

Altova MapForce Server 2025 Advanced Edition



Benutzer- und Referenzhandbuch

Altova MapForce Server 2025 Advanced Edition Benutzer- und Referenzhandbuch

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

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

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

Published: 2025

© 2019-2025 Altova GmbH

Inhaltsverzeichnis

1	Einführung	6
2	Installation und Lizenzierung	8
2.1	Einrichten unter Windows.....	9
2.1.1	Installation unter Windows.....	9
2.1.2	Installation auf Windows Server Core.....	10
2.1.3	Installation von LicenseServer (Windows).....	12
2.1.4	Starten von LicenseServer, MapForce Server (Windows).....	13
2.1.5	Registrieren von MapForce Server (Windows).....	14
2.1.6	Zuweisen einer Lizenz (Windows).....	15
2.2	Einrichten unter Linux.....	17
2.2.1	Installation unter Linux.....	17
2.2.2	Installation von LicenseServer (Linux).....	19
2.2.3	Starten von LicenseServer, MapForce Server (Linux).....	20
2.2.4	Registrieren von MapForce Server (Linux).....	20
2.2.5	Zuweisen einer Lizenz (Linux).....	21
2.3	Einrichten auf macOS.....	23
2.3.1	Installation auf macOS.....	23
2.3.2	Installation von LicenseServer (macOS).....	24
2.3.3	Starten von LicenseServer, MapForce Server (macOS).....	25
2.3.4	Registrieren von MapForce Server (macOS).....	25
2.3.5	Zuweisen einer Lizenz (macOS).....	26
2.4	Upgraden von MapForce Server.....	28
2.5	Migrieren von MapForce Server auf einen neuen Rechner.....	29
2.6	Sicherheitsaspekte.....	30
3	Ausführen von Mappings	31
3.1	Vorbereiten von Mappings für die Server-Ausführung.....	34

3.2	Globale Ressourcen.....	40
3.3	Join-Optimierung.....	42
3.4	Anmeldeinformationen.....	45
3.4.1	Beispiel: OAuth 2.0-Autorisierung.....	47
3.5	Dynamische Authentifizierung.....	62

4 MapForce Server-Befehlszeile 63

4.1	accepteula (nur Linux).....	66
4.2	assignlicense (nur Windows).....	67
4.3	exportresourcestrings.....	68
4.4	help	70
4.5	licenseserver.....	71
4.6	run	72
4.7	setdeflang.....	76
4.8	verifylicense.....	77
4.9	version.....	78

5 MapForce Server API 79

5.1	.NET-Schnittstelle.....	80
5.1.1	C#-Beispiel.....	81
5.1.2	Visual Basic .NET-Beispiel.....	84
5.2	COM-Schnittstelle.....	88
5.2.1	C++-Beispiel.....	88
5.2.2	VBScript-Beispiel.....	91
5.2.3	VBA-Beispiel.....	94
5.3	Java-Schnittstelle.....	97
5.3.1	Java-Beispiel.....	101
5.4	Beispiel: Ausführen eines Mappings mit Parametern.....	104
5.5	API-Referenz (COM, .NET).....	110
5.5.1	Schnittstellen.....	110
5.6	API-Referenz (Java).....	128
5.6.1	Klassen.....	128

6	Verwaltung digitaler Zertifikate	142
6.1	Definieren vertrauenswürdiger Zertifikate unter Linux.....	145
6.2	Definieren vertrauenswürdiger Zertifikate auf macOS-Systemen.....	146
6.3	Definieren vertrauenswürdiger Zertifikate unter Windows.....	147
6.4	Aufrufen der Zertifikatspeicher unter Windows.....	148
6.5	Exportieren von Zertifikaten aus Windows.....	149
6.6	Client-Zertifikate unter Linux.....	155
6.7	Client-Zertifikate auf macOS.....	157
6.8	Client-Zertifikate unter Windows.....	159
7	Taxonomie-Manager	161
7.1	Ausführen des Taxonomie-Managers.....	165
7.2	Statuskategorien.....	168
7.3	Anwenden eines Patch oder Installation einer Taxonomie.....	170
7.4	Deinstallieren einer Taxonomie, Zurücksetzen.....	172
7.5	Befehlszeilenschnittstelle (CLI).....	173
7.5.1	help	173
7.5.2	info	174
7.5.3	initialize.....	174
7.5.4	install	175
7.5.5	list	176
7.5.6	reset	176
7.5.7	uninstall.....	177
7.5.8	update.....	178
7.5.9	upgrade.....	179
8	Katalogdateien	180
	Index	181

1 Einführung

MapForce Server ist eine Software-Lösung für Unternehmen zur Ausführung von Datenmappingtransformationen auf Windows-, Linux- und macOS-Betriebssystemen. Die Datenmappings (oder Mapping-Design-Dateien, *.mfd) selbst werden auf grafischem Weg mit Altova MapForce (<https://www.altova.com/de/mapforce.html>) erstellt. Mit Altova MapForce werden die Inputs, Outputs sowie etwaige Zwischenverarbeitungsschritte für Ihre Daten definiert. Mit MapForce Server werden mit MapForce kompilierte MapForce Server-Ausführungsdateien (.mfx-Dateien) ausgeführt und je nach zugrunde liegendem Mapping Ausgabedateien oder Daten erzeugt oder sogar Datenbankaktualisierungen durchgeführt oder Webservices aufgerufen.



MapForce Server kann als eigenständiges Programm oder unter Verwaltung von Altova FlowForce Server (<https://www.altova.com/de/flowforce.html>) ausgeführt werden. Wenn FlowForce Server auf demselben Rechner wie MapForce Server installiert ist, automatisiert FlowForce Server die Ausführung von Mappings über geplante oder Trigger-basierte Aufträge, die auch als Webservices bereitgestellt werden können. Zusätzlich dazu enthält FlowForce Server eine Bibliothek von vordefinierten Funktionen, mit denen vor oder nach der Mapping-Ausführung weitere automatisierte Aktionen wie z.B. das Senden von E-Mails, das Kopieren von Dateien und Verzeichnissen, das Hochladen von Dateien auf FTP, die Ausführung von Shell-Befehlen und andere Aktionen durchgeführt werden können.

Funktionalitäten

- Ausführung von Datenmappings auf Hochleistungsservern
- Plattformübergreifender Einsatz: MapForce Server kann auf Windows-, Linux- oder macOS-Betriebssystemen ausgeführt werden
- Befehlszeilenschnittstelle
- eine API, die von C++, C#, Java-, VB.NET-, VBScript- oder VBA-Code aus aufgerufen werden kann
- Native Integration mit FlowForce Server.
- Unterstützung für globale Altova Ressourcen. Dadurch können Datei-, Ordner- und Datenbankreferenzen konfigurierbar und für mehrere Umgebungen und Altova-Applikationen portierbar gemacht werden, siehe [globale Altova Ressourcen](#) ⁴⁰
- Beschleunigte Ausführung von Mappings, wobei eine Join-Optimierung unterstützt wird (siehe [Join-Optimierung](#) ⁴²).
- Ausführung von Mappings, in denen Funktionen und Standardwerte auf mehrere Datenelemente gleichzeitig angewendet werden. Mit solchen Mappings können z.B. alle gefundenen Nullwerte einfach durch leere Strings oder benutzerdefinierten Text ersetzt werden.
- Ausführung von Mappings, bei denen Daten aus Protocol Buffers-Binärdateien ausgelesen und in diese geschrieben werden.
- Ausführung von Mappings, die Datenbank-Bulk-Einfügungen durchführen.

Einschränkungen

- digitale XML-Signaturen werden nicht unterstützt
- ADO-, ADO.NET- und ODBC-Datenbankverbindungen werden nur von Windows unterstützt. Unter Linux und macOS stehen native Datenbankverbindungen für SQLite und PostgreSQL zur Verfügung. Für andere unter Linux oder macOS ausgeführte Datenbanken sollte JDBC verwendet werden.

Letzte Aktualisierung: 20.03.2025

2 Installation und Lizenzierung

In diesem Abschnitt finden Sie Anleitungen zum Installieren, Lizenzieren und Konfigurieren. Es ist in die folgenden Abschnitte gegliedert:

- [Einrichten unter Windows](#) ⁹
- [Einrichten unter Linux](#) ¹⁷
- [Einrichten auf macOS](#) ²³
- [Upgraden von MapForce Server](#) ²⁸
- [Migrieren von MapForce Server auf einen neuen Rechner](#) ²⁹

2.1 Einrichten unter Windows

In diesem Abschnitt werden die [Installation](#)⁹ und Lizenzierung von MapForce Server auf Windows-Systemen beschrieben. Bei der Einrichtung müssen die folgenden Schritte durchgeführt werden:

1. [Installation von MapForce Server](#)⁹
2. [Installation von LicenseServer](#)¹²
3. [Starten von LicenseServer und MapForce Server](#)¹³
4. [Registrieren von MapForce Server auf LicenseServer](#)¹⁴
5. [Zuweisen einer Lizenz zu MapForce Server](#)¹⁵

Die einzelnen oben beschriebenen Schritte müssen nicht in genau derselben Reihenfolge wie aufgelistet erfolgen. Bevor Sie beginnen, muss jedoch zunächst die Installation durchgeführt werden. Außerdem muss MapForce Server auf LicenseServer registriert werden, bevor Sie MapForce Server über LicenseServer eine Lizenz zuweisen können.

Systemvoraussetzungen (Windows)

Beachten Sie die folgenden Systemvoraussetzungen:

- Windows 10, Windows 11
- Windows Server 2016 oder höher

Voraussetzungen

Beachten Sie dabei die folgenden Voraussetzungen:

- Führen Sie die Installation als Benutzer mit Admin-Rechten durch.
- Ab Version 2021 kann eine 32-Bit-Version von MapForce Server nicht über eine 64-Bit-Version oder eine 64-Bit-Version nicht über eine 32-Bit-Version installiert werden. Sie müssen (i) die ältere Version entweder vor der Installation der neueren Version entfernen oder (ii) auf eine neuere Version, die dieselbe Bit-Version wie Ihre ältere Version hat, aktualisieren.

2.1.1 Installation unter Windows

Installation von MapForce Server

MapForce Server kann auf Windows-Systemen folgendermaßen installiert werden:

- Als separates eigenständiges Server-Produkt. Um MapForce Server zu installieren, laden Sie das MapForce Server-Installationsprogramm herunter und starten Sie es. Befolgen Sie die Anweisungen auf dem Bildschirm.
- Um MapForce Server im Rahmen des [FlowForce Server](#)-Pakets zu installieren, laden Sie das FlowForce Server-Installationsprogramm herunter und starten Sie es. Befolgen Sie die Anweisungen auf dem Bildschirm und stellen Sie sicher, dass Sie die Option zur Installation von MapForce Server aktiviert haben.

Sowohl das Installationsprogramm von MapForce Server als auch das von [FlowForce Server](#) steht im Altova Download Center (<https://www.altova.com/de/download.html>) zur Verfügung. Sie können im Feld links unten im

Assistenten die Installationssprache auswählen. Beachten Sie, dass Sie damit auch die Standardsprache von MapForce Server definieren. Sie können die Sprache später über die Befehlszeile ändern.

Nach der Installation befindet sich die ausführbare MapForce Server-Datei standardmäßig unter dem folgenden Pfad:

<ProgramFilesFolder>\Altova\MapForceServer2025\bin\MapForceServer.exe

Alle erforderlichen Registrierungen zur Verwendung von MapForce Server über eine COM-Schnittstelle, als Java-Schnittstelle und in der :NET-Umgebung werden vom Installationsprogramm vorgenommen.

Deinstallieren von MapForce Server

Deinstallieren Sie MapForce Server folgendermaßen:

1. Klicken Sie mit der rechten Maustaste auf die Windows-Schaltfläche **Start** und wählen Sie **Einstellungen**.
2. Öffnen Sie die Systemsteuerung (Geben Sie die ersten Buchstaben davon ein und klicken Sie auf den Vorschlag).
3. Klicken Sie unter *Programme* auf **Programm deinstallieren**.
4. Wählen Sie in der Systemsteuerung MapForce Server aus und klicken Sie auf **Deinstallieren**.

Testlizenz

Sie haben während der Installation die Option, eine 30-Tage-Testlizenz für MapForce Server anzufordern. Die Testlizenz wird nach Erhalt der Anforderung an die von Ihnen registrierte E-Mail-Adresse gesendet.

2.1.2 Installation auf Windows Server Core

Windows Server Core ist eine Windows Minimalinstallation, bei der eine Reihe von Funktionen der Benutzeroberfläche nicht verwendet wird. Sie können MapForce Server folgendermaßen auf einem Windows Server Core-Rechner installieren:

1. Laden Sie die ausführbare MapForce Server-Installationsdatei von der Altova Website herunter. Diese Datei hat den Namen **MapForceServerAdv.exe**. Stellen Sie sicher, dass die ausführbare Datei zu Ihrer Server-Plattform (32-Bit oder 64-Bit) passt.
2. Führen Sie auf einem Standard-Windows-Rechner (und nicht dem Windows Server Core-Rechner) den folgenden Befehl aus: **MapForceServerAdv.exe /u**. Dadurch wird die **.msi**-Datei im selben Ordner entpackt, in dem sich auch die ausführbare Installationsdatei befindet.
3. Kopieren Sie die entpackte **.msi**-Datei auf den Windows Server Core-Rechner.
4. Wenn Sie eine frühere Version von MapForce Server aktualisieren, beenden Sie zuerst MapForce Server, bevor Sie den nächsten Schritt durchführen.
5. Verwenden Sie die **.msi**-Datei für die Installation, indem Sie den Befehl **msiexec /i MapForceServerAdvanced.msi** ausführen. Dadurch wird die Installation auf Windows Server Core gestartet.

Achtung: Bewahren Sie die .msi-Datei auf!

Beachten Sie die folgenden Punkte:

- Speichern Sie die extrahierte **.msi**-Datei an einem sicheren Ort. Sie benötigen diese später, um das Produkt zu deinstallieren, zu reparieren oder die Installation anzupassen.
- Wenn Sie die MSI-Datei umbenennen möchten, tun Sie das, bevor Sie MapForce Server installieren.
- Der Name der MSI-Datei wird in der Registry gespeichert. Sie können den Dateinamen dort aktualisieren, falls er geändert wurde.

Registrieren von MapForce Server auf LicenseServer

Wenn Sie MapForce Server zum ersten Mal installieren oder ein Upgrade auf eine **Hauptversion** installieren, müssen Sie MapForce Server auf einem Altova LicenseServer in Ihrem Netzwerk registrieren. Wenn Sie ein Upgrade auf eine Nicht-Hauptversion von MapForce Server installieren, kennt das Installationsprogramm die vorherige LicenseServer-Registrierung, daher muss MapForce Server nicht auf dem LicenseServer registriert werden. Wenn Sie den von MapForce Server verwendeten LicenseServer jedoch zu irgendeinem Zeitpunkt wechseln möchten, müssen Sie MapForce Server auf dem neuen LicenseServer registrieren.

Um MapForce Server bei der Installation auf einem Altova LicenseServer zu registrieren, führen Sie den Installationsbefehl mit der Eigenschaft **REGISTER_WITH_LICENSE_SERVER** aus, wie unten aufgelistet, und geben Sie den Namen oder die Adresse des LicenseServer-Rechners als Wert der Eigenschaft an, z.B:

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

Um MapForce Server nach der Installation auf einem Altova LicenseServer zu registrieren, starten Sie den folgenden Befehl:

```
msiexec /r MapForceServerAdvanced.msi REGISTER_WITH_LICENSE_SERVER="<MyLS-IPAddress>"
```

Nützliche Befehle

Im Folgenden finden Sie eine Reihe von Befehlen, die im Rahmen der Installation von Nutzen sind.

Den Rückgabewert der Installation können Sie mit einem Skript wie dem folgenden überprüfen. Der Rückgabecode befindet sich in der Umgebungsvariablen %errorlevel%. Der Rückgabecode **0** bedeutet, dass der Vorgang erfolgreich war.

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

Damit die Installation im Hintergrund, mit einem Rückgabecode und einem Log des Installationsvorgangs durchgeführt wird, führen Sie den folgenden Befehl aus:

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

Um die Installation zu ändern, führen Sie den folgenden Befehl aus:

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

Um die Installation zu reparieren, führen Sie den folgenden Befehl aus:

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

So deinstallieren Sie MapForce Server:

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

Um MapForce Server im Hintergrund zu deinstallieren und das Resultat in ein detailliertes Log zu schreiben, verwenden Sie folgenden Befehl:

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

Um MapForce Server in einer anderen Sprache zu installieren (die verfügbaren Sprachcodes sind: Deutsch=de; Spanisch=es; Französisch=fr), verwenden Sie:

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

Anmerkung: Auf Windows Server Core stehen die Diagramm-Funktionen von MapForce Server nicht zur Verfügung.

Anmerkung: Um Taxonomien zu installieren, verwenden Sie den Taxonomiepaket-Manager über die Befehlszeile. Nähere Informationen dazu finden Sie im MapForce Server-Benutzerhandbuch.

2.1.3 Installation von LicenseServer (Windows)

Damit MapForce Server ausgeführt werden kann, muss das Produkt über einen [Altova LicenseServer](#) in Ihrem Netzwerk lizenziert sein. Wenn Sie MapForce Server oder FlowForce Server auf Windows-Systemen installieren, können Sie LicenseServer zusammen mit MapForce Server oder FlowForce Server installieren. Wenn in Ihrem Netzwerk bereits ein LicenseServer installiert ist, muss kein weiterer installiert werden, es sei denn, Sie benötigen eine neuere Version von LicenseServer. (Siehe nächster Punkt, [LicenseServer-Versionen](#).)

Aktivieren Sie während der Installation von MapForce Server oder FlowForce Server je nach Bedarf die Option zum Installieren von LicenseServer.

Beachten Sie die folgenden Punkte:

- Wenn Sie Altova LicenseServer noch nicht installiert haben, belassen Sie die Standardeinstellungen unverändert. Der Assistent installiert daraufhin auf dem Rechner, auf dem Sie den Assistenten gestartet haben, die neueste Version.
- Wenn Sie LicenseServer noch nicht installiert haben und Altova LicenseServer auf einem anderen Rechner installieren möchten und diesen verwenden möchten, deaktivieren Sie das Kontrollkästchen *Altova LicenseServer auf diesem Rechner installieren* und wählen Sie **Später registrieren**. In diesem Fall müssen Sie LicenseServer separat auf dem anderen Rechner installieren und MapForce Server danach auf dem LicenseServer auf diesem Rechner registrieren.
- Wenn LicenseServer auf Ihrem Rechner bereits installiert wurde, aber eine niedrigere Versionsnummer als die vom Installationsassistenten angegebene hat, belassen Sie die Standardeinstellungen (zur Installation eines Upgrade auf die neuere Version) unverändert. In diesem Fall aktualisiert der Installationsassistent Ihre LicenseServer Version automatisch. Die vorhandenen Registrierungs- und Lizenzierungsdaten werden auf die neue Version von LicenseServer übertragen.
- Wenn LicenseServer bereits auf Ihrem Rechner oder in Ihrem Netzwerk installiert wurde und dieselbe Versionsnummer wie diejenige im Assistenten hat, gehen Sie folgendermaßen vor:
 - o Deaktivieren Sie das Kontrollkästchen *LicenseServer auf diesem Rechner installieren*.
 - o Wählen Sie unter *Dieses Produkt registrieren auf* den LicenseServer, auf dem Sie MapForce Server registrieren möchten. Wählen Sie alternativ dazu **Später registrieren**. Beachten Sie, dass Sie immer die Möglichkeit haben, **Später registrieren** auszuwählen, wenn Sie die LicenseServer-Verknüpfungen ignorieren möchten und mit der Installation von MapForce Server fortfahren möchten.

Lesen Sie nach, wie Sie MapForce Server auf dem [Altova LicenseServer registrieren](#)¹⁴ und [lizenzieren](#)¹⁵. Nähere Informationen dazu finden Sie außerdem in der [Dokumentation zu LicenseServer](#).

LicenseServer-Versionen

- Altova-Produkte müssen entweder (i) mit einer Version von LicenseServer, die der installierten Version von MapForce Server entspricht oder (ii) mit einer höheren Version von LicenseServer lizenziert werden.
- Die LicenseServer-Version, die der aktuellen Version von MapForce Server entspricht, ist **3.17**.
- Unter Windows Sie können die dazugehörige Version von LicenseServer zusammen mit MapForce Server installieren oder Sie können LicenseServer separat installieren. Auf Linux- und macOS-Systemen muss LicenseServer separat installiert werden.
- Bevor Sie eine neuere Version von LicenseServer installieren, muss eine eventuell vorhandene ältere Version deinstalliert werden.
- Bei der Deinstallation werden alle Registrierungs- und Lizenzierungsinformationen aus der älteren LicenseServer-Version in einer Datenbank auf Ihrem Server gespeichert. Diese Daten werden bei der Installation der neueren Version automatisch in die neuere Version importiert.
- LicenseServer-Versionen sind rückwärts kompatibel. Sie funktionieren auch mit älteren Versionen von MapForce Server.
- Die neueste Version von LicenseServer steht auf der Altova Website zur Verfügung. Diese Version funktioniert mit allen aktuellen oder älteren Versionen von MapForce Server.
- Sie finden die Versionsnummer des aktuell installierten LicenseServer am unteren Rand der [LicenseServer Konfigurationsseite](#) (alle Register).

