。使用可能なすべてのタクソノミ識別子をビューするには `list` コマンドを実行します。
2. Altova Web サイトからダウンロードされた **altova_taxonomies** ファイル。

コマンドライン インターフェイス (Linux, macOS)

パッケージをアンインストールするには、以下を実行します:

```
sudo ./taxonomymanager uninstall FILTER...
```

FILTER が以下を意味する箇所:

1. フォーマット `<name>-<version>` 内のタクソノミ識別子の例: **eba-2.10**、**us-gaap-2020.0**。使用可能なすべてのタクソノミ識別子をビューするには `list` コマンドを実行します。
2. Altova Web サイトからダウンロードされた **altova_taxonomies** ファイル。

7.5 コマンドラインインターフェイス

XBRL タクソノミ マネジャーをコマンドラインで呼び出すには場合実行可能ファイルのパスを知っている必要があります。デフォルトでXBRL タクソノミ マネジャー 実行可能ファイルを以下のパスで見つけることができます:

Linux*	/opt/Altova/MapForceServer2021/bin/taxonomymanager
macOS*	/usr/local/Altova/MapForceServer2021/bin/taxonomymanager
Windows	C:\ProgramData\Altova\SharedBetweenVersions\TaxonomyManager.exe

* Linux と macOS パスをこれらのオペレーティングシステム上でXBRL タクソノミ マネジャーを MapForce Server、StyleVision Server、または RaptorXML+XBRL Server などの Altova クロスプラットフォームサーバーと共に実行している場合に適用可能です。

規則としては、このドキュメントは特定のコマンドを説明する際に実行可能ファイルのフルパスを省略し、実行可能ファイルの名前を使用する代わりに **<exec>** を使用します。例:

```
<exec> help
```

<exec> は実行可能ファイルのパスまたは名前です。

XBRL タクソノミ マネジャーをインストールディレクトリから呼び出す方法

実行可能ファイルをフルパスを入力することなく呼び出すには、現在のディレクトリを以下の一つに変更します:

Linux	cd /opt/Altova/MapForceServer2021/bin
macOS	cd /usr/local/Altova/MapForceServer2021/bin
Windows	cd "C:\ProgramData\Altova\SharedBetweenVersions"

相対的なパスを使用してコマンドを実行し実行可能ファイルを呼び出すことができます。

Linux	sudo ./taxonomymanager help
macOS	sudo ./taxonomymanager help
Windows	TaxonomyManager.exe help

メモ Linux と macOS システム上では、プレフィックス **./** は実行可能ファイルが現在のディレクトリ内にあることを示しています。**sudo** はコマンドがルート特権と共に実行されなければならないことを示しています。

XBRL タクソノミ マネジャーをディレクトリから呼び出す方法

実行可能ファイルをディレクトリから呼び出すには絶対パスを使用して参照してください。または、プログラムを実行可能ファイル名を入力するだけで呼び出す場合、使用中のオペレーティングシステムの PATH 環境変数を編集することにより、実行可能ファイルのディレクトリへのフルパスが含まれるようになります。PATH 環境変数を変更する方法に関しては、使用中のオペレーティングシステムのドキュメンテーションを参照

してください。

メモ

- PATH 環境変数の変更後、ターミナルウィンドウを閉じ、変更の効果が反映されるように新規のウィンドウを開く必要がある場合があります。
- Linux と macOS 上では `sudo` の使用はユーザーの PATH を考慮してません。

コマンドライン構文

コマンドラインを使用するための一般的な構文は以下の通りです:

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

上のリスト内で垂直バー | は相互に排他的なアイテムを分割します。角かっこは [] は任意のアイテムを示しています。基本的には --h, --help、または --version オプションまたはコマンドが後に続く実行可能ファイルパスを入力することができます。各コマンドはオプションと引数を持つ場合があります。コマンドのリストは以下のセクションで説明されています:

7.5.1 help

このコマンドはXBRL タクソノミ マネジャー 実行可能ファイルに関連するコマンドに関するコンテキストのヘルプを提供します。

構文

```
<exec> help [command]
```

[command] が有効なコマンド名を指定するオプションの引数である箇所。

リマーク

-h または --help が後に続くコマンドを入力してヘルプを呼び出すことができます。例:

```
<exec> list -h
```

実行可能ファイルの後に -h または --help を入力して一般的なヘルプを呼び出すことができます。例:

```
<exec> --help
```

サンプル

次のコマンドは list コマンドに関するヘルプを表示します:

Linux、macOS	./taxonomymanager help list
Windows	TaxonomyManager.exe help list

7.5.2 info

このコマンドは引数として与えられたタクソノミのそれぞれの詳細を表示します。詳細にはタイトル、バージョン、詳細、後悔元、依存レファレンスが含まれています。

構文

```
<exec> info [options] FILTER...
```

複数のタクソノミを指定するには `FILTER` を必要な回数繰り返します。

引数

FILTER	名前内にこの文字列を含むタクソノミをリストします (例えば <code>eba</code> または <code>us-gaap-2020.0</code>)。この引数を複数回指定することができます。 識別子のフルリストをビューするには <code>list</code> コマンドを実行します。
--------	--

オプション

--help, --h	コマンドラインでこのコマンドに関するヘルプを表示します。
-------------	------------------------------

例

以下のコマンドは `eba-2.10` と `us-gaap-2020.0` タクソノミに関する詳細を表示します:

Linux, macOS	./taxonomymanager info eba-2.10 us-gaap-2020.0
Windows	TaxonomyManager.exe info eba-2.10 us-gaap-2020.0

7.5.3 initialize

このコマンドはXBRL タクソノミ マネジャー 環境を初期化します。タクソノミに関するすべての情報が保管されている個所でキャッシュディレクトリを作成します。XBRL タクソノミ マネジャーを含む Altova アプリケーションをインストールすると初回に初期化が実行されるため、通常このコマンドを実行する必要はありません。reset コマンドを実行後に通常このコマンドを実行する必要があります。

構文

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

このコマンドのエイリアスは `init` です。

オプション

--help, --h	コマンドラインでこのコマンドに関するヘルプを表示します。
-------------	------------------------------

<code>--silent, --s</code>	エラーメッセージのみを表示します。デフォルトの値は false です。
<code>--verbose, --v</code>	実行中の追加情報を表示します。デフォルトの値は false です。

7.5.4 install

このコマンドはつまたは複数のタクソノミをインストールします。以下の点に注意してください！

- XBRL タクソノミ マネジャー からタクソノミをインストールまたはアンインストールすると同じコンピューター上のすべてのユーザーアカウントに影響を及ぼします。
- XBRL タクソノミ マネジャー からタクソノミをインストールまたはアンインストールすると同じコンピューター上にインストールされている Altova XBRL-有効化済みのすべてのアプリケーションに影響を与えます。
- 現在のタクソノミは他のタクソノミ上に依存関係があり、依存するタクソノミもインストールされます（または適用可能な場合アンインストールされます）。

構文

```
<exec> install [options] FILTER...
```

複数のタクソノミを指定するには **FILTER** を必要な回数繰り返します。

引数

FILTER	FILTER が以下を意味する箇所: 1. フォーマット <name>-<version> 内のタクソノミ識別子の例: eba-2.10、us-gaap-2020.0。使用可能なすべてのタクソノミ識別子をビューするには list コマンドを実行します。 2. Altova Web サイトからダウンロードされた altova_taxonomies ファイル。 一意の場合省略された識別子を使用することができます、例えば eba。省略された識別子を使用する場合、そのタクソノミの使用可能な最新バージョンがインストールされます。
---------------	--

オプション

<code>--help, --h</code>	コマンドラインでこのコマンドに関するヘルプを表示します。
<code>--silent, --s</code>	エラーメッセージのみを表示します。デフォルトの値は false です。
<code>--verbose, --v</code>	実行中の追加情報を表示します。デフォルトの値は false です。

サンプル

以下のコマンドは最新の **eba**（欧州銀行監督局）と **us-gaap**（米国会計基準）タクソノミをインストールします:

Linux, macOS	<code>./taxonomymanager install eba us-gaap</code>
Windows	<code>TaxonomyManager.exe install eba us-gaap</code>

7.5.5 list

このコマンドを使用して以下の方法の1つによりコマンドラインでタクソノミをリストします。

- 使用可能なタクソノミを全てリストします
- 特定のタクソノミをリストします
- インストール済みのタクソノミのみをリストします
- アップグレードが必要なタクソノミのみをリストします