2.1.4 Starten von LicenseServer, MapForce Server (Windows)

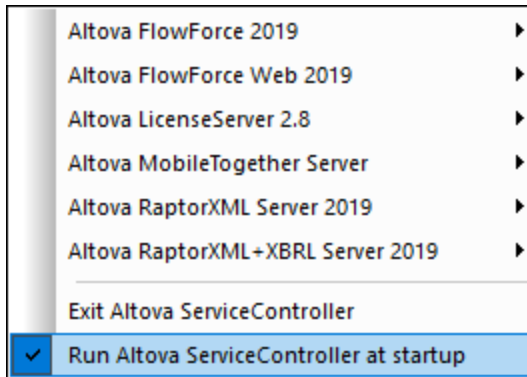
Altova LicenseServer (kurz LicenseServer) und MapForce Server werden beide über Altova ServiceController gestartet.

Altova ServiceController

Der Altova ServiceController (in der Folge ServiceController genannt) ist eine Applikation, mit der Sie Altova-Dienste **auf Windows-Systemen** starten, beenden und konfigurieren können. ServiceController wird mit Altova LicenseServer und als Dienst installierten Altova Server-Produkten installiert (DiffDog Server, FlowForce Server, Mobile Together Server, and RaptorXML(+XBRL) Server) und kann über die Task-Leiste (siehe Abbildung unten) aufgerufen werden.

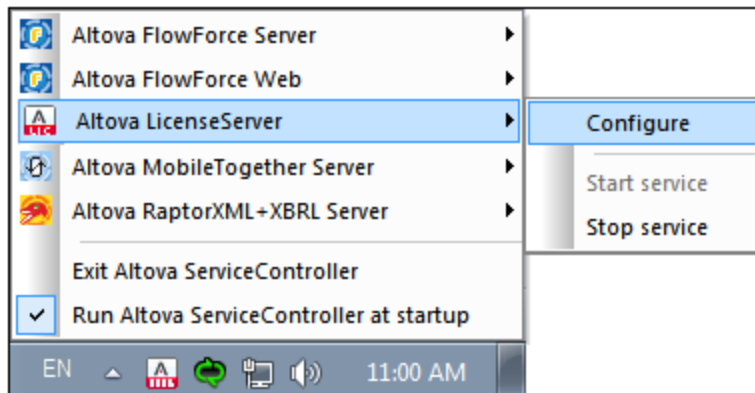


Damit der ServiceController automatisch ausgeführt wird, nachdem sich der Benutzer im System angemeldet hat, klicken Sie in der Task-Leiste auf das **ServiceController**-Symbol, um das **ServiceController**-Menü (Abbildung unten) aufzurufen und aktivieren Sie anschließend den Befehl **Run Altova ServiceController at Startup**. (Dieser Befehl ist standardmäßig aktiv). Um den ServiceController zu beenden, klicken Sie in der Task-Leiste auf das **ServiceController**-Symbol und klicken Sie im Menü, das daraufhin angezeigt wird (Abbildung unten) auf **Exit Altova ServiceController**.



Starten von LicenseServer

Um LicenseServer zu starten, klicken Sie in der Task-Leiste auf das **ServiceController**-Symbol, positionieren Sie den Mauszeiger im angezeigten Menü über **Altova LicenseServer** (siehe Abbildung unten) und wählen Sie anschließend im LicenseServer Untermenü den Befehl **Start Service**. Wenn LicenseServer bereits ausgeführt wird, ist die Option *Start Service* deaktiviert. Sie können den Dienst über Service Controller auch beenden.



2.1.5 Registrieren von MapForce Server (Windows)

Damit MapForce Server über Altova LicenseServer lizenziert werden kann, muss MapForce Server über einen LicenseServer in Ihrem Netzwerk lizenziert sein. Um MapForce Server über die Befehlszeilenschnittstelle zu registrieren, verwenden Sie den Befehl `licenseserver` und geben Sie die Adresse des LicenseServer-Rechners an (siehe unten).

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

Wenn z.B. `localhost` der Name des Servers ist, auf dem LicenseServer installiert ist, verwenden Sie den folgenden Befehl:

MapForceServer licenseserver localhost

Wenn MapForce Server im Rahmen von [FlowForce Server](#) installiert wurde, wird auch MapForce Server bei der Registrierung von FlowForce Server auf LicenseServer registriert. (i) Starten Sie Altova FlowForce Web als Dienst über den ServiceController (*siehe vorhergehender Punkt*); (ii) Geben Sie Ihr Passwort ein, um die Setup-Seite aufzurufen; (iii) Wählen Sie den Namen oder die Adresse des LicenseServers aus und klicken Sie auf **Auf LicenseServer registrieren**. Nähere Informationen dazu finden Sie unter [Registrieren von FlowForce Server](#).

Nachdem Sie MapForce Server erfolgreich registriert haben, gehen Sie zum [Register "Client Management" der LicenseServer-Konfigurationsseite](#) und weisen Sie dem Produkt eine Lizenz zu.

Nähere Informationen zum Registrieren von Altova-Produkten auf LicenseServer finden Sie im [LicenseServer-Benutzerhandbuch](#).

2.1.6 Zuweisen einer Lizenz (Windows)

Nachdem Sie MapForce Server erfolgreich registriert haben, wird das Produkt auf dem Register "Client Management" der LicenseServer-Konfigurationsseite aufgelistet. Gehen Sie zu diesem Register und weisen Sie MapForce Server [eine Lizenz zu](#).

Die Lizenzierung von Altova Server-Produkten basiert auf der Anzahl der auf dem Produktrechner verfügbaren Prozessorkerne. So hat z.B. ein Dual-Core-Prozessor zwei Prozessorkerne, ein Quad-Core-Prozessor vier Kerne, ein Hexa-Core-Prozessor sechs Kerne, usw. Die Anzahl der für ein Produkt lizenzierten Kerne muss größer oder gleich der Anzahl der auf diesem Serverrechner verfügbaren Kerne sein, unabhängig davon, ob es sich um einen physischen Rechner oder eine Virtual Machine handelt. Wenn ein Server z.B. acht Kerne hat (ein Octa-Core-Prozessor), müssen Sie mindestens eine Lizenz für acht Kerne erwerben. Sie können Lizenzen auch kombinieren, um das Produkt für die entsprechende Anzahl von Kernen zu lizenzieren. So können z.B. anstelle einer Lizenz für 8 Kerne auch zwei Lizenzen für 4 Kerne für einen 8-Kern-Prozessor verwendet werden.

Wenn Sie einen Server-Rechner mit einer großen Anzahl von CPU-Kernen verwenden, aber nur geringe Datenmengen verarbeiten müssen, können Sie auch eine Virtual Machine erstellen, der eine geringere Anzahl an Kernen zugewiesen ist und eine Lizenz für diese Anzahl an Kernen erwerben. In diesem Fall ist die Verarbeitungsgeschwindigkeit natürlich geringer als bei Verwendung aller Kerne des Rechners.

Anmerkung: Jede Altova Server-Produktlizenz kann immer nur für einen Client-Rechner gleichzeitig verwendet werden, selbst wenn die Lizenzkapazität dieser Lizenz noch nicht ausgeschöpft ist. (Ein Client-Rechner ist der Rechner, auf dem das Altova Server-Produkt installiert ist.) Wenn z.B. eine 10-Kern-Lizenz für einen Client-Rechner mit 6 CPU-Kernen verwendet wird, so können die verbleibenden Lizenzen für die restlichen 4 Kerne nicht gleichzeitig für einen anderen Client-Rechner verwendet werden.

Lizenzierung von FlowForce Server und MapForce Server

Die FlowForce Server Advanced Edition und die MapForce Server Advanced Edition laufen nur auf Rechnern mit zwei oder mehr Prozessorkernen.

Um zu eruieren, wie viele Prozessorkerne lizenziert werden sollten, sollten die zu verarbeitenden Datenvolumina und die gewünschte maximale Verarbeitungszeit berücksichtigt werden. In den meisten Szenarien bedeuten mehr Prozessorkerne auch die Verarbeitung größerer Datenvolumina in kürzerer Zeit. Im Folgenden finden Sie einige Tipps zu den Applikationen:

- FlowForce Server wird als Multi-Thread-Applikation ausgeführt. Bei einer großen Anzahl von

gleichzeitig an den Server gesendeten Requests kommt es bei zu wenig verfügbaren Prozessorkernen zu Latenzzeiten (Wartezeiten). Wenn Sie z.B. Aufträge als Webservices bereitstellen, können Hunderte von Requests von Clients gleichzeitig einlangen. In diesem Fall profitiert FlowForce Server deutlich, wenn eine größere Anzahl von Prozessorkernen verfügbar ist.

- MapForce Server wird als Single-Thread-Applikation ausgeführt und verwendet immer nur einen einzigen Prozessorkern pro Mapping. Wenn Sie daher mehrere Mappings gleichzeitig ausführen müssen, ist es empfehlenswert, mehr Prozessorkerne zu verwenden. Wenn MapForce Server z.B. unter Verwaltung von FlowForce Server ausgeführt wird, können sich, je nach Konfiguration, mehrere Mapping-Aufträge überlappen und gleichzeitig ausgeführt werden. Beachten Sie jedoch, dass es bei extrem großen Datenvolumina dennoch zu Latenzzeiten kommen kann.

Beachten Sie zusätzlich zu den oben genannten Punkten außerdem, dass das Verarbeitungsvolumen und die Verarbeitungszeiten des Servers normalerweise auch von verschiedenen externen Faktoren beeinflusst wird (z.B. Hardware, CPU-Auslastung, Arbeitsspeicher für andere auf dem Server laufende Applikationen). Um die Leistung möglichst genau messen zu können, empfiehlt es sich, die Tools zuerst in Ihrer Umgebung auszuführen und diese dann im Rahmen Ihrer jeweiligen Geschäftsbedingungen zu testen.

Single-Thread-Ausführung

Wenn bei einem Altova-Server-Produkt eine Single-Thread-Ausführung möglich ist, so steht eine Option für die *Single-Thread-Ausführung* zur Verfügung. Wenn in solchen Fällen im Lizenzpool eine Altova Serverproduktlizenz für nur einen Prozessorkern verfügbar ist, können Sie einem Rechner mit mehreren Kernen diese Lizenz für einen Kern zuweisen. In diesem Fall führt der Rechner das Produkt an einem einzigen Kern aus. Dadurch verlangsamt sich die Verarbeitungsgeschwindigkeit, da kein Multi-Threading (welches bei mehreren Prozessorkernen möglich wäre) zur Verfügung steht. Das Produkt wird auf diesem Rechner im Single Thread-Modus ausgeführt.

Um einem Mehrkernrechner eine Lizenz für nur einen Kern zuzuweisen, aktivieren Sie in LicenseServer für das entsprechende Produkt das Kontrollkästchen *Limit to single thread execution*.

Schätzung der benötigten Prozessorkerne

Es gibt eine Reihe von externen Faktoren, die das Verarbeitungsvolumen und die Verarbeitungszeiten Ihres Servers beeinflussen (z.B. Hardware, CPU-Auslastung, Arbeitsspeicher für andere auf dem Server laufende Applikationen). Um die Leistung möglichst genau messen zu können, empfiehlt es sich, die Applikationen in Ihrer Umgebung mit möglichst realistischen Datenvolumina und unter möglichst realistischen Bedingungen zu testen.

2.2 Einrichten unter Linux

In diesem Abschnitt werden die [Installation](#)¹⁷ und Lizenzierung von MapForce Server auf Linux-Systemen (Debian, Ubuntu, CentOS, RedHat) beschrieben. Bei der Einrichtung müssen die folgenden Schritte durchgeführt werden:

1. [Installation von MapForce Server](#)¹⁷
2. [Installation von LicenseServer](#)¹⁹
3. [Starten von LicenseServer](#)²⁰
4. [Registrieren von MapForce Server auf LicenseServer](#)²⁰
5. [Zuweisen einer Lizenz zu MapForce Server](#)²¹

Die einzelnen oben beschriebenen Schritte müssen nicht in genau derselben Reihenfolge wie aufgelistet erfolgen. Bevor Sie beginnen, muss jedoch zunächst die Installation durchgeführt werden. Außerdem muss MapForce Server auf LicenseServer registriert werden, bevor Sie MapForce Server über LicenseServer eine Lizenz zuweisen können.

Systemvoraussetzungen (Linux)

- Red Hat Enterprise Linux 7 oder neuer
- CentOS 7, CentOS Stream 8
- Debian 10 oder neuer
- Ubuntu 20.04, 22.04, 24.04
- AlmaLinux 9.0
- Rocky Linux 9.0

Voraussetzungen

- Führen Sie die Installation entweder als **root**-Benutzer durch oder als Benutzer mit **sudo**-Rechten.
- Die vorherige Version von MapForce Server muss deinstalliert werden, bevor Sie eine neue Version installieren.
- Um die Applikation installieren und ausführen zu können, werden die folgenden Bibliotheken benötigt. Falls die unten angeführten Pakete auf Ihrem Linux-Rechner noch nicht zur Verfügung stehen, führen Sie die Befehl yum (oder ggf. apt-get) aus, um sie zu installieren.

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

2.2.1 Installation unter Linux

MapForce Server steht für die Installation auf Linux-Systemen zur Verfügung. Führen Sie die Installation entweder als root-Benutzer durch oder als Benutzer mit sudo-Rechten.

Integration von FlowForce Server mit anderen Altova Server-Produkten

Wenn Sie MapForce Server zusammen mit FlowForce Server installieren, sollten Sie zuerst FlowForce Server installieren. Wenn Sie MapForce Server vor FlowForce Server installiert haben, führen Sie im Anschluss an die Installation von MapForce Server und FlowForce Server den folgenden Befehl aus:

```
cp /opt/Altova/MapForceServer2025/etc/*.tool /opt/Altova/FlowForceServer2025/tools
```

Dieser Befehl kopiert die **.tool**-Datei aus dem Verzeichnis **/etc** von MapForce Server in das FlowForce Server **/tools**-Verzeichnis. Die Datei **.tool** wird von FlowForce Server benötigt. Sie enthält den Pfad zur ausführbaren MapForce Server-Datei. Sie müssen diesen Befehl nicht ausführen, wenn Sie FlowForce Server vor MapForce Server installieren.

Deinstallieren von MapForce Server

Bevor Sie MapForce Server installieren, sollten Sie ältere Versionen deinstallieren.

So überprüfen Sie, welche Altova Server-Produkte aktuell installiert sind:

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

So deinstallieren Sie eine alte Version von MapForce Server:

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

Auf Debian- und Ubuntu-Systemen kann es vorkommen, dass MapForce Server auch nach seiner Deinstallation noch in der Liste der installierten Produkte angezeigt wird. Führen Sie in diesem Fall den **purge**-Befehl aus, um MapForce Server aus der Liste zu entfernen. Anstelle des oben aufgelisteten Befehls **remove** können Sie auch den **purge**-Befehl verwenden.

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

Herunterladen des MapForce Server Linux-Pakets

Auf der [Altova Website](#) stehen MapForce Server-Installationspakete für die folgenden Linux-Systeme zur Verfügung.

Distribution	Paketerweiterung
Debian	.deb
Ubuntu	.deb
CentOS	.rpm
RedHat	.rpm

Nachdem Sie das Linux-Paket heruntergeladen haben, kopieren Sie das Paket in ein beliebiges Verzeichnis auf dem Linux-System. Da MapForce Server auf einen [Altova LicenseServer](#) lizenziert werden muss, benötigen, sollten Sie gleichzeitig mit MapForce Server auch LicenseServer von der [Altova Website](#) herunterladen.

Installieren von MapForce Server

Wechseln Sie in einem Terminal-Fenster zu dem Verzeichnis, in das Sie das Linux-Paket kopiert haben. Wenn Sie es z.B. in ein Benutzerverzeichnis namens MyAltova (z.B. im Verzeichnis /home/User) kopiert haben, dann wechseln Sie folgendermaßen zu diesem Verzeichnis:

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

Installieren Sie MapForce Server mit dem entsprechenden Befehl:

```
[Debian]:  sudo dpkg --install mapforceserveradv-2025-debian.deb
[Ubuntu]:  sudo dpkg --install mapforceserveradv-2025-ubuntu.deb
[CentOS]:  sudo rpm -ivh mapforceserveradv-2025-1.x86_64.rpm
[RedHat]:  sudo rpm -ivh mapforceserveradv-2025-1.x86_64.rpm
```

Sie müssen den Namen des obigen Pakets eventuell anpassen, damit er der aktuellen Release- oder Service Pack-Version entspricht.

Das MapForce Server-Paket wird im folgenden Ordner installiert:

```
/opt/Altova/MapForceServer2025
```

2.2.2 Installation von LicenseServer (Linux)

Damit MapForce Server ausgeführt werden kann, muss das Produkt über einen [Altova LicenseServer](#) in Ihrem Netzwerk lizenziert sein. Laden Sie Altova LicenseServer von der [Altova Website](#) herunter und kopieren Sie das Paket in ein beliebiges Verzeichnis. Installieren Sie es genau wie MapForce Server (siehe [vorheriges Kapitel](#)¹⁷).

```
[Debian]:  sudo dpkg --install licenseserver-3.17-debian.deb
[Ubuntu]:  sudo dpkg --install licenseserver-3.17-ubuntu.deb
[CentOS]:  sudo rpm -ivh licenseserver-3.17-1.x86_64.rpm
[RedHat]:  sudo rpm -ivh licenseserver-3.17-1.x86_64.rpm
```

Das LicenseServer-Paket wird im folgenden Ordner installiert:

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

Lesen Sie nach, wie Sie MapForce Server auf dem [Altova LicenseServer registrieren](#)²⁰ und [lizenzieren](#)²¹. Nähere Informationen dazu finden Sie außerdem in der [Dokumentation zu LicenseServer](#).

LicenseServer-Versionen

- Altova-Produkte müssen entweder (i) mit einer Version von LicenseServer, die der installierten Version von MapForce Server entspricht oder (ii) mit einer höheren Version von LicenseServer lizenziert werden.
- Die LicenseServer-Version, die der aktuellen Version von MapForce Server entspricht, ist **3.17**.
- Unter Windows Sie können die dazugehörige Version von LicenseServer zusammen mit MapForce Server installieren oder Sie können LicenseServer separat installieren. Auf Linux- und macOS-Systemen muss LicenseServer separat installiert werden.

- Bevor Sie eine neuere Version von LicenseServer installieren, muss eine eventuell vorhandene ältere Version deinstalliert werden.
- Bei der Deinstallation werden alle Registrierungs- und Lizenzierungsinformationen aus der älteren LicenseServer-Version in einer Datenbank auf Ihrem Server gespeichert. Diese Daten werden bei der Installation der neueren Version automatisch in die neuere Version importiert.
- LicenseServer-Versionen sind rückwärts kompatibel. Sie funktionieren auch mit älteren Versionen von MapForce Server.
- Die neueste Version von LicenseServer steht auf der Altova Website zur Verfügung. Diese Version funktioniert mit allen aktuellen oder älteren Versionen von MapForce Server.
- Sie finden die Versionsnummer des aktuell installierten LicenseServer am unteren Rand der [LicenseServer Konfigurationsseite](#) (alle Register).

2.2.3 Starten von LicenseServer, MapForce Server (Linux)

Starten Sie LicenseServer und MapForce Server entweder als root-Benutzer oder als Benutzer mit sudo-Rechten.

Starten von LicenseServer

Um MapForce Server korrekt auf LicenseServer zu registrieren und zu lizenzieren, muss LicenseServer als Daemon im Netzwerk ausgeführt werden. Starten Sie LicenseServer mit dem folgenden Befehl als Daemon:

```
sudo systemctl start licenseserver
```

(Um LicenseServer zu beenden, ersetzen Sie im obigen Befehl **start** durch **stop**.) Beispiel:

```
sudo systemctl stop licenseserver
```

2.2.4 Registrieren von MapForce Server (Linux)

Damit MapForce Server über Altova LicenseServer lizenziert werden kann, muss MapForce Server über einen LicenseServer in Ihrem Netzwerk lizenziert sein.

Um MapForce Server zu registrieren, rufen Sie seine CLI auf und verwenden Sie den Befehl licenseserver:

```
sudo /opt/Altova/MapForceServer2025/bin/mapforceserver licenseserver [options]  
ServerName-Or-IP-Address
```

Wenn z.B. localhost der Name des Servers ist, auf dem LicenseServer installiert ist:

```
sudo /opt/Altova/MapForceServer2025/bin/mapforceserver licenseserver localhost
```

Im obigen Befehl ist localhost der Name des Servers, auf dem LicenseServer installiert ist. Beachten Sie, dass der Pfad der ausführbaren MapForce Server-Datei der folgende ist:

```
/opt/Altova/MapForceServer2025/bin/
```

Nachdem Sie MapForce Server erfolgreich registriert haben, gehen Sie zum [Register "Client Management" der LicenseServer-Konfigurationsseite](#) und weisen Sie dem Produkt eine Lizenz zu.

Nähere Informationen zum Registrieren von Altova-Produkten auf LicenseServer finden Sie im [LicenseServer-Benutzerhandbuch](#).

2.2.5 Zuweisen einer Lizenz (Linux)

Nachdem Sie MapForce Server erfolgreich registriert haben, wird das Produkt auf dem Register "Client Management" der LicenseServer-Konfigurationsseite aufgelistet. Gehen Sie zu diesem Register und weisen Sie MapForce Server [eine Lizenz zu](#).

Die Lizenzierung von Altova Server-Produkten basiert auf der Anzahl der auf dem Produktrechner verfügbaren Prozessorkerne. So hat z.B. ein Dual-Core-Prozessor zwei Prozessorkerne, ein Quad-Core-Prozessor vier Kerne, ein Hexa-Core-Prozessor sechs Kerne, usw. Die Anzahl der für ein Produkt lizenzierten Kerne muss größer oder gleich der Anzahl der auf diesem Serverrechner verfügbaren Kerne sein, unabhängig davon, ob es sich um einen physischen Rechner oder eine Virtual Machine handelt. Wenn ein Server z.B. acht Kerne hat (ein Octa-Core-Prozessor), müssen Sie mindestens eine Lizenz für acht Kerne erwerben. Sie können Lizenzen auch kombinieren, um das Produkt für die entsprechende Anzahl von Kernen zu lizenzieren. So können z.B. anstelle einer Lizenz für 8 Kerne auch zwei Lizenzen für 4 Kerne für einen 8-Kern-Prozessor verwendet werden.

Wenn Sie einen Server-Rechner mit einer großen Anzahl von CPU-Kernen verwenden, aber nur geringe Datenmengen verarbeiten müssen, können Sie auch eine Virtual Machine erstellen, der eine geringere Anzahl an Kernen zugewiesen ist und eine Lizenz für diese Anzahl an Kernen erwerben. In diesem Fall ist die Verarbeitungsgeschwindigkeit natürlich geringer als bei Verwendung aller Kerne des Rechners.

Anmerkung: Jede Altova Server-Produktlizenz kann immer nur für einen Client-Rechner gleichzeitig verwendet werden, selbst wenn die Lizenzkapazität dieser Lizenz noch nicht ausgeschöpft ist. (Ein Client-Rechner ist der Rechner, auf dem das Altova Server-Produkt installiert ist.) Wenn z.B. eine 10-Kern-Lizenz für einen Client-Rechner mit 6 CPU-Kernen verwendet wird, so können die verbleibenden Lizenzen für die restlichen 4 Kerne nicht gleichzeitig für einen anderen Client-Rechner verwendet werden.

Lizenzierung von FlowForce Server und MapForce Server

Die FlowForce Server Advanced Edition und die MapForce Server Advanced Edition laufen nur auf Rechnern mit zwei oder mehr Prozessorkernen.

Um zu eruieren, wie viele Prozessorkerne lizenziert werden sollten, sollten die zu verarbeitenden Datenvolumina und die gewünschte maximale Verarbeitungszeit berücksichtigt werden. In den meisten Szenarien bedeuten mehr Prozessorkerne auch die Verarbeitung größerer Datenvolumina in kürzerer Zeit. Im Folgenden finden Sie einige Tipps zu den Applikationen:

- FlowForce Server wird als Multi-Thread-Applikation ausgeführt. Bei einer großen Anzahl von gleichzeitig an den Server gesendeten Requests kommt es bei zu wenig verfügbaren Prozessorkernen zu Latenzzeiten (Wartezeiten). Wenn Sie z.B. Aufträge als Webservices bereitstellen, können Hunderte von Requests von Clients gleichzeitig einlangen. In diesem Fall profitiert FlowForce Server deutlich, wenn eine größere Anzahl von Prozessorkernen verfügbar ist.
- MapForce Server wird als Single-Thread-Applikation ausgeführt und verwendet immer nur einen einzigen Prozessorkern pro Mapping. Wenn Sie daher mehrere Mappings gleichzeitig ausführen müssen, ist es empfehlenswert, mehr Prozessorkerne zu verwenden. Wenn MapForce Server z.B. unter Verwaltung von FlowForce Server ausgeführt wird, können sich, je nach Konfiguration, mehrere

Mapping-Aufträge überlappen und gleichzeitig ausgeführt werden. Beachten Sie jedoch, dass es bei extrem großen Datenvolumina dennoch zu Latenzzeiten kommen kann.

Beachten Sie zusätzlich zu den oben genannten Punkten außerdem, dass das Verarbeitungsvolumen und die Verarbeitungszeiten des Servers normalerweise auch von verschiedenen externen Faktoren beeinflusst wird (z.B. Hardware, CPU-Auslastung, Arbeitsspeicher für andere auf dem Server laufende Applikationen). Um die Leistung möglichst genau messen zu können, empfiehlt es sich, die Tools zuerst in Ihrer Umgebung auszuführen und diese dann im Rahmen Ihrer jeweiligen Geschäftsbedingungen zu testen.

Single-Thread-Ausführung

Wenn bei einem Altova-Server-Produkt eine Single-Thread-Ausführung möglich ist, so steht eine Option für die *Single-Thread-Ausführung* zur Verfügung. Wenn in solchen Fällen im Lizenzpool eine Altova Serverproduktlizenz für nur einen Prozessorkern verfügbar ist, können Sie einem Rechner mit mehreren Kernen diese Lizenz für einen Kern zuweisen. In diesem Fall führt der Rechner das Produkt an einem einzigen Kern aus. Dadurch verlangsamt sich die Verarbeitungsgeschwindigkeit, da kein Multi-Threading (welches bei mehreren Prozessorkernen möglich wäre) zur Verfügung steht. Das Produkt wird auf diesem Rechner im Single Thread-Modus ausgeführt.

Um einem Mehrkernrechner eine Lizenz für nur einen Kern zuzuweisen, aktivieren Sie in LicenseServer für das entsprechende Produkt das Kontrollkästchen *Limit to single thread execution*.

Schätzung der benötigten Prozessorkerne

Es gibt eine Reihe von externen Faktoren, die das Verarbeitungsvolumen und die Verarbeitungszeiten Ihres Servers beeinflussen (z.B. Hardware, CPU-Auslastung, Arbeitsspeicher für andere auf dem Server laufende Applikationen). Um die Leistung möglichst genau messen zu können, empfiehlt es sich, die Applikationen in Ihrer Umgebung mit möglichst realistischen Datenvolumina und unter möglichst realistischen Bedingungen zu testen.

2.3 Einrichten auf macOS

In diesem Abschnitt werden die [Installation](#)²³ und Lizenzierung von MapForce Server auf macOS-Systemen beschrieben. Bei der Einrichtung müssen die folgenden Schritte durchgeführt werden:

1. [Installation von MapForce Server](#)²³
2. [Installation von LicenseServer](#)²⁴
3. [Starten von LicenseServer](#)²⁵
4. [Registrieren von MapForce Server auf LicenseServer](#)²⁵
5. [Zuweisen einer Lizenz zu MapForce Server](#)²⁶

Die einzelnen oben beschriebenen Schritte müssen nicht in genau derselben Reihenfolge wie aufgelistet erfolgen. Bevor Sie beginnen, muss jedoch zunächst die Installation durchgeführt werden. Außerdem muss MapForce Server auf LicenseServer registriert werden, bevor Sie MapForce Server über LicenseServer eine Lizenz zuweisen können.

Systemvoraussetzungen (macOS)

Beachten Sie die folgende Systemvoraussetzung:

- macOS 12 oder neuer

Voraussetzungen

Beachten Sie dabei die folgenden Voraussetzungen:

- Stellen Sie sicher, dass Altova LicenseServer installiert wurde und ausgeführt wird.
- Führen Sie die Installation entweder als root-Benutzer durch oder als Benutzer mit sudo-Rechten.
- Die vorherige Version von MapForce Server muss deinstalliert werden, bevor Sie eine neue Version installieren.
- Der macOS-Rechner muss so konfiguriert sein, dass sein Name zu einer IP-Adresse aufgelöst wird (d.h. der Host-Name muss vom Terminal aus mit dem Befehl **ping <hostname>** erfolgreich angepingt werden können).

2.3.1 Installation auf macOS

In diesem Kapitel werden die Installation und Konfiguration von MapForce Server auf macOS-Systemen beschrieben.

Integration mit FlowForce

Wenn Sie MapForce Server zusammen mit FlowForce Server installieren, sollten Sie zuerst FlowForce Server installieren. Wenn Sie MapForce Server vor FlowForce Server installiert haben, führen Sie im Anschluss an die Installation der beiden Produkte den folgenden Befehl aus:

```
cp /usr/local/Altova/MapForceServer2025/etc/*.tool /usr/local/Altova/FlowForceServer2025/tools
```

Dieser Befehl kopiert die `.tool`-Datei aus dem Verzeichnis `/etc` von MapForce Server in das FlowForce Server `/tools`-Verzeichnis. Die Datei `.tool` wird von FlowForce Server benötigt. Sie enthält den Pfad zur ausführbaren MapForce Server-Datei. Sie müssen diesen Befehl nicht ausführen, wenn Sie FlowForce Server vor MapForce Server installieren.

Deinstallieren von MapForce Server

Klicken Sie im Finder unter Applications mit der rechten Maustaste auf das MapForce Server-Symbol und wählen Sie den Befehl **Move to Trash**. Die Applikation wird daraufhin in den Papierkorb verschoben. Sie müssen die Applikation jetzt noch aus dem Ordner `usr` entfernen. Führen Sie dazu den folgenden Befehl aus:

```
sudo rm -rf /usr/local/Altova/MapForceServer2025/
```

Wenn Sie eine ältere Version von Altova LicenseServer deinstallieren müssen, beenden Sie den Dienst zuerst mit dem folgenden Befehl:

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

Um zu überprüfen, ob ein Dienst beendet wurde, öffnen Sie den Activity Monitor im Finder und stellen Sie sicher, dass sich LicenseServer nicht in der Liste befindet. Fahren Sie anschließend mit der Deinstallation fort, wie oben für MapForce Server beschrieben.

Installieren von MapForce Server

Um MapForce Server zu installieren, gehen Sie folgendermaßen vor:

1. Laden Sie die `.dmg` (Disk Image)-Datei von MapForce Server von der Altova Website (<https://www.altova.com/de/download.html>) in ein lokales Verzeichnis herunter.
2. Klicken Sie auf die heruntergeladene Disk Image-Datei (`.dmg`), um sie zu öffnen. Dadurch wird der MapForce Server Installer als neues virtuelles Laufwerk auf Ihrem Computer angezeigt.
3. Doppelklicken Sie in diesem neuen virtuellen Laufwerk auf das Installer-Paket (`.pkg`).
4. Befolgen Sie die selbsterklärenden Anweisungen des Installationsassistenten, in dem Sie auch die Lizenzvereinbarung akzeptieren müssen, bevor Sie mit der Installation fortfahren können.
5. Um das Laufwerk nach der Installation auszuwerfen, klicken Sie mit der rechten Maustaste darauf und wählen Sie **Eject**.

Das MapForce Server-Paket wird im folgenden Ordner installiert:

```
/usr/local/Altova/MapForceServer2025 (Applikationsbinärdateien)  
/var/Altova/MapForceServer (Datendateien: Datenbank und Logs)
```

Der MapForce Server Server-Daemon wird nach einer Installation und einem Neustart des Rechners automatisch gestartet. Sie können MapForce Server jederzeit mit dem folgenden Befehl als Daemon starten:

```
sudo launchctl load /Library/LaunchDaemons/com.altova.MapForceServer2025.plist
```

2.3.2 Installation von LicenseServer (macOS)

Altova LicenseServer kann von der Altova Website (<https://www.altova.com/de/download.html>) heruntergeladen werden. Führen Sie die Installation durch, wie [hier](#)²³ beschrieben.

Das LicenseServer-Paket wird im folgenden Ordner installiert:

`/usr/local/Altova/LicenseServer`

Lesen Sie nach, wie Sie MapForce Server auf dem [Altova LicenseServer registrieren](#)²⁵ und [lizenzieren](#)²⁶. Nähere Informationen dazu finden Sie außerdem in der [Dokumentation zu LicenseServer](#).

LicenseServer-Versionen

- Altova-Produkte müssen entweder (i) mit einer Version von LicenseServer, die der installierten Version von MapForce Server entspricht oder (ii) mit einer höheren Version von LicenseServer lizenziert werden.
- Die LicenseServer-Version, die der aktuellen Version von MapForce Server entspricht, ist **3.17**.
- Unter Windows Sie können die dazugehörige Version von LicenseServer zusammen mit MapForce Server installieren oder Sie können LicenseServer separat installieren. Auf Linux- und macOS-Systemen muss LicenseServer separat installiert werden.
- Bevor Sie eine neuere Version von LicenseServer installieren, muss eine eventuell vorhandene ältere Version deinstalliert werden.
- Bei der Deinstallation werden alle Registrierungs- und Lizenzierungsinformationen aus der älteren LicenseServer-Version in einer Datenbank auf Ihrem Server gespeichert. Diese Daten werden bei der Installation der neueren Version automatisch in die neuere Version importiert.
- LicenseServer-Versionen sind rückwärts kompatibel. Sie funktionieren auch mit älteren Versionen von MapForce Server.
- Die neueste Version von LicenseServer steht auf der Altova Website zur Verfügung. Diese Version funktioniert mit allen aktuellen oder älteren Versionen von MapForce Server.
- Sie finden die Versionsnummer des aktuell installierten LicenseServer am unteren Rand der [LicenseServer Konfigurationsseite](#) (alle Register).

2.3.3 Starten von LicenseServer, MapForce Server (macOS)

Starten Sie LicenseServer und MapForce Server entweder als root-Benutzer oder als Benutzer mit sudo-Rechten.

Starten von LicenseServer

Um MapForce Server korrekt auf LicenseServer zu registrieren und zu lizenzieren, muss LicenseServer als Daemon ausgeführt werden. Starten Sie LicenseServer mit dem folgenden Befehl als Daemon:

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

Um LicenseServer zu beenden, ersetzen Sie im obigen Befehl **load** durch **unload**.)

2.3.4 Registrieren von MapForce Server (macOS)

Damit MapForce Server über Altova LicenseServer lizenziert werden kann, muss MapForce Server über einen LicenseServer in Ihrem Netzwerk lizenziert sein.

Um MapForce Server über die Befehlszeile zu registrieren, verwenden Sie den Befehl `licenseserver`:

```
sudo /usr/local/Altova/MapForceServer2025/bin/MapForceServer licenseserver [options]  
ServerName-Or-IP-Address
```

Wenn z.B. localhost der Name des Servers ist, auf dem LicenseServer installiert ist:

```
sudo /usr/local/Altova/MapForceServer2025/bin/MapForceServer licenseserver localhost
```

Im obigen Befehl ist localhost der Name des Servers, auf dem LicenseServer installiert ist. Beachten Sie, dass der Pfad der ausführbaren MapForce Server-Datei der folgende ist:

```
/usr/local/Altova/MapForceServer2025/bin/
```

Nachdem Sie MapForce Server erfolgreich registriert haben, gehen Sie zum [Register "Client Management" der LicenseServer-Konfigurationsseite](#) und weisen Sie dem Produkt eine Lizenz zu.

Nähere Informationen zum Registrieren von Altova-Produkten auf LicenseServer finden Sie im [LicenseServer-Benutzerhandbuch](#).

2.3.5 Zuweisen einer Lizenz (macOS)

Nachdem Sie MapForce Server erfolgreich registriert haben, wird das Produkt auf dem Register "Client Management" der LicenseServer-Konfigurationsseite aufgelistet. Gehen Sie zu diesem Register und weisen Sie MapForce Server [eine Lizenz zu](#).

Die Lizenzierung von Altova Server-Produkten basiert auf der Anzahl der auf dem Produktrechner verfügbaren Prozessorkerne. So hat z.B. ein Dual-Core-Prozessor zwei Prozessorkerne, ein Quad-Core-Prozessor vier Kerne, ein Hexa-Core-Prozessor sechs Kerne, usw. Die Anzahl der für ein Produkt lizenzierten Kerne muss größer oder gleich der Anzahl der auf diesem Serverrechner verfügbaren Kerne sein, unabhängig davon, ob es sich um einen physischen Rechner oder eine Virtual Machine handelt. Wenn ein Server z.B. acht Kerne hat (ein Octa-Core-Prozessor), müssen Sie mindestens eine Lizenz für acht Kerne erwerben. Sie können Lizenzen auch kombinieren, um das Produkt für die entsprechende Anzahl von Kernen zu lizenzieren. So können z.B. anstelle einer Lizenz für 8 Kerne auch zwei Lizenzen für 4 Kerne für einen 8-Kern-Prozessor verwendet werden.

Wenn Sie einen Server-Rechner mit einer großen Anzahl von CPU-Kernen verwenden, aber nur geringe Datenmengen verarbeiten müssen, können Sie auch eine Virtual Machine erstellen, der eine geringere Anzahl an Kernen zugewiesen ist und eine Lizenz für diese Anzahl an Kernen erwerben. In diesem Fall ist die Verarbeitungsgeschwindigkeit natürlich geringer als bei Verwendung aller Kerne des Rechners.

Anmerkung: Jede Altova Server-Produktlizenz kann immer nur für einen Client-Rechner gleichzeitig verwendet werden, selbst wenn die Lizenzkapazität dieser Lizenz noch nicht ausgeschöpft ist. (Ein Client-Rechner ist der Rechner, auf dem das Altova Server-Produkt installiert ist.) Wenn z.B. eine 10-Kern-Lizenz für einen Client-Rechner mit 6 CPU-Kernen verwendet wird, so können die verbleibenden Lizenzen für die restlichen 4 Kerne nicht gleichzeitig für einen anderen Client-Rechner verwendet werden.

Lizenzierung von FlowForce Server und MapForce Server

Die FlowForce Server Advanced Edition und die MapForce Server Advanced Edition laufen nur auf Rechnern mit zwei oder mehr Prozessorkernen.

Um zu eruieren, wie viele Prozessorkerne lizenziert werden sollten, sollten die zu verarbeitenden Datenvolumina

und die gewünschte maximale Verarbeitungszeit berücksichtigt werden. In den meisten Szenarien bedeuten mehr Prozessorkerne auch die Verarbeitung größerer Datenvolumina in kürzerer Zeit. Im Folgenden finden Sie einige Tipps zu den Applikationen:

- FlowForce Server wird als Multi-Thread-Applikation ausgeführt. Bei einer großen Anzahl von gleichzeitig an den Server gesendeten Requests kommt es bei zu wenig verfügbaren Prozessorkernen zu Latenzzeiten (Wartezeiten). Wenn Sie z.B. Aufträge als Webservices bereitstellen, können Hunderte von Requests von Clients gleichzeitig einlangen. In diesem Fall profitiert FlowForce Server deutlich, wenn eine größere Anzahl von Prozessorkernen verfügbar ist.
- MapForce Server wird als Single-Thread-Applikation ausgeführt und verwendet immer nur einen einzigen Prozessorkern pro Mapping. Wenn Sie daher mehrere Mappings gleichzeitig ausführen müssen, ist es empfehlenswert, mehr Prozessorkerne zu verwenden. Wenn MapForce Server z.B. unter Verwaltung von FlowForce Server ausgeführt wird, können sich, je nach Konfiguration, mehrere Mapping-Aufträge überlappen und gleichzeitig ausgeführt werden. Beachten Sie jedoch, dass es bei extrem großen Datenvolumina dennoch zu Latenzzeiten kommen kann.

Beachten Sie zusätzlich zu den oben genannten Punkten außerdem, dass das Verarbeitungsvolumen und die Verarbeitungszeiten des Servers normalerweise auch von verschiedenen externen Faktoren beeinflusst wird (z.B. Hardware, CPU-Auslastung, Arbeitsspeicher für andere auf dem Server laufende Applikationen). Um die Leistung möglichst genau messen zu können, empfiehlt es sich, die Tools zuerst in Ihrer Umgebung auszuführen und diese dann im Rahmen Ihrer jeweiligen Geschäftsbedingungen zu testen.