構文

```
<exec> list [options] [FILTER...]
```

このコマンドは `ls` を使用して省略することができます。

引数

FILTER	名前内にこの文字列を含むタクソノミのみをリストします。この引数を複数回指定することができます。
--------	---

オプション

--help, --h	コマンドラインでこのコマンドに関するヘルプを表示します。
--installed, --i	インストール済みのタクソノミのみをリストします。デフォルトの値は false です。
--upgradeable, --u	パッチ更新が使用可能なタクソノミのみをリストします。デフォルトの値は false です。

例

使用可能な全てのタクソノミをリストするには、以下を実行します:

Linux、macOS	<code>./taxonomymanager list</code>
Windows	<code>TaxonomyManager.exe list</code>

インストール済みのタクソノミのみをリストするには、以下を実行します:

Linux、macOS	<code>./taxonomymanager list -i</code>
Windows	<code>TaxonomyManager.exe list -i</code>

名前内に "eba" または "us-gaap" を含むすべてのタクソノミをリストするには、以下を実行します:

Linux、macOS	<code>./taxonomymanager list eba us-gaap</code>
Windows	<code>TaxonomyManager.exe list eba us-gaap</code>

7.5.6 reset

このコマンドはすべてのインストールされたタクソノミとキャッシュディレクトリを削除します。

警告: このコマンドはすべてのインストールされたタクソノミとその情報を削除します。

このコマンドの実行後、キャッシュディレクトリを再作成するために `initialize` コマンドを実行してください。または `reset` コマンドを `-i` オプションと共に実行してください。

`reset -i` は製品オリジナルのインストールを復元するためのリセット後に更に `update` コマンドを実行することが奨励されます。または、`-i` と `-u` オプションを使用してコマンドをリセットします。

構文

```
<exec> reset [options]
```

オプション

<code>--help, --h</code>	コマンドラインでこのコマンドに関するヘルプを表示します。
<code>--init, --i</code>	XBRL タクソノミマネジャー環境をリセット後初期化してください。有効な値は true または false です。デフォルトの値は false です。
<code>--silent, --s</code>	エラーメッセージのみを表示します。デフォルトの値は false です。
<code>--update, --u</code>	XBRL タクソノミマネジャー環境をリセット後に初期化し更新してください。有効な値は true または false です。デフォルトの値は false です。
<code>--verbose, --v</code>	実行中の追加情報を表示します。デフォルトの値は false です。

例

XBRL タクソノミ マネジャーをリセットするには、以下を実行します:

<i>Linux, macOS</i>	<code>./taxonomymanager reset</code>
<i>Windows</i>	<code>TaxonomyManager.exe reset</code>

7.5.7 setdeflang

このコマンドはXBRL タクソノミ マネジャー の言語を設定します。

構文

```
<exec> setdeflang language
```

`language` が言語コードを提供する必須の引数の箇所。このコマンドのエイリアスは `sdl` です。

引数

language	設定する言語有効な言語は英語(en)、フランス語(fr)、ドイツ語(de)、日本語(ja)、およびスペイン語(es)です。
----------	---

オプション

--help, --h	コマンドラインでこのコマンドに関するヘルプを表示します。
-------------	------------------------------

サンプル

言語をスペイン語に設定する場合は、以下を実行します:

Linux, macOS	./taxonomymanager setdeflang es
Windows	TaxonomyManager.exe setdeflang es

7.5.8 uninstall

このコマンドは一つまたは複数のタクソノミをアンインストールします。デフォルトでは現在のタクソノミにより参照されるタクソノミもアンインストールされます。現在のタクソノミだけをアンインストールし、参照されたタクソノミを保つ場合はオプション `--k` を設定します。

構文

<code><exec> uninstall FILTER...</code>

複数のタクソノミを指定する場合は `FILTER` を必要な回数繰り返します。

引数

FILTER	FILTER が以下を意味する箇所: 1. フォーマット <code><name>-<version></code> 内のタクソノミ識別子の例: eba-2.10、us-gaap-2020.0。使用可能なすべてのタクソノミ識別子をビューするには <code>list</code> コマンドを実行します。 2. Altova Web サイトからダウンロードされた altova_taxonomies ファイル。
--------	--