Single-Thread-Ausführung

Wenn bei einem Altova-Server-Produkt eine Single-Thread-Ausführung möglich ist, so steht eine Option für die *Single-Thread-Ausführung* zur Verfügung. Wenn in solchen Fällen im Lizenzpool eine Altova Serverproduktlizenz für nur einen Prozessorkern verfügbar ist, können Sie einem Rechner mit mehreren Kernen diese Lizenz für einen Kern zuweisen. In diesem Fall führt der Rechner das Produkt an einem einzigen Kern aus. Dadurch verlangsamt sich die Verarbeitungsgeschwindigkeit, da kein Multi-Threading (welches bei mehreren Prozessorkernen möglich wäre) zur Verfügung steht. Das Produkt wird auf diesem Rechner im Single Thread-Modus ausgeführt.

Um einem Mehrkernrechner eine Lizenz für nur einen Kern zuzuweisen, aktivieren Sie in LicenseServer für das entsprechende Produkt das Kontrollkästchen *Limit to single thread execution*.

Schätzung der benötigten Prozessorkerne

Es gibt eine Reihe von externen Faktoren, die das Verarbeitungsvolumen und die Verarbeitungszeiten Ihres Servers beeinflussen (z.B. Hardware, CPU-Auslastung, Arbeitsspeicher für andere auf dem Server laufende Applikationen). Um die Leistung möglichst genau messen zu können, empfiehlt es sich, die Applikationen in Ihrer Umgebung mit möglichst realistischen Datenvolumina und unter möglichst realistischen Bedingungen zu testen.

2.4 Upgraden von MapForce Server

Am einfachsten lässt sich die Lizenz aus der vorherigen Version von MapForce Server bei der Installation auf die neuere Version übertragen: Die wichtigsten Schritte bei der Installation sind die folgenden:

1. Registrieren Sie die neue Version von MapForce Server auf dem LicenseServer, auf dem sich die Lizenz der älteren Version von MapForce Server befindet.
2. Akzeptieren Sie die Lizenzvereinbarung von MapForce Server. (Wenn Sie der Vereinbarung nicht zustimmen, wird die neue Version nicht installiert.)

Anmerkung: Wenn Sie MapForce Server nicht während der Installation auf LicenseServer registrieren, können Sie dies später nachholen und die Lizenzierung erst dann abschließen.

2.5 Migrieren von MapForce Server auf einen neuen Rechner

Wenn Sie MapForce Server von einem Rechner auf einen anderen (eventuell auch auf eine andere Plattform) migrieren möchten, befolgen Sie die Richtlinien weiter unten.

Bei der Migration von MapForce Server auf einen neuen Rechner wird die Lizenz vom alten Rechner einem neuen Rechner zugewiesen. Gehen Sie dazu folgendermaßen vor:

1. Installieren Sie MapForce Server auf dem neuen Rechner. Wenn das Produkt bereits im Rahmen der FlowForce Server-Installation installiert wurde, ignorieren Sie diesen Schritt.
2. Registrieren Sie MapForce Server auf dem neuen Rechner auf Altova LicenseServer.
3. Stellen Sie sicher, dass der Server auf dem alten Rechner von keinen Clients verwendet wird (dass z.B. keine Mappings ausgeführt werden).
4. Öffnen Sie die Altova LicenseServer-Verwaltungsseite. Deaktivieren Sie die Lizenz des alten MapForce Server Rechners und weisen Sie sie dem neuen Rechner zu.

Anmerkung: Wenn Sie auf dem alten Rechner XML-Kataloge verwendet haben, migrieren Sie diese auf den neuen Rechner.

2.6 Sicherheitsaspekte

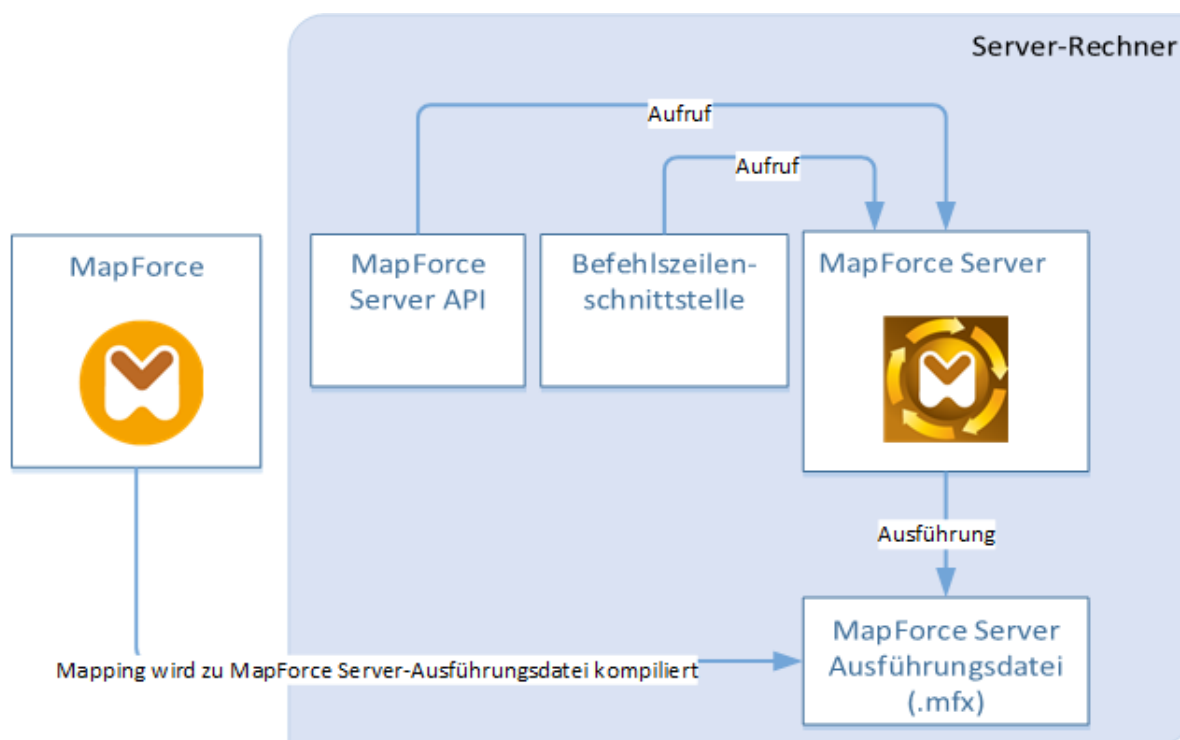
XSLT, XPath und XQuery sind Turing-vollständige funktionale Programmiersprachen mit Zugriff auf lokale und entfernte Dateien und der Möglichkeit zur dynamischen Ausführung. Es wird daher empfohlen, den Zugriff auf diese Dateien für Transformationen und/oder die Dateiverarbeitung nur in einer sicheren und geregelten Umgebung, in der die Kontrolle über die Input-Dateien gewährleistet ist und sichergestellt werden kann, dass nur zuvor überprüfte Scripts ausgeführt werden, zu gestatten. Sollte die Notwendigkeit bestehen, diese Dateien von einem externen/öffentlichen Netzwerk (oder einem nicht sicheren Sub-Netzwerk) aufzurufen, wird empfohlen, den Zugriff mit Hilfe eines Reverse Proxy Servers, der eine Benutzerauthentifizierung und -autorisierung implementiert, einzuschränken. Außerdem wird empfohlen, den Vorgang über ein separates Benutzerkonto auszuführen, bei dem der Zugriff auf Betriebssystemebene konfiguriert ist, um den Zugriff auf die autorisierten Bereiche des Dateisystems einzuschränken.

3 Ausführen von Mappings

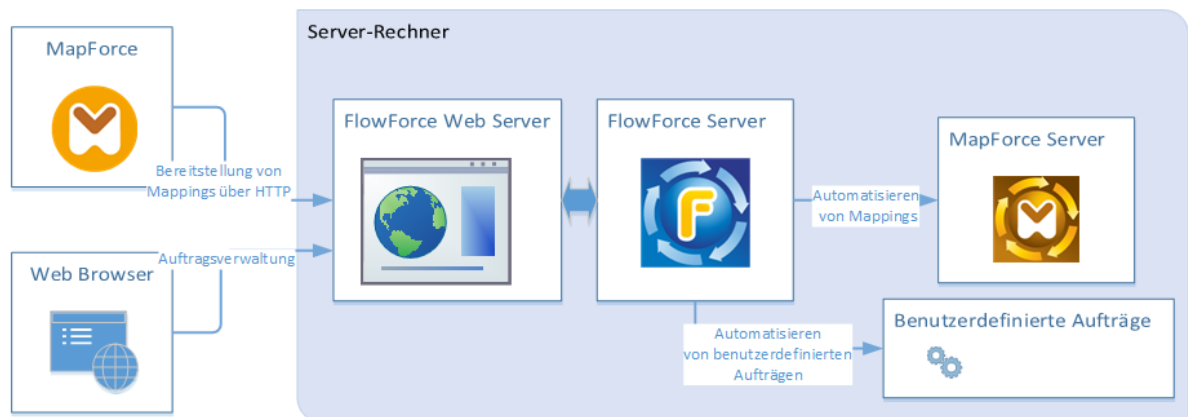
Die Aufgabe von MapForce Server ist es, mit Altova MapForce (<https://www.altova.com/de/mapforce.html>) erstellte Datenmappings auszuführen.

Zuerst müssen die Datenmappings (oder Mapping-Design-Dateien, *.mfd) mit grafischen Mitteln in MapForce erstellt werden. In MapForce werden die Inputs, Outputs und Zwischenverarbeitungsschritte für Ihre Daten (wie Sortieren, Filtern, benutzerdefinierte Funktionen und andere) erstellt. Sobald Ihr Mapping fertig ist, können Sie es auf eine der folgenden Arten mit MapForce Server ausführen:

- Kompilieren Sie das Mapping auf dem Windows-Rechner, auf dem MapForce ausgeführt wird, zu einer MapForce Server-Ausführungsdatei (.mfx). Bei den .mfx-Dateien handelt es sich um Datenmappings, die in ein Paket für die Ausführung in einer Serverumgebung verpackt wurden. Sie können solche Dateien auf jedes der unterstützten Betriebssysteme, auf denen MapForce Server ausgeführt wird (auch auf andere Plattformen, siehe [Systemanforderungen](#)⁸) kopieren. Auf dem Server können Sie die .mfx-Datei über die Befehlszeile von MapForce Server oder über die MapForce Server API ausführen.



- Stellen Sie das Mapping von dem Windows-Rechner, auf dem MapForce ausgeführt wird, auf einem Server-Rechner, auf dem sowohl MapForce Server als auch FlowForce Server installiert sind, bereit. Beim Server-Rechner kann es sich auch um ein anderes Betriebssystem handeln (siehe [Systemanforderungen](#)⁸). Mappings, die auf diese Art bereitgestellt wurden, werden zu FlowForce-Funktionen, für die Sie geplante oder Trigger-basierte Aufträge erstellen können. Wenn Mappings als FlowForce Server-Aufträge ausgeführt werden, können Sie auch als Webservices, als anderen Aufträgen untergeordnete Schritte, als Teile von Workflows, die auch das Senden von E-Mails, das Überprüfen von Exit-Codes, das Ausführen von Shell-Befehlen usw. umfassen, definiert werden.



Nähere Informationen zu diesem Szenario finden Sie in der FlowForce Server-Dokumentation (<https://www.altova.com/de/documentation>).

So führen Sie als MapForce Server-Ausführungsdatei kompilierte Mappings aus

1. Starten Sie die MapForce Enterprise oder Professional Edition.
2. Öffnen Sie das zu kompilierende Mapping.
3. Klicken Sie im Menü **Datei** auf den Befehl **Zu MapForce Server-Ausführungsdatei kompilieren** und wählen Sie ein Zielverzeichnis aus.
4. Kopieren Sie die .mfx-Datei zusammen mit allen Input-Dateien oder Abhängigkeiten in das Zielverzeichnis bzw. auf den gewünschten Server. Nähere Informationen dazu finden Sie unter [Vorbereiten von Mappings für die Server-Ausführung](#)³⁴.
5. Rufen Sie den Befehl "run" der [Befehlszeilenschnittstelle](#)⁶³ oder die entsprechende Methode der MapForce Server API (siehe [MapForce Server API](#)⁷⁹) auf.

So führen Sie auf FlowForce Server bereitgestellte Mappings aus

1. Öffnen Sie das gewünschte Mapping in MapForce Enterprise oder Professional.
2. Stellen Sie sicher, dass als Transformationssprache (Ausführungsprozessor) des Mappings Built-in ausgewählt ist. Um den Ausführungsprozessor in Built-in zu ändern, wählen Sie den Menübefehl **Ausgabe | Built-In Ausführungsprozessor** oder klicken Sie auf die Symbolleiste-Schaltfläche **Built-In Ausführungsprozessor auswählen** ().
3. Klicken Sie im Menü **Datei** auf **Auf FlowForce Server bereitstellen**.
4. Geben Sie die Serververbindungsinformationen (Host, Port), die FlowForce-Anmeldeinformationen und den FlowForce-Ziel-Container ein. Damit der FlowForce-Auftrag sofort im Browser erstellt wird, aktivieren Sie die Option **Web Browser zur Erstellung eines neuen Auftrags öffnen**. Sie können den FlowForce-Auftrag auch zu einem späteren Zeitpunkt erstellen (siehe nächster Schritt).
5. Öffnen Sie einen Browser, melden Sie sich auf der FlowForce Server Web-Verwaltungsschnittstelle an und navigieren Sie zu dem Container, in dem Sie das Mapping bereitgestellt haben (siehe vorheriger Schritt). Dieser Schritt fällt weg, wenn Sie die Option **Web Browser zur Erstellung eines neuen Auftrags öffnen** im vorigen Schritt aktiviert haben.
6. Definieren Sie den FlowForce Server-Auftrag mit seinen Triggern, Parametern oder zusätzlichen Ausführungsschritten (als Beispiel siehe FlowForce Server-Dokumentation <https://www.altova.com/de/documentation>). Wenn der Auftrag laut Konfiguration gestartet wird, wird das zugrunde liegende Mapping ausgeführt und die Mapping-Ausgabe erzeugt.

Anmerkung: Wenn MapForce Server auf einem anderen Rechner als dem, auf dem das Mapping erstellt wurde, ausgeführt wird, müssen Sie die Pfade zu den Input-Dateien oder die

Datenbankverbindungsinformationen so definieren, dass sie in der neuen Ausführungsumgebung funktionieren. Wenn z.B. eine Datenbank in einem Mapping aufgerufen wird und dafür ein Datenbanktreiber benötigt wird, so muss der Treiber auch in der Zielumgebung installiert sein, damit das Mapping erfolgreich ausgeführt werden kann. Klicken Sie in MapForce mit der rechten Maustaste auf die Datenbankkomponente und wählen Sie **Eigenschaften**, um die Datenbankverbindungsinformationen anzupassen. Nachdem Sie in MapForce Änderungen am Mapping-Design vorgenommen haben, muss es erneut zu einer MapForce Server-Ausführungsdatei (.mfx) kompiliert oder, je nach Situation, erneut auf FlowForce Server bereitgestellt werden. Nähere Informationen finden Sie unter [Preparing Mappings for Server Execution](#) ³⁴.

3.1 Vorbereiten von Mappings für die Server-Ausführung

Ein Mit MapForce erstelltes Mapping, das auch in der MapForce-Vorschau angezeigt wird, kann Ressourcen (z.B. Datenbanken) referenzieren, die sich nicht auf dem aktuellen Rechner und Betriebssystem befinden. Außerdem entsprechen in MapForce alle Mapping-Pfade standardmäßig den Windows-Konventionen. Zudem unterstützt der Rechner, auf dem MapForce Server läuft, nicht notwendigerweise dieselben Datenbankverbindungen wie der Rechner, auf dem das Mapping erstellt wurde. Aus diesem Grund, müssen Sie ein Mapping entsprechend vorbereiten, bevor Sie es in einer Server-Umgebung ausführen, vor allem dann, wenn der Zielrechner nicht derselbe wie der Quellrechner ist.

Anmerkung: Mit "Quellrechner" wird der Computer bezeichnet, auf dem MapForce installiert ist und mit "Zielrechner" wird der Computer bezeichnet, auf dem MapForce Server oder FlowForce Server installiert ist. Im einfachsten Szenario handelt es sich hierbei um denselben Computer. In einem komplexeren Szenario läuft MapForce auf einem Windows-Rechner, während MapForce Server auf einem Linux- oder macOS-Rechner ausgeführt wird.

Stellen Sie am besten immer sicher, dass sich das Mapping in MapForce erfolgreich validieren lässt, bevor Sie es auf FlowForce Server bereitstellen oder zu einer MapForce Server-Ausführungsdatei kompilieren.

Wenn MapForce Server alleine (ohne FlowForce Server) ausgeführt wird, werden die folgenden Lizenzen benötigt:

- Auf dem Quellrechner benötigen Sie zur Erstellung des Mappings und zum Kompilieren zu einer Server-Ausführungsdatei (.mfx) die MapForce Enterprise oder Professional Edition.
- Auf dem Zielrechner benötigen Sie zur Ausführung des Mappings MapForce Server oder MapForce Server Advanced Edition.

Wenn MapForce Server unter Verwaltung von FlowForce Server ausgeführt wird, gelten die folgenden Voraussetzungen:

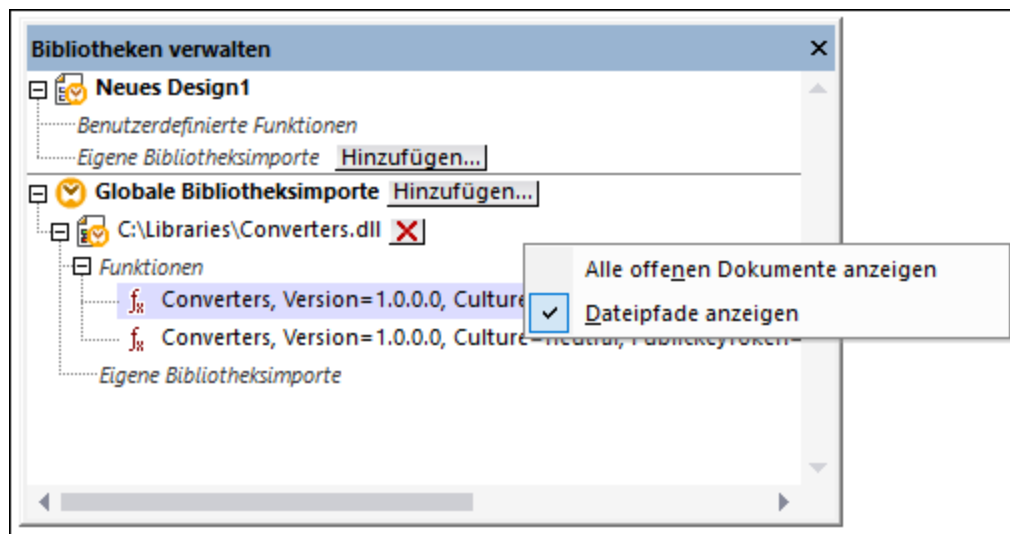
- Auf dem Quellrechner benötigen Sie zur Erstellung des Mappings und für die Bereitstellung des Mappings auf einem Zielrechner die MapForce Enterprise oder Professional Edition.
- Sowohl MapForce Server als auch FlowForce Server müssen auf dem Zielrechner lizenziert sein. Die Aufgabe von MapForce Server ist die Ausführung des Mappings; die Aufgabe von FlowForce ist es, das Mapping in Form eines Auftrags bereitzustellen, wodurch Funktionen wie Ausführung nach einem Zeitplan oder auf Wunsch, die Ausführung als Webservice, Fehlerbehandlung, Ausführung auf Basis von Bedingungen, E-Mail-Benachrichtigungen usw. zur Verfügung stehen.
- FlowForce Server muss an der konfigurierten Netzwerkadresse und am konfigurierten Port gestartet sein. Dabei muss der "FlowForce Web Server"-Dienst gestartet sein und so konfiguriert sein, dass er Verbindungen von HTTP-Clients (oder bei Bedarf von HTTPS-Clients) zulässt und er darf nicht durch die Firewall blockiert werden. Auch der "FlowForce Server"-Dienst muss unter der vorgesehenen Adresse und am angegebenen Port gestartet und verfügbar sein.
- Sie haben ein FlowForce Server-Benutzerkonto und Zugriff auf einen der Container (standardmäßig steht der Container **/public** jedem authentifizierten Benutzer zur Verfügung).

Allgemeine Überlegungen

- Wenn Sie beabsichtigen, das Mapping mit MapForce Server alleine auf einem Zielrechner auszuführen, müssen alle vom Mapping referenzierten Input-Dateien ebenfalls auf den Zielrechner kopiert werden. Wenn MapForce Server unter Verwaltung von FlowForce Server ausgeführt wird, müssen die Dateien

nicht manuell kopiert werden. In diesem Fall sind die Instanz- und Schema-Dateien in dem auf dem Zielrechner bereitgestellten Paket inkludiert.

- Wenn das Mapping Datenbankkomponenten enthält, für die spezielle Datenbanktreiber benötigt werden, müssen diese Treiber auch auf dem Zielrechner installiert werden. Wenn in Ihrem Mapping z.B. Daten aus einer Microsoft Access-Datenbank ausgelesen werden, so muss Microsoft Access oder Microsoft Access Runtime (<https://www.microsoft.com/en-us/download/details.aspx?id=50040>) auch auf dem Zielrechner installiert sein.
- Wenn Sie ein Mapping auf Nicht-Windows-Plattformen bereitstellen, werden ADO-, ADO.NET- und ODBC-Datenbankverbindungen automatisch in JDBC konvertiert. Native SQLite- und PostgreSQL-Verbindungen werden als solche beibehalten und müssen nicht zusätzlich konfiguriert werden. Siehe auch "Datenbankverbindungen" weiter unten.
- Wenn das Mapping benutzerdefinierte Funktionsaufrufe enthält (z.B. Aufrufe von .dll- oder .class-Dateien), werden diese Abhängigkeiten nicht zusammen mit dem Mapping bereitgestellt, da sie vor der Laufzeit nicht bekannt sind. Kopieren Sie diese in diesem Fall manuell auf den Zielrechner. Der Pfad der .dll- oder .class-Datei auf dem Server muss derselbe sein, der in MapForce im Dialogfeld "Bibliotheken verwalten" definiert wurde, z.B.:



- In einigen Mappings werden mehrere Input-Dateien über einen Platzhalter-Pfad ausgelesen. In diesem Fall sind die Namen der Input-Dateien vor der Laufzeit noch nicht bekannt und können daher nicht bereitgestellt werden. Damit das Mapping erfolgreich ausgeführt werden kann, muss die Input-Datei auf dem Zielrechner vorhanden sein.
- Wenn der Mapping-Ausgabepfad Verzeichnisse enthält, müssen diese auf dem Zielrechner vorhanden sein, da sonst bei Ausführung des Mappings ein Fehler generiert wird. Im Gegensatz dazu werden in MapForce nicht vorhandene Verzeichnisse automatisch generiert, wenn die Option Output-Datei als temporäre Datei generieren aktiviert ist.
- Wenn im Mapping ein Webservice aufgerufen wird, für den eine HTTPS-Authentifizierung mit einem Client-Zertifikat erforderlich ist, muss das Zertifikat ebenfalls an den Zielrechner übertragen werden (siehe [Verwaltung digitaler Zertifikate](#)¹⁴²).
- Wenn im Mapping eine Verbindung zu dateibasierten Datenbanken wie Microsoft Access und SQLite hergestellt wird, muss die Datenbank manuell auf den Zielrechner transferiert werden oder in einem freigegebenen, sowohl Quell- als auch Zielrechner zugänglichen Verzeichnis gespeichert werden und von dort aus referenziert werden, siehe "Dateibasierte Datenbanken", weiter unten.

Übertragbarmachen von Pfaden

Wenn Sie beabsichtigen, das Mapping auf einem Server auszuführen, müssen Sie sicherstellen, dass es den jeweils geltenden Pfadkonventionen entspricht und dass eine unterstützte Datenbankverbindung verwendet wird.

Um Pfade auf Nicht-Windows-Betriebssysteme übertragen zu können, verwenden Sie beim Erstellen des Mappings in MapForce relative anstelle von absoluten Pfaden:

1. Öffnen Sie die gewünschte Mapping-Design-Datei (.mfd) mit MapForce unter Windows.
2. Wählen Sie im Menü **Datei** den Befehl **Mapping-Einstellungen** und deaktivieren Sie das Kontrollkästchen **Pfade im generierten Code absolut machen**, falls es aktiviert ist.
3. Öffnen Sie für jede Mapping-Komponente das Dialogfeld **Eigenschaften** (z.B. durch Doppelklick auf die Titelleiste) und ändern Sie alle Dateipfade von absoluten in relative um. Aktivieren Sie außerdem das Kontrollkästchen **Alle Dateipfade relativ zur MFD-Datei speichern**. Sie können alle Input-Dateien und Schemas aus praktischen Gründen in denselben Ordner wie das Mapping kopieren und diese einfach über den Dateinamen referenzieren.

Nähere Informationen zum Arbeiten mit relativen und absoluten Pfaden bei der Erstellung eines Mappings finden Sie in der Dokumentation zu MapForce,.

Beachten Sie, dass sowohl MapForce Server als auch FlowForce Server ein so genanntes Arbeitsverzeichnis ("working directory") unterstützen, anhand dessen alle relativen Pfade aufgelöst werden. Das Arbeitsverzeichnis wird zur Mapping-Laufzeit folgendermaßen definiert:

- in FlowForce Server durch Bearbeitung des Auftragsparameters "Working-directory".
- in der MapForce Server API über die Eigenschaft `workingDirectory` der COM und der .NET API oder über die Methode `setWorkingDirectory` der Java API.
- in der MapForce Server-Befehlszeile ist das Arbeitsverzeichnis das aktuelle Verzeichnis der Befehlszeile.

Datenbankverbindungen

Beachten Sie, dass ADO-, ADO.NET- und ODBC-Verbindungen auf Linux- und macOS-Rechnern nicht unterstützt werden. Wenn es sich beim Zielrechner daher um einen Linux- oder macOS-Rechner handelt, werden diese Verbindungen in JDBC konvertiert, wenn Sie das Mapping auf FlowForce Server bereitstellen oder das Mapping zu einer MapForce Server-Ausführungsdatei kompilieren. In diesem Fall haben Sie die folgenden Möglichkeiten, bevor Sie das Mapping bereitstellen oder es zu einer Server-Ausführungsdatei kompilieren:

- Erstellen Sie in MapForce eine JDBC-Verbindung zur Datenbank, bevor Sie das Mapping bereitstellen
- Füllen Sie in MapForce die JDBC-Datenbankverbindungsinformationen im Abschnitt "JDBC-spezifische Einstellungen" der Datenbankkomponente aus .

Wenn im Mapping eine native Verbindung zu einer PostgreSQL- oder SQLite-Datenbank verwendet wird, wird die native Verbindung beibehalten und es wird keine JDBC-Konvertierung durchgeführt. Wenn im Mapping eine Verbindung zu einer dateibasierten Datenbank wie Microsoft Access und SQLite hergestellt wird, müssen zusätzliche Konfigurationsschritte durchgeführt werden, siehe "Dateibasierte Datenbanken" weiter unten.

Für die Ausführung von Mappings mit JDBC-Verbindungen muss Java Runtime Environment oder Java Development Kit auf dem Server-Rechner installiert sein. Dabei kann es sich entweder um Oracle JDK oder einen Open Source Build wie z.B. Oracle OpenJDK handeln.

- Die JAVA_HOME-Umgebungsvariable muss auf das JDK-Installationsverzeichnis verweisen.
- Unter Windows hat ein Java Virtual Machine-Pfad aus der Windows Registry Vorrang vor der JAVA_HOME-Variablen.
- Die JDK-Plattform (64-Bit, 32-Bit) muss mit der von MapForce Server identisch sein, da Sie sonst eventuell einen Fehler erhalten, weil "kein Zugriff auf JVM besteht".

So richten Sie unter Linux oder macOS eine JDBC-Verbindung ein:

1. Laden Sie den vom Datenbankanbieter bereitgestellten JDBC-Treiber herunter und installieren Sie ihn auf dem Betriebssystem. Wählen Sie die 32-Bit-Version aus, wenn Ihr Betriebssystem mit 32 Bit ausgeführt wird, und die 64-Bit-Version, wenn Ihr Betriebssystem mit 64 Bit ausgeführt wird.
2. Definieren Sie für die Umgebungsvariablen den Pfad, unter dem der JDBC-Treiber installiert ist. Normalerweise müssen Sie die CLASSPATH-Variable und eventuell noch einige weitere Variablen definieren. Nähere Informationen dazu, welche Umgebungsvariablen konfiguriert werden müssen, finden Sie in der mit dem JDBC-Treiber mitgelieferten Dokumentation.

Anmerkung: Unter macOS werden die installierten JDBC-Bibliotheken im Verzeichnis **/Library/Java/Extensions** gesucht. Es wird daher empfohlen, den JDBC-Treiber unter diesem Pfad zu entpacken; andernfalls müssen Sie das System so konfigurieren, dass es die JDBC-Bibliothek unter dem Pfad sucht, unter dem Sie den JDBC-Treiber installiert haben.

Oracle Instant Client-Verbindungen auf macOS

Wenn Sie über den **Oracle Database Instant Client** eine Verbindung zu einer Oracle-Datenbank herstellen, gelten auf macOS die folgenden Voraussetzungen:

- Es muss Java 8.0 oder höher installiert sein. Sie können auch über die **JDBC Thin for All Platforms**-Bibliothek eine Verbindung herstellen, wenn auf dem Mac-Rechner eine Java-Version älter als Version 8 läuft. In diesem Fall können Sie die unten stehende Anleitung ignorieren.
- Oracle Instant Client muss installiert sein. Sie können den Oracle Instant Client von der offiziellen Oracle Download-Seite herunterladen. Beachten Sie, dass auf dieser Seite mehrere Instant Client-Pakete zum Download zur Verfügung stehen. Stellen Sie sicher, dass Sie ein Paket mit Oracle Call Interface (OCI)-Unterstützung (z.B. Instant Client Basic) auswählen. Stellen Sie außerdem sicher, dass Sie die 32-Bit-Version auswählen, falls Ihr Betriebssystem auf 32 Bit läuft, und die 64-Bit-Version, falls Ihr Betriebssystem auf 64 Bit läuft.

Nachdem Sie den Oracle Instant Client heruntergeladen und entpackt haben, bearbeiten Sie die mit dem Installer mitgelieferte Eigenschaftsliste (plist-Datei), sodass die folgenden Umgebungsvariablen auf den entsprechenden Treiberpfad verweisen, z.B.:

Variable	Beispielwert
CLASSPATH	/opt/oracle/instantclient_11_2/ojdbc6.jar:/opt/oracle/instantclient_11_2/ojdbc5.jar
TNS_ADMIN	/opt/oracle/NETWORK_ADMIN
ORACLE_HOME	/opt/oracle/instantclient_11_2

DYLD_LIBRARY_PATH	/opt/oracle/instantclient_11_2
PATH	\$PATH:/opt/oracle/instantclient_11_2

Anmerkung: Bearbeiten Sie die obigen Beispielwerte, sodass sie auf die Pfade verweisen, unter denen die Oracle Instant Client-Dateien auf Ihrem Betriebssystem installiert sind.

Dateibasierte Datenbanken

Dateibasierte Datenbanken wie Microsoft Access und SQLite sind in dem auf FlowForce Server bereitgestellten Paket oder in der kompilierten MapForce Server-Ausführungsdatei nicht enthalten. Wenn der Quellrechner daher nicht derselbe wie der Zielrechner ist, gehen Sie folgendermaßen vor:

1. Klicken Sie in MapForce mit der rechten Maustaste auf das Mapping und deaktivieren Sie das Kontrollkästchen **Pfade im generierten Code absolut machen**.
2. Klicken Sie mit der rechten Maustaste im Mapping auf die Datenbankkomponente und fügen Sie eine Verbindung zu einer Datenbankdatei über einen relativen Pfad hinzu. Eine einfache Methode, um Pfadprobleme zu vermeiden, ist, das Mapping-Design (die .mfd-Datei) im selben Verzeichnis wie die Datenbankdatei zu speichern und letztere vom Mapping aus nur über den Dateinamen (d.h. über einen relativen Pfad) zu referenzieren.
3. Kopieren Sie die Datenbankdatei in ein Verzeichnis auf dem Zielrechner (nennen wir es das "Arbeitsverzeichnis"). Dieses Verzeichnis benötigen Sie später, um das Mapping auf dem Server auszuführen (siehe unten).

Um ein solches Mapping auf dem Server auszuführen, wählen Sie eine der folgenden Methoden:

- Wenn das Mapping unter Verwaltung von FlowForce Server mit MapForce Server ausgeführt wird, konfigurieren Sie den FlowForce Server-Auftrag so, dass das zuvor erstellte Arbeitsverzeichnis verwendet wird. Die Datenbankdatei muss sich in diesem Arbeitsverzeichnis befinden. Ein Beispiel dazu finden Sie im Kapitel "Bereitstellen eines Auftrags als Web-Dienst" (https://www.altova.com/manual/de/flowforceserveradvanced/2025.2/index.html?fs_example_web_service.htm).
- Wenn das Mapping von MapForce Server alleine über die Befehlszeile ausgeführt wird, ändern Sie das aktuelle Verzeichnis in das Arbeitsverzeichnis (z.B. `cd pfad\zum\arbeitsverzeichnis`), bevor Sie den MapForce Server-Befehl `run` aufrufen.
- Wenn das Mapping über die MapForce Server API ausgeführt wird, definieren Sie das Arbeitsverzeichnis programmatisch, bevor Sie das Mapping ausführen. Zu diesem Zweck steht für das MapForce Server-Objekt in der COM und .NET API die Eigenschaft `workingDirectory` zur Verfügung. In der Java API steht dafür die Methode `setworkingDirectory` zur Verfügung.

Wenn es sich sowohl beim Quell- als auch beim Zielrechner um Windows-Rechner im lokalen Netzwerk handelt, wäre eine alternative Methode, das Mapping so zu konfigurieren, dass die Datenbankdatei folgendermaßen aus einem gemeinsamen, freigegebenen Verzeichnis gelesen wird:

1. Speichern Sie die Datenbankdatei in einem gemeinsamen freigegebenen Verzeichnis, auf das sowohl der Quell- als auch der Zielrechner Zugriff hat.
2. Klicken Sie mit der rechten Maustaste im Mapping auf die Datenbankkomponente und fügen Sie über einen absoluten Pfad eine Verbindung zur Datenbankdatei hinzu.

Globale Ressourcen

Wenn ein Mapping anstelle von direkten Pfaden oder Datenbankverbindungen Referenzen auf globale Ressourcen enthält, können Sie auch serverseitig globale Ressourcen verwenden. Wenn Sie ein Mapping zu einer MapForce Server-Ausführungsdatei (.mfx) kompilieren, bleiben die Referenzen auf globale Ressourcen intakt, so dass Sie diese zur Mapping-Laufzeit auf Seite des Servers bereitstellen können. Wenn Sie ein Mapping auf FlowForce Server bereitstellen, können Sie optional auswählen, ob darin die Ressourcen auf dem Server verwendet werden sollen.

Damit Mappings (oder, im Fall von FlowForce Server, Mapping-Funktionen) erfolgreich ausgeführt werden können, muss die eigentliche Datei, der eigentliche Ordner bzw. die Datenbankverbindungsinformationen, die Sie als globale Ressourcen bereitstellen mit der Server-Umgebung kompatibel sein. So muss z.B. bei Ausführung des Mappings auf einem Linux-Server bei Datei- und Ordnerpfaden die Linux-Pfadkonvention verwendet werden. Ebenso müssen als Datenbankverbindungen definierte globale Ressourcen auf dem Server-Rechner möglich sein.

Nähere Informationen dazu finden Sie unter [Globale Ressourcen](#) ⁴⁰.

XBRL-Taxonomiepakete

Wenn Sie ein Mapping bereitstellen, in dem XBRL-Taxonomiepakete referenziert werden, sammelt MapForce alle externen Referenzen aus dem Mapping und löst diese dann unter Verwendung der aktuellen Konfiguration und der aktuell installierten Taxonomiepakete auf. Falls es aufgelöste externe Referenzen gibt, die auf ein Taxonomiepaket verweisen, so wird das Taxonomiepaket zusammen mit dem Mapping bereitgestellt. FlowForce Server verwendet dieses Paket in dem Zustand, in dem es sich bei der Bereitstellung befand, um das Mapping auszuführen. Um das von FlowForce Server verwendete Taxonomiepaket zu aktualisieren, müssen Sie es in MapForce ändern und das Mapping erneut bereitstellen.

Beachten Sie, dass der Root-Katalog von MapForce Server einen Einfluss darauf hat, wie Taxonomien auf dem Zielrechner aufgelöst werden. Der Root-Katalog befindet sich unter dem folgenden Pfad relativ zum MapForce Server-Installationsverzeichnis: **etc/RootCatalog.xml**.

Mit einem Mapping bereitgestellte Taxonomiepakete werden verwendet, wenn der Root-Katalog von MapForce Server nicht bereits ein solches Paket enthält oder kein Paket enthält, das für dasselbe URL-Präfix definiert ist. Der Root-Katalog von MapForce Server hat Vorrang vor einer bereitgestellten Taxonomie.

Wenn MapForce Server alleine (ohne FlowForce Server) ausgeführt wird, kann der für das Mapping zu verwendende Root-Katalog folgendermaßen definiert werden:

- in der Befehlszeile durch Hinzufügen der Option `-catalog` zum Befehl `run`.
- in der MapForce Server API durch Aufruf der Methode `SetOption` und Bereitstellen des String `"catalog"` als erstes Argument und des Pfads zum Root-Katalog als zweites Argument.

Wenn in einem Mapping XBRL-Komponenten mit Tabellen-Linkbases verwendet werden, muss das Taxonomiepaket oder die Taxonomiepaketkonfigurationsdatei dem Mapping folgendermaßen zur Laufzeit zur Verfügung gestellt werden:

- Fügen Sie in der MapForce Server-Befehlszeile die Option `--taxonomy-package` oder `--taxonomy-packages-config-file` zum Befehl `run` hinzu.
- Rufen Sie über die MapForce Server API die Methode `SetOption` auf. Das erste Argument muss entweder `"taxonomy-package"` oder `"taxonomy-packages-config-file"` sein. Das zweite Argument muss der eigentliche Pfad zum Taxonomiepaket (oder zur Taxonomiepaketkonfigurationsdatei) sein.

3.2 Globale Ressourcen

Globale Altova-Ressourcen sind Aliasse für Datei-, Ordner und Datenbankressourcen. Jeder Alias kann mehrere Konfigurationen haben, wobei jede Konfiguration genau einer Ressource zugeordnet wird. Wenn Sie daher eine globale Ressource verwenden, können Sie zwischen ihren Konfigurationen wechseln. So könnten Sie etwa eine Datenbank-Ressource mit zwei Konfigurationen erstellen: Entwicklung und Produktion. Je nachdem, was Sie bezwecken möchten, können Sie zwischen diesen Konfigurationen wechseln. In MapForce Server könnten Sie Daten entweder aus der Entwicklungs- oder der Produktionsdatenbank abrufen, indem Sie die gewünschte Konfiguration zur Mapping-Laufzeit als Befehlszeilenparameter angeben.

Globale Ressourcen können applikationsübergreifend in verschiedenen Altova Applikationen verwendet werden (siehe *Unterabschnitt weiter unten*).

Globale Ressourcen in anderen Altova-Produkten

Wenn Datei-, Ordner- und Datenbankverbindungsinformationen als globale Ressourcen gespeichert werden, lassen sich diese in mehreren Altova-Applikationen wiederverwenden. Wenn Sie ein und dieselbe Datei z.B. häufig in verschiedenen Altova Desktop-Applikationen öffnen müssen, können Sie diese als globale Ressource definieren. Wenn Sie den Dateipfad ändern müssen, muss er nur an einer einzigen Stelle geändert werden. Derzeit können globale Ressourcen in den folgenden Altova-Produkten definiert und verwendet werden:

- [Altova Authentic](#)
- [DatabaseSpy](#)
- [MobileTogether Designer](#)
- [MapForce](#)
- [StyleVision](#)
- [XMLSpy](#)
- [FlowForce Server](#)
- [MapForce Server](#)
- [RaptorXML Server/RaptorXML+XBRL Server](#)

Nähere Informationen zur Erstellung von globalen Ressourcen finden Sie in der MapForce-Dokumentation im Kapitel "Globale Altova-Ressourcen".

Ressourcen in MapForce Server

Wenn Sie ein Mapping zu einer MapForce Server-Ausführungsdatei (**.mfx**) kompilieren, werden im Mapping verwendete globale Ressource-Referenzen beibehalten und nicht aufgelöst. Das bedeutet, Sie müssen diese Referenzen auf dem Server zur Verfügung stellen, damit das Mapping erfolgreich ausgeführt werden kann. In MapForce Server werden folgende Informationen benötigt, damit eine **.mfx**-Datei, in der globale Ressourcen verwendet werden, ausgeführt werden kann:

1. *Die Definitionsdatei für globale Ressourcen.* Die Datei hat auf dem Rechner, auf dem MapForce installiert ist, den Namen **GlobalResources.xml**. Sie befindet sich im Ordner **Dokumente\Altova**. Sie können diese Datei auf den Rechner, auf dem MapForce Server ausgeführt wird, kopieren und gegebenenfalls auch mehrere dieser Dateien erstellen.
2. *Der Name der globalen Ressourcenkonfiguration.* Jede globale Ressource hat eine Standardkonfiguration. Sie können auch zusätzliche Konfigurationen erstellen.