オプション

--help, --h	コマンドラインでこのコマンドに関するヘルプを表示します。
--keep-references, --k	このオプションが設定されると、参照されたタクソノミはアンインストールされません。デフォルトの値は false です。
--silent, --s	エラーメッセージのみを表示します。デフォルトの値は false です。
--verbose, --v	実行中の追加情報を表示します。デフォルトの値は false です。

サンプル

以下のコマンドは **eba-2.10** と **us-gaap-2020.0** タクソノミとその依存関係をアンインストールします:

Linux, macOS	<code>./taxonomymanager uninstall eba-2.10 us-gaap-2020.0</code>
Windows	<code>TaxonomyManager.exe uninstall eba-2.10 us-gaap-2020.0</code>

7.5.9 update

このコマンドはオンラインストレージから使用可能なタクソノミのリストをフェッチ、ローカルのキャッシュディレクトリを更新します。この情報の更新は明示的に行われ、`reset` と `initialize` を実行した場合を除いてこのコマンドを実行する必要はありません。

構文

```
<exec> update [options]
```

オプション

<code>--help, --h</code>	コマンドラインでこのコマンドに関するヘルプを表示します。
<code>--silent, --s</code>	エラーメッセージのみを表示します。デフォルトの値は false です。
<code>--verbose, --v</code>	実行中の追加情報を表示します。デフォルトの値は false です。

サンプル

以下のコマンドはタクソノミの情報を明示的に更新します。

Linux, macOS	<code>./taxonomymanager update</code>
Windows	<code>TaxonomyManager.exe update</code>

7.5.10 upgrade

このコマンドは全ての対象のタクソノミを使用可能な最新 *patch* バージョンにアップグレードします。すなわち、特定のリリースの *patch* レベルでのみ更新を行います。このコマンドを実行することは使用可能なタクソノミを更新する場合のみ意味があります。`list -u` コマンドを実行して更新可能なタクソノミを識別することができます。

構文

```
<exec> upgrade [options]
```

オプション

<code>--help, --h</code>	コマンドラインでこのコマンドに関するヘルプを表示します。
--------------------------	------------------------------

--silent, --s	エラーメッセージのみを表示します。デフォルトの値は false です。
--verbose, --v	実行中の追加情報を表示します。デフォルトの値は false です。

8 カタログファイル

MapForce Server は OASIS XML カタログメカニズムのサブセットをサポートします (<https://www.oasis-open.org/committees/entity/spec-2001-08-06.html>)。カタログメカニズムにより MapForce Server が共通に使用される DTD と XML スキーマ (およびスタイルシートと他のファイル) を、ブロック URI から解決する代わりにローカルフォルダから抽出することができるようになります。これは全体的な処理スピードを増やしオフライン (すなわちネットワークに接続されていない状態) での作業を可能にし、ドキュメントのポータビリティを改善します。

カタログのしくみ

カタログは一般的に、ブロック DTD および通常は、ローカルファイル (すなわち) ローカル URI へのスキーマ参照をリダイレクトするために使用されます。これを達成するために、XML フォーマットのカタログファイルは、ブロックスキーマ URI とローカル URI の間のマッピングを定義します。MapForce Server が XML ドキュメントを解析する都度、カタログファイル内部のスキーマ URI (または、適用可能な場合、DTD のブロックおよびシステム識別子) が最初に検索されます。マッピングがカタログファイル内で見つかる場合、その参照が使用されるスキーマはローカルファイルから読み取られます。カタログファイル内でマッピングが見つからない場合、XML ドキュメントの URI はそのまま解決されます。

例えば、以下の XML ファイルは MapForce Server により処理される必要があると仮定します。

```
<?xml version="1.0" encoding="UTF-8"?>
<Articles xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="Articles.xsd">
  <Article>
    <Number>1</Number>
    <Name>T-Shirt</Name>
    <SinglePrice>25</SinglePrice>
  </Article>
</Articles>
```