Die Definitionsdatei für globale Ressourcen und der Name der globalen Ressourcenkonfiguration werden in MapForce über die grafische Benutzeroberfläche definiert bzw. geändert. Diese werden in MapForce Server zur

Mapping-Laufzeit definiert (siehe unten).

- Wenn Sie das Mapping über die Befehlszeilenschnittstelle ausführen, definieren Sie nach dem Befehl `run` die Optionen `--globalresourceconfig` und `--globalresourcefile`, z.B:

```
C:\Programme (x86)\Altova\MapForceServer2025\bin\MapForceServer.exe run  
SomeMapping.mfx --globalresourcefile="C:  
\Users\me\Documents\Altova\GlobalResources.xml" --globalresourceconfig="Default"
```

- Wenn Sie das Mapping über die MapForce Server API ausführen, rufen Sie die Methode `SetOptions` zwei Mal auf, bevor Sie die `Run`-Methode aufrufen. Mit dem ersten Aufruf wird der Dateipfad zur Definitionsdatei für globale Ressourcen als Option bereitgestellt, mit dem zweiten wird der Name der globalen Ressourcenkonfiguration bereitgestellt. Nähere Informationen dazu finden Sie unter [MapForce Server API](#)⁷⁹.

3.3 Join-Optimierung

Mit Hilfe der Join-Optimierung lässt sich die Ausführung von Datenmappings, in denen große Datenmengen gefiltert oder verknüpft werden, beschleunigen.

Bei der Join-Optimierung werden verschachtelte, bei der Ausführung des Mappings auftretende interne Schleifen eliminiert. Verschachtelte Schleifen entstehen, wenn bei einem Mapping jedes Datenelement einer Datengruppe so oft iteriert wird, wie es in einer zweiten Datengruppe vorkommt. Beachten Sie, dass es aufgrund des Mapping-Designs normal für den Mapping-Ausführungsprozessor* ist, dass an verschiedenen Datenelementsequenzen eine Schleifenverarbeitung (Iteration) durchgeführt wird. Wenn es zu unabhängigen verschachtelten Schleifen kommt (d.h. bei Schleifenverarbeitungen, die über andere Schleifen iterieren), kann eine Join-Optimierung, die die Ausführungsdauer des Mappings erheblich verringern würde, sinnvoll sein. Verschachtelte Schleifen fallen bei der Ausführung von Mappings mit wenigen Input-Daten kaum ins Gewicht. Im Fall von Mappings, in denen Dateien oder Datenbanken mit sehr vielen Datensätzen verarbeitet werden, kann dies jedoch zu Problemen führen.

* Der Ausführungsprozessor eines Mappings kann MapForce, MapForce Server oder ein mit MapForce generiertes C#-, C++- oder Java-Programm sein. Die Join-Optimierung steht ausschließlich in der MapForce Server Advanced Edition zur Verfügung.

Um MapForce Server als Ausführungszielprozessor zu definieren, klicken Sie in MapForce auf die Symbolleiste-Schaltfläche BUILT-IN (). Damit gewährleisten Sie außerdem, dass in Ihrem Mapping der größte verfügbare Funktionsumfang genutzt werden kann. Wenn Sie eine andere Transformationssprache auswählen, werden bestimmte MapForce-Funktionalitäten eventuell in dieser Sprache nicht unterstützt.

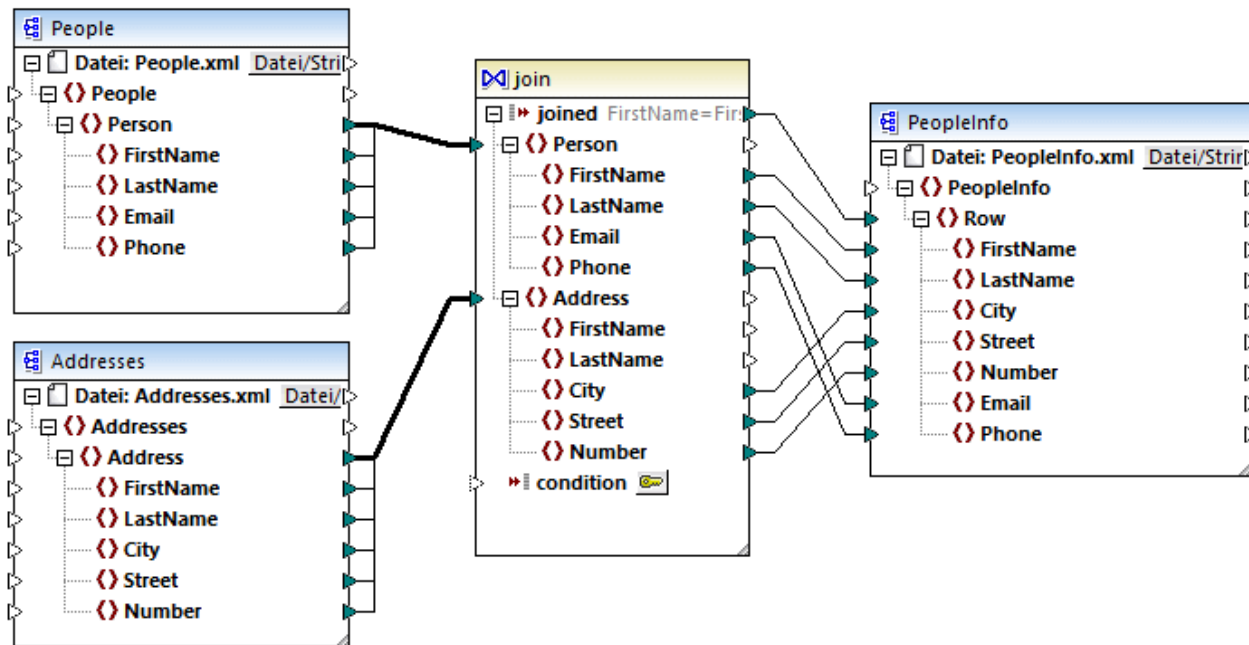
Wie oben erwähnt, geht es bei Join-Optimierungen in erster Linie darum, verschachtelte Schleifen effizient zu verarbeiten. Wir wollen nun genauer analysieren, wie verschachtelte Schleifen überhaupt entstehen.

Typischerweise kommt es dann zu verschachtelten Schleifen, wenn das Mapping mindestens eine Join-Komponente enthält und der SQL JOIN-Modus** nicht möglich ist.

** Unter bestimmten Bedingungen kann ein spezieller Ausführungsmodus in MapForce, nämlich der "SQL Join-Modus", verwendet werden. Dies gilt nur für Mappings, in denen Daten aus einer Datenbank ausgelesen werden. Wenn Daten auf diese Art mittels Join verknüpft werden, wird die Join-Operation von der Datenbank durchgeführt (d.h. es wird ein SQL JOIN durchgeführt). Dadurch muss der Mapping-Ausführungsprozessor keine verschachtelten Schleifen mehr durchführen. Nähere Informationen zum SQL Join-Modus finden Sie in der MapForce-Dokumentation (<https://www.altova.com/de/documentation.html>).

In der Abbildung unten sehen Sie ein (mit Altova MapForce erstelltes) Beispiel-Mapping, in dem Daten aus zwei XML-Dateien mit Hilfe einer Join-Komponente miteinander kombiniert werden. Sie finden dieses Mapping auf dem Rechner, auf dem MapForce installiert ist, unter dem folgenden Pfad: ..

!Dokumente\Altova\MapForce2025\MapForceExamples\Tutorial\JoinPeopleInfo.mfd. Einige Personendaten (People) stehen nur in der ersten XML-Datei (**Email**, **Phone**) zur Verfügung, während andere Daten nur in der zweiten XML-Datei (**City**, **Street**, **Number**) verfügbar sind. Ziel des Mappings ist es, die zusammengeführten Daten aller Personen, bei denen **FirstName** und **LastName** in beiden Quellstrukturen miteinander übereinstimmen, in die XML-Zieldatei zu schreiben.



JoinPeopleInfo.mfd

Eine Join-Komponente in MapForce verbindet Datenelemente in zwei Datengruppen gemäß einer benutzerdefinierten Bedingung. Zu diesem Zweck wird jedes Datenelement in der Datengruppe 1 mit jedem Datenelement in der Datengruppe 2 verglichen. Die Gesamtanzahl der Vergleiche bildet das Kreuzprodukt (das kartesische Produkt) beider Datengruppen. Wenn etwa die erste Datengruppe 50 Datenelemente und die zweite 100 enthält, so werden insgesamt 5000 Vergleiche (50×100) durchgeführt. Im obigen Mapping entsprechen die verglichenen Gruppen allen Instanzdatenelementen der zwei mit der Join-Komponente verbundenen XML-Strukturen.

Anmerkung: Eine Join-Optimierung (eine Funktionalität der MapForce Server Advanced Edition) ist nicht mit Join-Komponenten (einer Funktionalität von MapForce) zu verwechseln. Nähere Informationen zu Join-Komponenten finden Sie in der MapForce-Dokumentation (<https://www.altova.com/de/documentation.html>).

Von der Verarbeitungsleistung her gesehen, würden Mappings, die verschachtelte Schleifen enthalten, mehr Verarbeitungszeit in Anspruch nehmen. Stellen Sie sich vor, Sie haben einen Fall, in dem beide miteinander verknüpften Datengruppen Millionen von Datensätzen enthalten. Dies wirkt sich natürlich auf die Verarbeitungsgeschwindigkeit aus. Hier wäre eine Join-Optimierung sinnvoll. Eine Join-Optimierung verhält sich grob gesprochen wie ein Datenbankprozessor, der so optimiert ist, dass er extrem große Datengruppen indiziert, bloß dass eine Join-Optimierung, wie im Mapping oben gezeigt, nicht nur mit Daten aus Datenbanken operiert. Mit einer Join-Optimierung werden verschachtelte Schleifen unabhängig von der Datenart eliminiert, indem, wo dies möglich ist, interne Lookup-Tabellen erstellt werden, die zur Mapping-Laufzeit abgefragt werden. Dadurch wird die Mapping-Verarbeitungsgeschwindigkeit erheblich verbessert und weniger Zeit für die Ausführung des Mappings benötigt.

Anmerkung: Bei Join-Optimierungen dauert die Mapping-Verarbeitung weniger lang, doch wird normalerweise auch mehr Arbeitsspeicher in Anspruch genommen. Denken Sie daran, dass die Arbeitsspeicherbeanspruchung von verschiedenen komplexen Faktoren abhängig ist; aus diesem Grund kann das beobachtete Verhalten je nach Fall unterschiedlich sein.

Mit Hilfe von Join-Optimierungen können nicht nur Mappings mit Joins, sondern auch Mappings, die Filterkomponenten enthalten, beschleunigt werden. Ein Filter verarbeitet in MapForce eine Sequenz von Datenelementen (d.h. er überprüft für jede Instanz des mit dem **node/row**-Input verbundenen Datenelements eine angegebene Boolesche Bedingung). Wenn die Boolesche Bedingung mit einer Funktion verbunden ist, die selbst wiederum über eine weitere Sequenz von Datenelementen iterieren muss und es der Mapping-Kontext verlangt, so kann es zu einer ähnlichen Situation wie bei einem Join kommen. Wenn mit dem Filter ein Kreuzvergleich jedes Datenelements in zwei Datengruppen durchgeführt werden muss, so ist eine Join-Optimierung hier sinnvoll.

Damit die Join-Optimierung in einem Mapping genutzt werden kann, muss es mit der MapForce Server Advanced Edition ausgeführt werden. Um ein Mapping mit der MapForce Server Advanced Edition auszuführen, öffnen Sie es in MapForce und kompilieren Sie es mit dem Menübefehl **Datei | Zu MapForce Server-Ausführungsdatei kompilieren** zu einer Mapping-Ausführungsdatei (.mfx). Führen Sie die .mfx-Datei anschließend entweder mit einer API-Methode in der Programmiersprache Ihrer Wahl aus oder verwenden Sie den Befehl run der Befehlszeilenschnittstelle (siehe auch [Funktionalität](#) ³¹).

3.4 Anmeldeinformationen

Mit Hilfe von Anmeldeinformationsobjekten können Authentifizierungsdaten (wie Benutzernamen, Passwörter und OAuth-Authentifizierungsinformationen) zwischen verschiedenen Mapping-Ausführungsumgebungen auf sichere Art portabel gemacht werden. Anmeldeinformationen eignen sich für die Verwendung in Mappings, für die eine einfache HTTP-Authentifizierung oder eine OAuth 2.0-Authentifizierung erforderlich ist. Anmeldeinformationen können sowohl in MapForce als auch FlowForce Server definiert werden. Wenn die Anmeldeinformationen in MapForce definiert wurden, können diese, ähnlich wie Mappings, optional auch auf FlowForce Server bereitgestellt werden.

Nachdem Sie das Mapping zu einer MapForce Server-Ausführungsdatei (.mfx) kompiliert haben, führt MapForce Server die .mfx-Datei gemäß der von Ihnen zum Zeitpunkt des Designs des Mapping getroffenen Auswahl aus.

Wenn Sie bei der Erstellung der Anmeldeinformationen in MapForce das Kontrollkästchen **Verschlüsselt in MapForce Server-Ausführungsdatei und Mapping-Bereitstellung inkludieren** aktiviert haben, verwendet MapForce Server zur Mapping-Laufzeit die in der .mfx-Datei gespeicherten Anmeldeinformationen. Das bedeutet, Sie können das Mapping mit einem Befehl wie dem folgenden ausführen:

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

Hierbei steht **<exec>** für den Pfad zur ausführbaren MapForce Server-Datei. Dieser Pfad kann entweder ein absoluter sein oder Sie können nur den Namen der ausführbaren Datei eingeben, wenn das aktuelle Verzeichnis dasselbe wie das der ausführbaren Datei ist.

Wenn Sie in MapForce nur den Namen der Anmeldeinformationen (ohne Benutzernamen und Passwort) eingegeben haben, müssen Sie diese Informationen mit Hilfe der Befehlszeilenoption **--credential** des Befehls run zur Mapping-Laufzeit explizit bereitstellen. Auf diese Art können Sie z.B. in der Produktion andere Anmeldeinformationen als zum Zeitpunkt des Mapping-Designs verwenden. Die Option **--credential** hat die Form **--credential=KEY:VALUE**., wobei

- **KEY** der Name der in MapForce definierten Anmeldeinformationen ist.
- **VALUE** eine Anmeldeinformationseigenschaft oder eine durch das Zeichen (&) getrennte Liste von Eigenschaften ist. Bei Anmeldeinformationen vom Typ "Passwort" sind die möglichen Eigenschaften **username** und **password**. Bei Anmeldeinformationen vom Typ OAuth 2.0 ist die einzige unterstützte Eigenschaft die Eigenschaft **oauth:token**.
- Die eigentlichen Eigenschaftswerte werden genau wie Abfrageparameter in einer URL mit Hilfe des Zeichens "=" bereitgestellt.

Zum Beispiel:

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

Der Wert der Option **--credential** im Codefragment oben wurde in Anführungszeichen gesetzt, damit der Wert als Literal behandelt wird, da der Benutzername und das Passwort durch ein Kaufmännisches Und-Zeichen voneinander getrennt sind.

Wenn mehrere benannte Anmeldeinformationsobjekte erforderlich sind, damit Ihr Mapping ausgeführt werden kann, können Sie die Option **--credential** mehrmals definieren.

Anmeldeinformationen, die als Befehlszeilenoptionen bereitgestellt werden, haben Vorrang vor gespeicherten Anmeldeinformationen.

Wenn Sie die Option **Verschlüsselt in MapForce Server-Ausführungsdatei und Mapping-Bereitstellung inkludieren** nicht aktiviert haben, fehlen die Felder mit den sensiblen Daten, d.h. Sie müssen in der Befehlszeile das Passwort angeben, während Sie das Anmeldeinformationsobjekt anhand seines Namens referenzieren, z.B:

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

Die folgenden Felder gelten als Felder, die sensible Daten enthalten:

- **Passwort** (bei Anmeldeinformationen vom Typ "Passwort")
- **Client Secret, Access Token und Refresh Token** (bei Anmeldeinformationen vom Typ "OAuth 2.0")

Bei Mappings, für die eine OAuth 2.0-Autorisierung erforderlich ist, wird in der MapForce Server-Befehlszeile zur Mapping-Laufzeit ein OAuth 2.0 Access Token als Input akzeptiert. Beachten Sie, dass es in der MapForce Server-Befehlszeile keine interaktive Benutzeroberfläche gibt. Daher müssen Sie das OAuth 2.0 Access Token in diesem Fall auf externem Weg (z.B. durch Anforderung von MapForce aus) abrufen. Dies ist jedoch nicht notwendig, wenn MapForce Server unter FlowForce Server-Verwaltung ausgeführt wird, da FlowForce Server zur Laufzeit selbst ein neues OAuth 2.0 Access Token anfordern kann.

Ein Mapping kann solange mit gespeicherten Anmeldeinformationen über die Befehlszeile ausgeführt werden, als das gespeicherte OAuth 2.0 Token noch nicht abgelaufen ist oder vom Webservice-Anbieter widerrufen wurde. Andernfalls müssen Sie über die Option **--credential** ein neues (auf externem Weg abgerufenes) OAuth 2.0 Access Token angeben, z.B:

```
<exec> run mapping.mfx --  
credential=my_oauth_credential:oauth:token=jdsafkajlkewsaiurthczv904215-jhd
```

wobei:

- **my_oauth_credential** der Name der mit MapForce erstellten OAuth 2.0-Anmeldeinformationen ist.
- **oauth:token** die Methode ist, um MapForce Server mitzuteilen, dass zur Laufzeit ein neues OAuth 2.0 Access Token bereitgestellt wird.

MapForce Server API

Die MapForce Server API bietet Methoden, um Anmeldeinformationen zu erstellen, Eigenschaften zu Anmeldeinformationen hinzuzufügen und Anmeldeinformationen zu schließen, nachdem Sie diese fertig deklariert haben. Der folgende Code ist ein Beispiel dafür, wie Passwort-Anmeldeinformationen in einem C#-Programm, das ein Mapping ausführt, normalerweise deklariert werden:

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

```
// Set the credential name as it was defined in MapForce
objMFS.BeginCredential("mycredential");
// Add the credential properties
objMFS.AddCredentialProperty("username", "altova");
objMFS.AddCredentialProperty("password", "b45ax78!");
// Close the credential
objMFS.EndCredential();
```

Um eine OAuth 2.0-Autorisierung von einem Programm, das ein Mapping ausführt, durchzuführen, muss der Eigenschaftsname der Anmeldeinformationen, wie unten gezeigt, als **oauth:token** definiert werden:

```
//Create a MapForce Server object
Altova.MapForceServer.Server objMFS = new Altova.MapForceServer.Server();
// Set the credential name as it was defined in MapForce
objMFS.BeginCredential("my_oauth_credential");
// Add the credential properties
objMFS.AddCredentialProperty("oauth:token", "jdsafklkajlkewsaiurthczv904215-jhd");
// Close the credential
objMFS.EndCredential();
```

Wenn mehrere Anmeldeinformationen im Mapping benötigt werden, gehen Sie vor, wie oben gezeigt, um ganz nach Bedarf mehrere Anmeldeinformationsgruppen hinzuzufügen. Sobald Sie alle erforderlichen Anmeldeinformationen deklariert haben, können Sie die Mapping-Ausführungsdatei standardmäßig durch Aufruf der `Run()` Methode ausführen. Nähere Informationen dazu finden Sie in der [API-Referenz](#)⁷⁹.

3.4.1 Beispiel: OAuth 2.0-Autorisierung

In diesem Beispiel wird gezeigt, wie Sie einen REST-Webservice, für den eine OAuth 2.0-Autorisierung erforderlich ist, aufrufen. Die Client-Applikation ist eine MapForce Server-Ausführungsdatei (.mfx). Darin werden über die Google Calendar API (<https://developers.google.com/calendar/>) Kalenderereignisse abgerufen. Aus Gründen der Einfachheit ruft die .mfx-Datei die Kalenderinformationen im Istzustand ab und gibt nur das rohe JSON-Result aus, ohne dieses weiter zu verarbeiten.

Voraussetzungen:

- MapForce Enterprise Edition
- MapForce Server Advanced Edition
- Um dieses Beispiel durchspielen zu können, benötigen Sie ein Google-Konto. Wenn Sie einen anderen Webservice aufrufen möchten, fordern Sie OAuth 2.0-Anmeldeinformationen von Ihrem Webservice-Anbieter an und verwenden Sie diese in der unten stehenden Anleitung.

Anforderung der OAuth 2.0-Anmeldeinformationen

Wenn Sie die erforderlichen OAuth 2.0-Anmeldeinformationen für den Aufruf des Webservices bereits zur Verfügung haben, können Sie diesen Schritt überspringen. Andernfalls ist der genaue Ablauf vom Anbieter des Webservices, der von Ihrem Mapping aufgerufen wird, abhängig. Um die Google Calendar API, wie in diesem Beispiel gezeigt, aufzurufen, gehen Sie folgendermaßen vor:

1. Melden Sie sich bei der Google API-Konsole (<https://console.developers.google.com/>) an.
2. Erstellen Sie ein neues Projekt.

Google APIs

Neues Projekt

In Ihrem Kontingent sind noch 12 projects offen. Fordern Sie eine Erhöhung an oder löschen Sie Projekte. [Weitere Informationen](#)

[MANAGE QUOTAS](#)

Projektname *

Project ID: mapforcedemo. Sie kann später nicht mehr geändert werden.

[BEARBEITEN](#)

Speicherort * [DURCHSUCHEN](#)

Übergeordnete Organisation oder übergeordneter Ordner

[ERSTELLEN](#) [ABBRECHEN](#)

3. Klicken Sie auf **OAuth-Zustimmungsbildschirm**.
4. Wählen Sie als Benutzertyp **Extern**, es sei denn, Sie haben ein G Suite-Konto, über das Sie nur Benutzern in Ihrem Unternehmen API-Zugriff einräumen können.

The screenshot shows the Google APIs console interface. On the left, a sidebar lists navigation options: Dashboard, Bibliothek, Anmeldeinformationen, OAuth-Zustimmungsbildschirm (selected), Domainbestätigung, and Seitennutzungsvereinbarung... The main content area is titled 'OAuth-Zustimmungsbildschirm'. It contains instructions: 'Wählen Sie aus, wie Sie Ihre Anwendung konfigurieren und registrieren möchten, einschließlich Ihrer Anwendergruppe. Sie können nur eine Anwendung mit Ihrem Projekt verknüpfen.' Below this, the 'User Type' section has two radio buttons: 'Intern' (unselected) and 'Extern' (selected). The 'Extern' option is described as 'Für jeden Nutzer mit einem Google-Konto verfügbar.' At the bottom right is a blue 'ERSTELLEN' button.

5. Geben Sie "mapforce-demo" als Applikationsnamen ein und speichern Sie die Einstellungen.

This screenshot shows the 'OAuth-Zustimmungsbildschirm' configuration page. It includes a paragraph explaining the consent screen's purpose. Below, the 'Bestätigungsstatus' is set to 'Nicht veröffentlicht'. The 'Name der Anwendung' field is filled with 'mapforce-demo'. The 'Logo der Anwendung' section has a text input field with 'Lokale Datei zum Hochladen' and a 'Durchsuchen' button.

6. Klicken Sie auf **Anmeldeinformationen erstellen** wählen Sie anschließend **OAuth-Client-ID** aus.
7. Geben Sie als Applikationstyp **Desktopanwendung** und als Client-Namen "MapForce Client" ein.



← OAuth-Client-ID erstellen

Eine Client-ID wird zur Identifizierung einer einzelnen Anwendung bei Googles OAuth-Servern verwendet. Wenn Ihre Anwendung auf mehreren Plattformen ausgeführt wird, erfordert jede eine eigene Client-ID. Weitere Informationen finden Sie unter [OAuth 2.0 einrichten](#).

Anwendungstyp *
Desktopanwendung

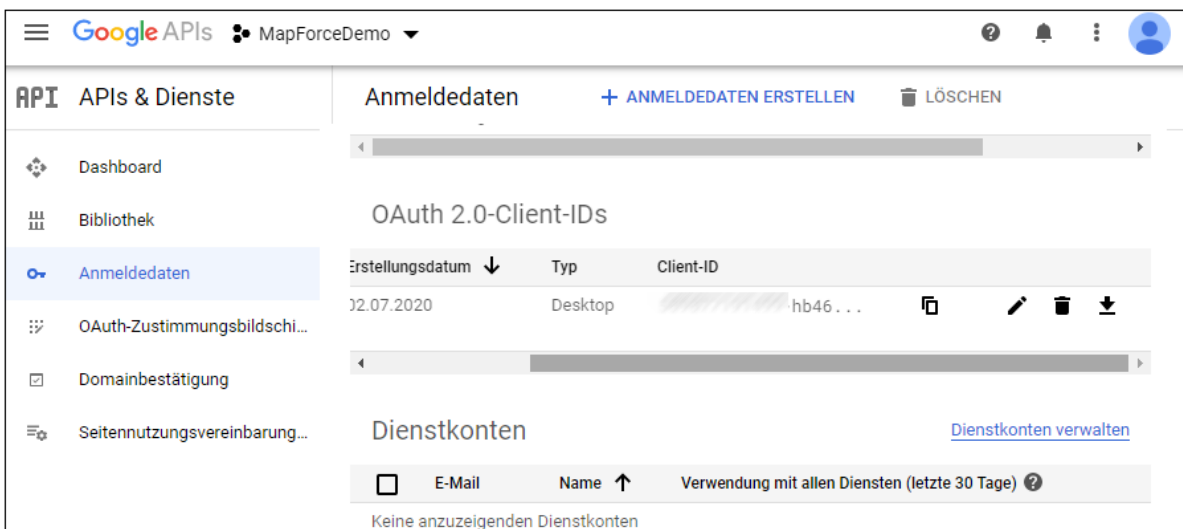
[Weitere Informationen](#) zu OAuth-Clienttypen

Name *
MapForce Client

Der Name Ihres OAuth 2.0-Clients. Dieser Name wird nur zum Identifizieren des Clients in der Console verwendet und wird Endnutzern nicht angezeigt.

ERSTELLEN ABBRECHEN

8. Klicken Sie auf **Erstellen**. Daraufhin wird die Client-ID erstellt und steht auf der Seite **Anmeldedaten** zur Verfügung.



Google APIs MapForceDemo

APIs & Dienste

Anmeldedaten + ANMELDEDATEN ERSTELLEN LÖSCHEN

OAuth 2.0-Client-IDs

Erstellungsdatum ↓	Typ	Client-ID
02.07.2020	Desktop	hb46...

Dienstkonten [Dienstkonten verwalten](#)

Keine anzuzeigenden Dienstkonten

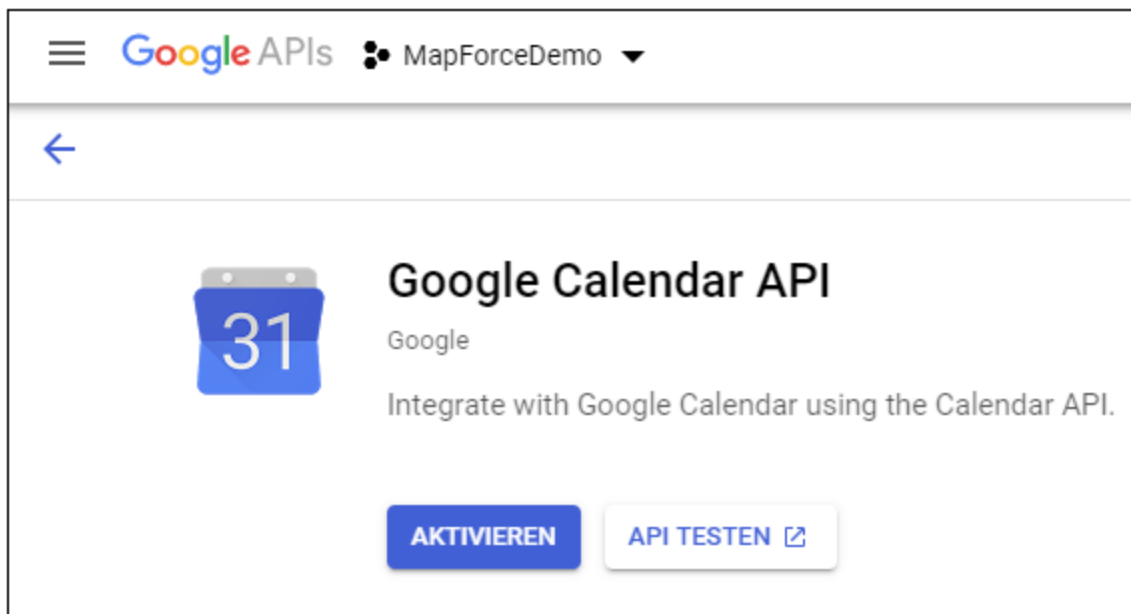
9. Klicken Sie auf , um die OAuth 2.0-Autorisierungsinformationen als JSON-Datei herunterzuladen.

Sie haben nun die folgenden OAuth 2.0-Autorisierungsinformationen von der Google Console API erhalten:

1. Autorisierungsendpunkt
2. Token-Endpunkt
3. Client ID
4. Client Secret

Aktivieren der Google Calendar API

Um Aufrufe von Clients zu akzeptieren, muss die in diesem Beispiel verwendete Google Calendar API aktiviert werden. Klicken Sie in der Google API-Konsole auf **Bibliothek**, suchen Sie die Google Calendar API und aktivieren Sie sie:



In diesem Beispiel werden wir die **list**-Methode der **Events** Entity aufrufen. Ausführliche Informationen über diese API-Methode finden Sie unter <https://developers.google.com/calendar/v3/reference/events/list>. Beachten Sie die folgenden wichtigen Punkte:

1. Wie in der Dokumentation beschrieben, muss die Methode durch Senden eines GET Request an <https://www.googleapis.com/calendar/v3/calendars/calendarId/events> aufgerufen werden, wobei **calendarId** der Identifier eines Google Calendar ist. Der Request-Parameter **calendarId** wird in der Folge später über MapForce konfiguriert.
2. Für den Aufruf der API-Methode ist mindestens einer der folgenden Geltungsbereiche erforderlich:
 - <https://www.googleapis.com/auth/calendar.readonly>
 - <https://www.googleapis.com/auth/calendar>
 - <https://www.googleapis.com/auth/calendar.events.readonly>
 - <https://www.googleapis.com/auth/calendar.events>

Ihr Mapping muss während der OAuth 2-Autorisierung einen dieser Geltungsbereiche angeben. Auch dieser wird in einem späteren Schritt konfiguriert. Für dieses Beispiel ist der erste Geltungsbereich "read-only" ausreichend.

Anforderung eines Autorisierungstoken

Um in MapForce eine Vorschau auf das Mapping anzeigen zu können, benötigen Sie die OAuth 2.0-Autorisierungsinformationen für das Mapping und müssen, wie unten gezeigt, ein Autorisierungstoken anfordern.

1. Klicken Sie in MapForce mit der rechten Maustaste in einen leeren Bereich des Mappings und wählen Sie im Kontextmenü den Befehl **Anmeldeinformationen-Manager öffnen**.
2. Klicken Sie auf **+ Anmeldeinformationen hinzufügen**.
3. Geben Sie einen Namen ein (in diesem Beispiel "my.oauth") und wählen Sie als Typ **OAuth 2** aus.
4. Füllen Sie die Textfelder **Autorisierungsendpunkt**, **Token-Endpunkt**, **Client ID**, **Client Secret** mit den entsprechenden Werten aus der zuvor heruntergeladenen JSON-Datei aus.
5. Geben Sie `https://www.googleapis.com/auth/calendar.readonly` in das Textfeld **Geltungsbereich** ein.
6. Belassen Sie alle anderen Einstellungen unverändert.

Anmeldeinformationen hinzufügen

Name:

Typ: ☐ Passwort ☒ OAuth 2

Autorisierungsendpunkt:

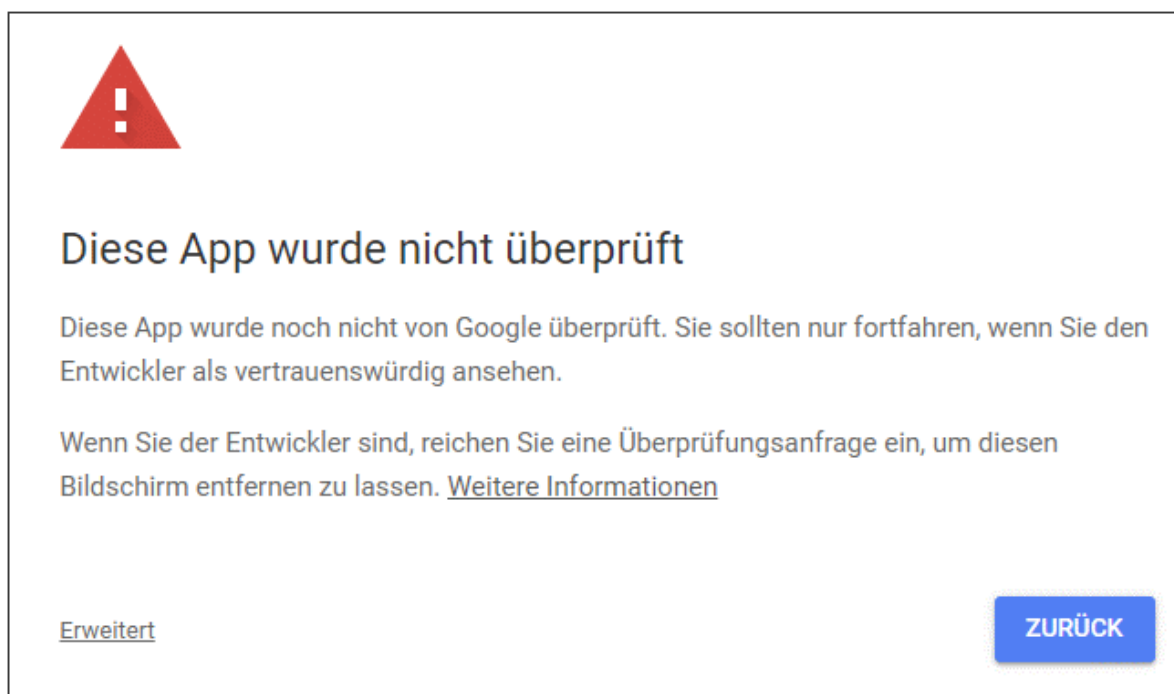
Token-Endpunkt:

Client ID:

Client Secret:

Geltungsbereich:

7. Klicken Sie auf **Access Token anfordern**, um das Token vom Autorisierungsserver (in diesem Beispiel Google) anzufordern. Daraufhin wird ein Browser-Fenster geöffnet, in dem Sie gebeten werden, sich mit Ihrem Google-Konto zu verbinden.
8. Melden Sie sich bei Ihrem Google-Konto an. Da Sie noch keine App-Überprüfungsanforderungen an Google gesendet haben, wird die folgende Seite angezeigt.



9. Klicken Sie auf **Erweitert** und anschließend auf **mapforce-demo öffnen (unsicher)**.

 Über Google anmelden

Auswahl bestätigen



Sie haben **mapforce-demo** folgende Berechtigungen erteilt:

☒ Kalender anzeigen

mapforce-demo vertrauen?

Eventuell teilen Sie vertrauliche Informationen mit dieser Website oder App. In den Nutzungsbedingungen und der Datenschutzerklärung von mapforce-demo erfahren Sie alles zum Umgang mit Ihren Daten. In Ihrem **Google-Konto** können Sie die Zugriffsberechtigungen jederzeit einsehen oder entfernen.

[Weitere Informationen zu den Risiken](#)

Abbrechen

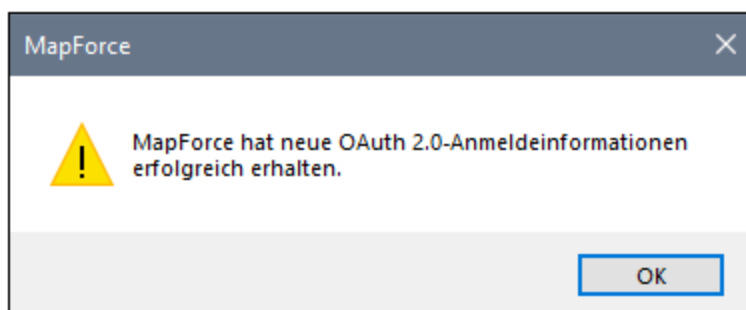
Zulassen

10. Klicken Sie auf **Zulassen**. Daraufhin wird eine Bestätigung im Browser angezeigt.

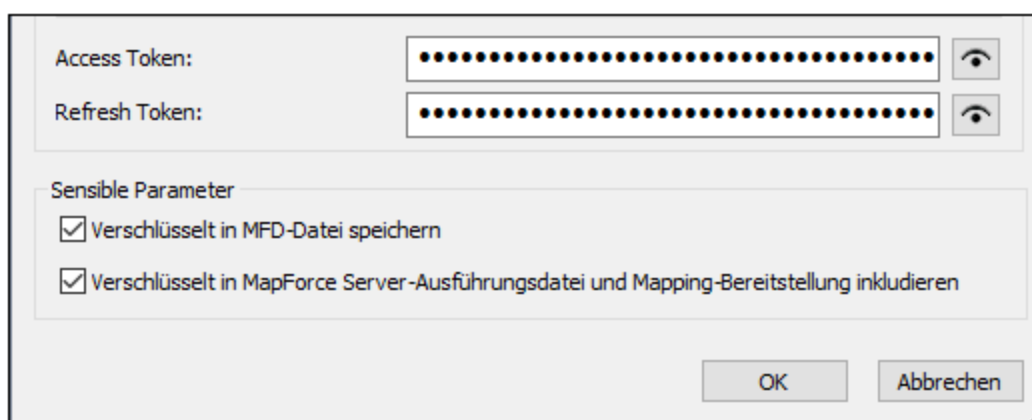
OAuth 2.0-Autorisierungscode wurde abgerufen.

Zurück zu Altova MapForce.

MapForce zeigt ebenfalls eine Meldung an, dass der OAuth 2.0-Autorisierungscode erfolgreich abgerufen wurde.



11. Klicken Sie auf **OK**. Beachten Sie, dass die Felder **Access Token** und **Refresh Token** nun ausgefüllt wurden.



12. Speichern Sie das Mapping unter dem Namen **GetCalendarEvents.mfd**.

In diesem Tutorial wurde das Kontrollkästchen **Verschlüsselt in MFD-Datei speichern** im Dialogfeld "Anmeldeinformationen bearbeiten" aktiviert. Daher werden die sensiblen Daten in den Feldern **Client Secret**, **Autorisierungs-Token** und **Refresh Token** beim Speichern des Mappings in verschlüsselter Form in der Mapping-Design-Datei (.mfd) gespeichert.

Beachten Sie, dass das Autorisierungs-Token nach einem gewissen Zeitraum abläuft. Wenn dies passiert, können Sie das Mapping nicht mehr ausführen (derzeit wurde noch kein Mapping erstellt, dies geschieht in einem späteren Schritt). Wenn Sie manuell einen neuen Autorisierungscode anfordern möchten, klicken Sie auf **Access Token anfordern** gehen Sie vor, wie unten beschrieben.

Erstellen des Webservice-Aufrufs

Das bisher erstellte Mapping **GetCalendarEvents.mfd** hat noch keine Funktionen. Es enthält bisher nur OAuth 2.0-Anmeldeinformationen, die den Zugriff auf die Google Calender API aktivieren.

Wir werden nun den Webservice-Aufruf in MapForce folgendermaßen erstellen:

1. Öffnen Sie das Mapping **GetCalendarEvents.mfd**.
2. Klicken Sie im Menü **Einfügen** auf **Webservice-Funktion**. Daraufhin wird das Dialogfeld "Webservice-Call-Einstellungen" aufgerufen.
3. Klicken Sie auf **Manuell**.

4. Wählen Sie als Request-Methode **GET** aus und geben Sie die URL zu dem im vorigen Schritt erwähnten Webservice ein:
<https://www.googleapis.com/calendar/v3/calendars/calendarId/events>.
5. Da **calendarId** ein Platzhalter ist, der in Form eines Parameters angegeben werden muss, setzen Sie ihn in geschweifte Klammern, wie unten gezeigt.

Webservice-Call-Einstellungen

Service-Definition

☐ WSDL

☐ OpenAPI

☒ Manuell

Aus WADL-Datei importieren... Von URL importieren...