(MapForce Server が既知の通り) **catalog.xml** ファイルがローカルディレクトリに存在し、以下のテンプレートが含まれると想定します。この点については以下で詳しく説明されています。

```
<catalog>
  <!-- ... -->
  <uri name="http://www.w3.org/2001/XMLSchema-instance.xsd" uri="files/XMLSchema-
instance.xsd"/>
  <!-- ... -->
</catalog>
```

XML ファイルを解析する場合、MapForce Server はカタログファイル内でスキーマ参照の `http://www.w3.org/2001/XMLSchema-instance.xsd` のための一致を検知します。この結果スキーマはカタログファイルに対して相対的なローカル (すなわち) `files/XMLSchema-instance.xsd` からロードされます。カタログファイル内でマッピングが見つからない場合、スキーマは `http://www.w3.org/2001/XMLSchema-instance` からロードされます。

ルートカタログ

MapForce Server が開始されると、**RootCatalog.xml** と呼ばれるファイルを `%ProgramFilesFolder%\Altova\MapForceServer2021\etc` ディレクトリからロードします。**RootCatalog.xml** はそれぞれが `nextCatalog` 要素内にあるカタログファイルのリストを含んでいます。これらのカタログファイルは、おレックアップされた内部にある URI は指定されているマッピングに従い MapForce Server により解決されます。

```
<?xml version="1.0" encoding="UTF-8"?>
<catalog xmlns="urn:oasis:names:tc:entity:xmlns:xml:catalog"
xmlns:spy="http://www.altova.com/catalog_ext"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oasis:names:tc:entity:xmlns:xml:catalog
Catalog.xsd">
  <nextCatalog catalog="CustomCatalog.xml"/>
  <nextCatalog catalog="CoreCatalog.xml"/>
  <!-- Include all catalogs under common schemas folder on the first directory level
-->
  <nextCatalog spy:recurseFrom="Schemas" catalog="catalog.xml" spy:depth="1"/>
</catalog>
```

RootCatalog.xml

上記のリスティングで、以下のカタログがリンクアップのためリストされていることに注意してください

- **CustomCatalog.xml** は自身のマッピングを作成することができるファイルです。このファイルは %ProgramFilesFolder%\Altova\MapForceServer2021\etc ディレクトリ内に存在します。Altova 構成済みのカタログファイルで既にカスタムスキーマのために対処されていない場合 (次のポイントを参照してください) CustomCatalog.xml にマッピングを追加することができます。
- **CoreCatalog.xml** はスキーマをロケートするための Altova 固有のマッピングが含まれています。このファイルは %ProgramFilesFolder%\Altova\MapForceServer2021\etc ディレクトリ内に存在します。
- **Schemas** ディレクトリからの RootCatalog.xml ファイルに相対的な複数の catalog.xml ファイル。各 catalog.xml ファイルは (SVG、DITA、DocBook、WSDL などの) 特定のスキーマのディレクトリの内部にあり、それぞれは対応するスキーマのローカルで保存されたコピーを指す URI へのブロックおよびおのシステム識別子をマップします。

以下の点に注意してください

- **CustomCatalog.xml** を変更する場合 [サポートされる要素](#) を使用してください。エラーを引き起こす可能性があるため既存のマッピングを複製しないでください。

サポートされる要素

CustomCatalog.xml 内でエントリを作成するには、以下にリストされる要素のみを使用して下さい。OASIS XML カタログ仕様の他の要素はサポートされていません。

要素	属性	例
public	- systemId はリソースの ブロック識別子を指定します。 - uri URI レファレンスを指定します (例えば、ローカルディレクトリに対する相対的なパス)	<pre><public publicId="-//W3C//DTD XMLSCHEMA 200102//EN" uri="files/XMLSchema.dtd"/></pre>
system	- systemId はリソースのシステム識別子を指定します。 - uri URI レファレンスを指定します (例えば、ローカルディレクトリに対する相対的なパス)	<pre><system systemId="http://www.w3.org/2009 /XMLSchema/datatypes.dtd" uri="files/datatypes.dtd"/></pre>
uri	- name は URI レファレンスを指定します - Uri は (例えば、ローカルファイルに対する相対的なパスなど) 代替 URI レファレンス	<pre><uri name="http://www.w3.org/2009/XML Schema/XMLSchema.xsd"</pre>

要素	属性	例
	を指定します	<code>uri="files/XMLSchema.xsd"/></code>
rewriteURI	<code>uriStartString</code> は再書き込みする URI の開始部分を指定します。 <code>Uri</code> 置換文字列を指定します (例えば ローカルディレクトリに対する相対的なパス)	<pre><rewriteURI uriStartString='http://www.altova.com/schemas/svg/' rewritePrefix='files/' /></pre>
rewriteSystem	<code>systemIdStartString</code> は再書き込みするシステムの識別子の開始部分を指定します。 <code>rewritePrefix</code> 置換文字列を指定します (例えば ローカルディレクトリに対する相対的なパス)	<pre><rewriteSystem systemIdStartString='http://www.altova.com/schemas/svg/' rewritePrefix='files/' /></pre>

public、**system**、および **uri** 要素は相対的な URI が解決されるベース URI を指定する `xml:base` 属性を取ることができます。詳細に関しては XML カタログ仕様 (<http://www.oasis-open.org/committees/entity/spec-2001-08-06.html>) を参照してください。

インデックス

A

`assignlicense`,
CLI コマンドとして, 27

C

C#,
サンプル, 39
マッピングの実行, 39

C++,
サンプル, 45
マッピングの実行, 45

CLI コマンド,
`assignlicense`, 27
`exportresourcestrings`, 28
`help`, 29
`licenseserver`, 30
`run`, 31
`setdeflang`, 34
`verifylicense`, 35, 36

E

`exportresourcestrings`,
CLI コマンドとして, 28

F

FlowForce Server, 17

H

`help`,
CLI コマンドとして, 29

HTTPS,
Web サービスの呼び出し, 92

J

Java,
MapForce Server の呼び出し, 55
MapForce Server を CLASSPATH に追加, 53
サンプル, 55

L

`licenseserver`,
CLI コマンドとして, 30

Linux,
HTTPS を介して Web
サービスを使用してマッピングを実行する, 104
MapForce Server のインストール, 11
MapForce Server をアンインストールする方法, 12
クライアント証明書の転送, 104
サポート情報, 8
現在インストールされている Altova 製品の確認, 12
信頼できるサーバー証明書, 106

M

Mac,
HTTPS を介して Web
サービスを使用してマッピングを実行する, 105
クライアント証明書の転送, 105
信頼できるサーバー証明書, 108

macOS,
MapForce Server のインストール, 13
MapForce Server をアンインストールする方法, 14
サポート情報, 8

MapForce Server API,
.NET のための, 38
C# サンプル, 39
C++ サンプル, 45
COM のための, 45
Java のための, 53
Java サンプル, 55
VB.NET サンプル, 41
VBA サンプル, 50

MapForce Server API,
 VBScript サンプル, 47

S

setdeflang,
 CLI コマンドとして, 34

V

VB.NET,
 サンプル, 41
 マッピングの実行, 41

VBA,
 サンプル, 50
 マッピングの実行, 50

VBScript,
 サンプル, 47
 マッピングの実行, 47

verifylicense,
 CLI コマンドとして, 35

version,
 CLI コマンドとして, 36

W

Web サービス,
 HTTPS を使用した呼び出し, 92

Windows,
 HTTPS を介して Web
 サービスを使用してマッピングを実行する, 110
 MapForce Server をアンインストールする方法, 16
 MapForce Server のインストール, 15
 サポート情報, 8
 信頼できるサーバー証明書, 109

X

XML カタログ,
 しくみ, 130
 構成, 130

Z

グローバルリソース,
 マッピング内で使用, 24
 定義, 24

コマンドラインインターフェイス,
 使用方法, 26

マッピング,
 C# と実行, 39
 C++ と実行, 45
 FlowForce Server にデプロイ, 17
 Java と作業, 55
 VB.NET を使用して実行, 41
 VBA と実行, 50
 VBScript を使用して実行, 47
 コマンドラインで実行, 31
 実行可能ファイルのコンパイル, 17

実行,
 CLI コマンドとして, 31

電子証明書,
 Linux への移行, 104
 Linux 上で信頼できる, 106
 Mac に転送する, 105
 Mac 上で信頼する, 108
 MapForce マッピング内, 92
 Windows からエクスポートする, 98
 Windows 上での管理, 95
 Windows 上で信頼できる, 109