Request-Methode: GET

Verbindungseinstellungen

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

☐ Dynamische URL (wird durch Mapping bereitgestellt)

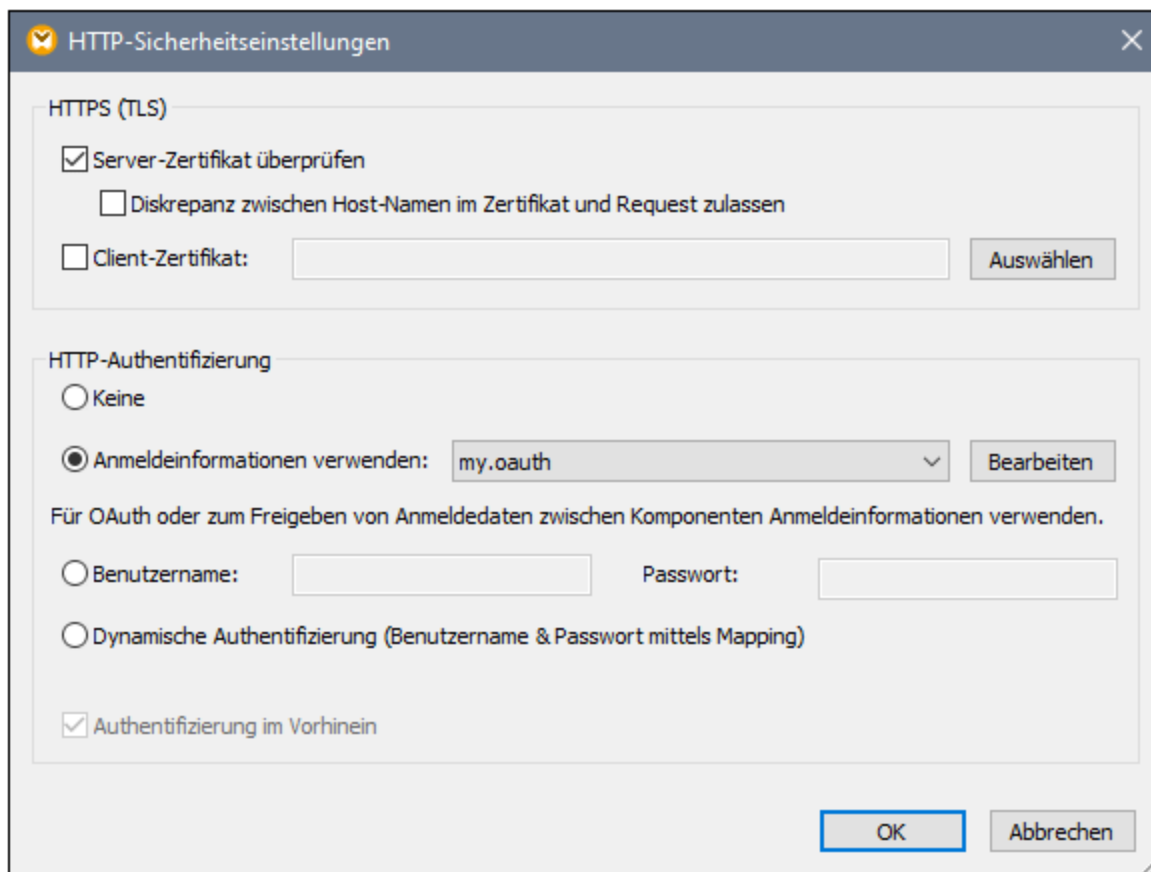
Timeout: 40 Sekunden ☐ Unendlich

6. Klicken Sie auf die Schaltfläche **Parameter hinzufügen** und definieren Sie die folgenden Parameterinformationen:

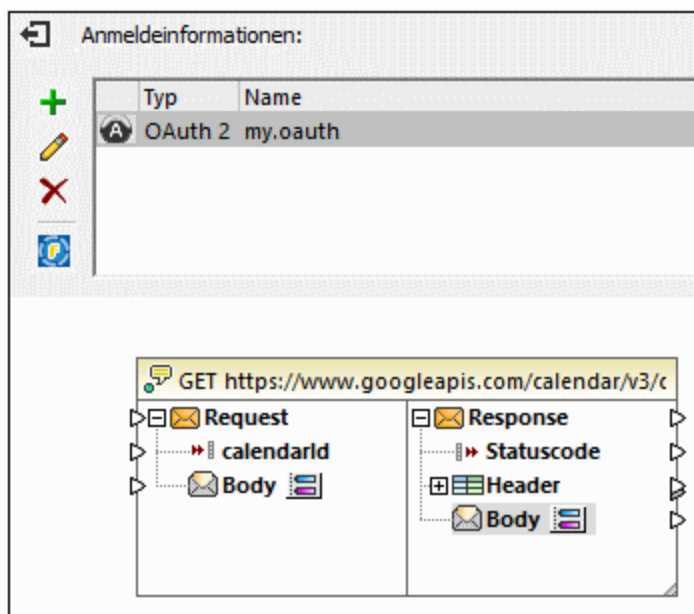
Parameter:							
Name	Stil	Typ	Mapbar	Fester Wert	Obligatorisch	Wiederholt	Beschreibung
calendarId	Vorlage	string	☒		☒	☐	

Aufgrund des Stils "Vorlage" in der Konfiguration oben kann der in geschweifte Klammern gesetzte Teil der URL zur Laufzeit durch den Parameterwert ersetzt werden. "Mapbar" bedeutet, dass der Wert über das Mapping bereitgestellt werden kann (z.B. über eine Konstante oder eventuell einen Input-Parameter). Außerdem wurde der Parameter als "Obligatorisch" gekennzeichnet, da der API-Aufruf nicht ohne diesen Parameter durchgeführt werden kann.

7. Klicken Sie neben **HTTP-Sicherheitseinstellungen** auf **Bearbeiten**.
8. Aktivieren Sie im Dialogfeld "HTTP-Sicherheitseinstellungen" die Option **Anmeldeinformationen verwenden** und wählen Sie den zuvor konfigurierten Datensatz "my.oauth" aus.



Der bisher konfigurierte Webservice wird im Mapping folgendermaßen angezeigt:

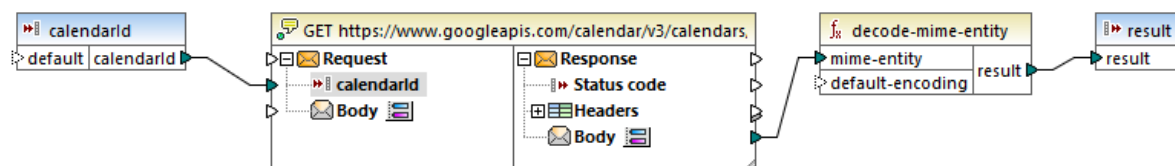


Sie können das Design nun folgendermaßen fertig stellen:

1. Klicken Sie im Menü **Einfügen** auf **Input-Komponente einfügen** und konfigurieren Sie die Komponente folgendermaßen:

Wie oben gezeigt, hat die Input-Komponente den Design-Zeit-Wert "primary". Laut API-Dokumentation bewirkt der Wert "primary", dass der API-Server den primären Google-Kalender des aktuell angemeldeten Benutzers aufruft. Beachten Sie dass dieser Wert ein Design-Zeit-Wert ist und nur gilt, wenn Sie in MapForce eine Vorschau auf das Mapping anzeigen. Wenn das Mapping in einer Server-Umgebung ausgeführt wird, muss der gewünschte Wert zur Laufzeit bereitgestellt werden.

2. Ziehen Sie die Funktion **decode-mime-entity** aus dem Fenster "Bibliotheken" in den Mapping-Bereich. Diese Funktion konvertiert den vom Server erhaltenen rohen MIME-Body in einen String.
3. Klicken Sie im Menü **Einfügen** auf **Output-Komponente einfügen** und fügen Sie eine einfache Output-Komponente hinzu, deren Aufgabe es ist, das Ergebnis als einfachen String auszugeben.
4. Verbinden Sie die Komponenten als nächstes wie unten gezeigt.



Damit ist der Design-Teil in MapForce abgeschlossen.

Testen der Mapping-Ausführung

Um die Ausführung des Mappings in MapForce zu testen, klicken Sie auf das Register **Ausgabe**, um das Ergebnis im Fenster "Meldungen" zu sehen.

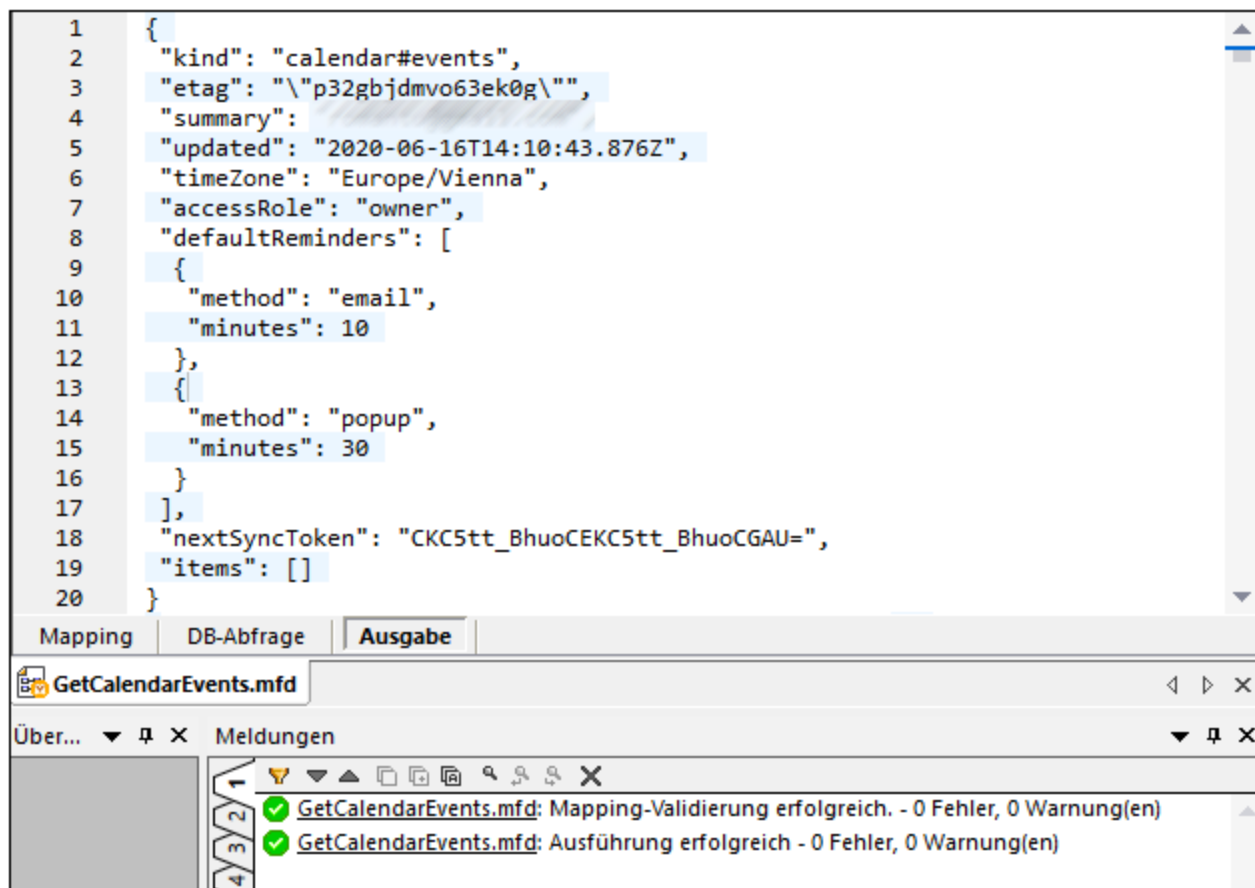
Wenn Sie einen Autorisierungsfehler wie z.B. "Unauthorized (401)" erhalten, beachten Sie die folgenden Tipps zur Fehlerbehebung:

1. Stellen Sie sicher, dass die Google Calendar API aktiviert ist, siehe [Aktivieren der Google Calendar API](#) ⁵¹.
2. Falls das zuvor erhaltene Access Token bereits abgelaufen ist, [fordern Sie ein neues Autorisierungs-](#)

[Token an](#) ⁵².

- Überprüfen Sie, ob alle OAuth 2.0-Informationen korrekt in MapForce eingegeben wurden.

Wenn das Mapping erfolgreich ausgeführt werden konnte und MapForce die OAuth 2.0-Autorisierung abrufen konnte, sollte die Mapping-Ausgabe ähnlich wie unten gezeigt aussehen:



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

Mapping DB-Abfrage **Ausgabe**

GetCalendarEvents.mfd

Über... Meldungen

- ✓ GetCalendarEvents.mfd: Mapping-Validierung erfolgreich. - 0 Fehler, 0 Warnung(en)
- ✓ GetCalendarEvents.mfd: Ausführung erfolgreich - 0 Fehler, 0 Warnung(en)

Wenn Sie ein Google-Konto verwendet haben, das wie in diesem Beispiel keine Kalenderereignisse enthält, ist das Array "items" in der Antwort leer. Wenn Sie jedoch ein Ereignis zu Ihrem Google-Kalender hinzufügen und das Mapping erneut ausführen, wird dies in der Ausgabe entsprechend angezeigt. Sie können übrigens auch Ereignisse aus einem anderen Kalender, als dem Standardkalender abrufen. So könnten Sie z.B. Daten aus einem öffentlichen Kalender wie z.B. "Holidays in United States" abrufen. Setzen Sie zu diesem Zweck den Wert des Parameters **calendarId** auf **en.usa#holiday@group.v.calendar.google.com** anstelle von **primary**.

Informationen zu anderen Parametern, die Sie zum API-Aufruf hinzufügen können, finden Sie in der Dokumentation zur API-Methode unter <https://developers.google.com/calendar/v3/reference/events/list>.

Ausführen des Mappings mit MapForce Server (Standalone)

In diesem Abschnitt widmen wir uns der Ausführung des Demo-OAuth 2.0-Mappings mit einem als Standalone-Produkt installierten MapForce Server, der sich nicht unter FlowForce Server-Verwaltung befindet. Informationen über die Ausführung solcher Mappings mit MapForce Server unter FlowForce Server-Verwaltung finden Sie in der FlowForce Server-Dokumentation, wo dieses Beispiel fortgeführt wird.

Um ein OAuth 2.0-Mapping mit MapForce Server Standalone auszuführen, gibt es zwei Methoden der Behandlung von OAuth 2.0-Anmeldeinformationen:

- Inkludieren Sie das OAuth 2.0-Token (in verschlüsselter Form) in der kompilierten .mfx-Datei. Bei dieser Methode müssen Sie die OAuth 2.0-Anmeldeinformationen nicht in der Befehlszeile (oder im MapForce Server API-Aufruf) angeben, da die eingebetteten Anmeldeinformationen verwendet werden. Dadurch kann jedoch jeder, der Zugriff auf die .mfx-Datei hat, dieses Mapping ausführen, ohne das Autorisierungs-Token angeben zu müssen, bis das Token abläuft oder der Autorisierungsserver es widerruft. Beachten Sie, dass Sie das Autorisierungs-Token jederzeit über die Befehlszeile außer Kraft setzen können, ohne die .mfx-Datei neu kompilieren zu müssen (siehe nächster Punkt).
- Inkludieren Sie das OAuth 2.0-Token nicht in der kompilierten .mfx-Datei. Bei dieser Methode müssen Sie (oder jeder andere Benutzer, der die .mfx-Datei ausführt) das OAuth 2.0-Autorisierungs-Token in der Befehlszeile oder im MapForce Server API-Aufruf angeben. Das Autorisierungs-Token selbst muss von außerhalb von MapForce Server, z.B. über MapForce (siehe Beschreibung oben) angefordert werden.

In diesem Beispiel wird das Autorisierungs-Token nicht in der kompilierten .mfx-Datei inkludiert. Stattdessen wird es zur Laufzeit bereitgestellt.

1. Klicken Sie in MapForce mit der rechten Maustaste in einen leeren Bereich des Mappings und wählen Sie den Befehl **Anmeldeinformationen-Manager öffnen**.
2. Doppelklicken Sie auf den Anmeldedatensatz (in diesem Beispiel "my.oauth") und deaktivieren Sie das Kontrollkästchen **Verschlüsselt in MapForce Server-Ausführungsdatei und Mapping-Bereitstellung inkludieren**.
3. Speichern Sie die Mapping-Design-Datei (.mfd).

Kompilieren Sie das Mapping nun zu einer MapForce Server-Ausführungsdatei (.mfx):

1. Klicken Sie im Menü **Datei** auf den Befehl **Zu MapForce Server-Ausführungsdatei kompilieren**.
2. Wählen Sie ein Zielverzeichnis aus und speichern Sie die Datei als **GetCalendarEvents.mfx**.

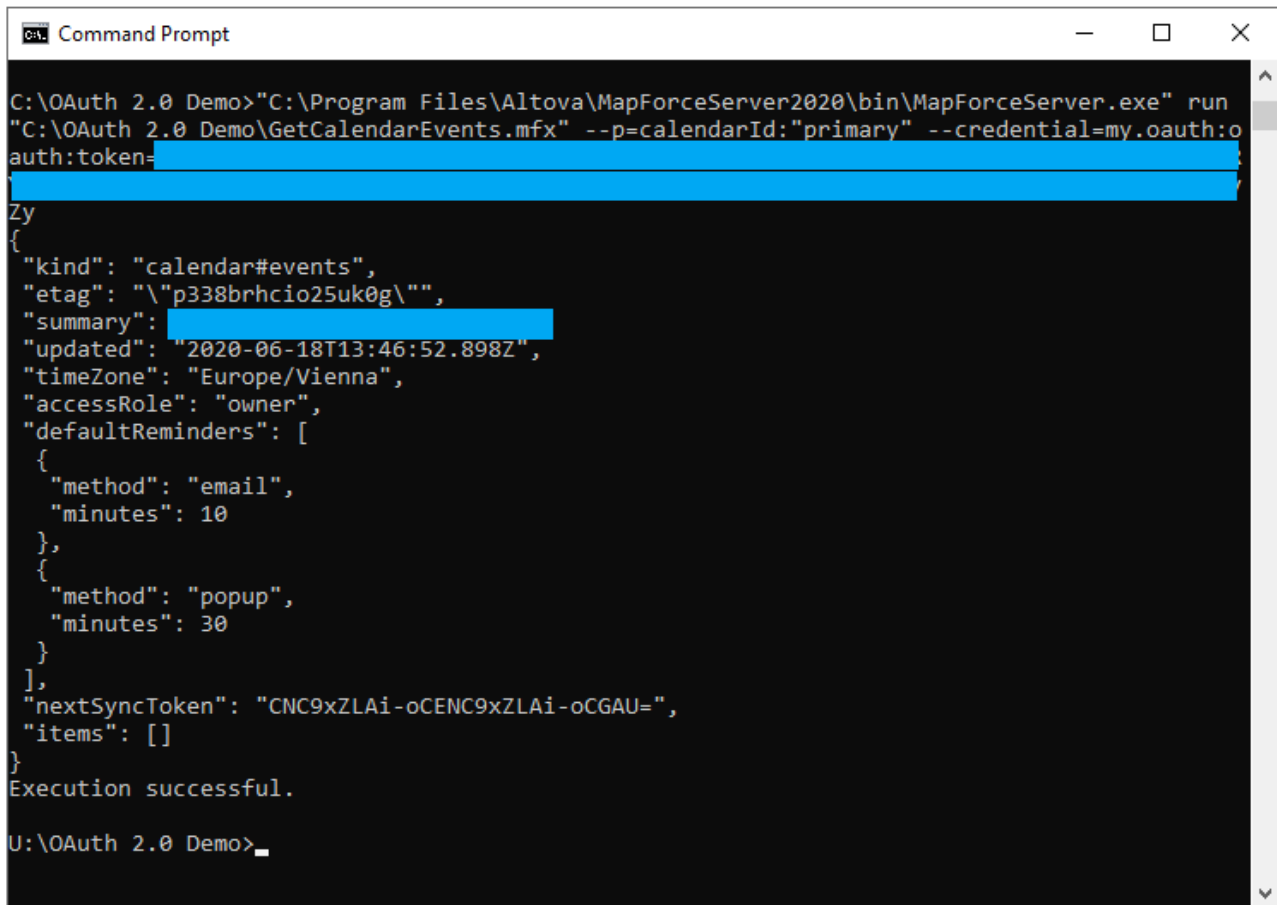
Sie können nun ein Eingabeaufforderungsfenster öffnen und die .mfx-Datei mit einem Befehl wie dem folgenden ausführen:

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

wobei:

- **mapforceserver-exec** der Pfad zur ausführbaren MapForce Server-Datei ist, normalerweise **C:\Programme\Altova\MapForceServer2025\bin\MapForceServer.exe**.
- **GetCalendarEvents.mfx** der Pfad zur .mfx-Datei relativ zu aktuellem Verzeichnis der Befehlszeile ist. Passen Sie die Pfade gegebenenfalls an oder verwenden Sie einen absoluten Pfad.
- **calendarId** der Name des in MapForce erstellten Input-Parameters ist
- **my.oauth** der Name der in MapForce im vorherigen Schritt erstellten Anmeldeinformationen ist.
- **mytoken** der Wert des extern (in diesem Fall mit MapForce) abgerufenen Autorisierungs-Token ist.

Bei erfolgreicher Ausführung und OAuth 2.0-Autorisierung wird in der Befehlszeilenausgabe die von der Google Calendar API retournierte Antwort angezeigt, z.B:



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

Beachten Sie, dass das Autorisierungs-Token sehr schnell abläuft (das Intervall hängt vom Autorisierungsserver, in diesem Fall Google ab). Eventuell müssen Sie ein neues anfordern, wenn Sie "Unauthorized"-Fehler erhalten, siehe [Anforderung eines Autorisierungs-Token](#) ⁵².

3.5 Dynamische Authentifizierung

Sie können in MapForce Mappings definieren, die Webservices für eine einfache HTTP-Authentifizierung aufrufen. Eine dynamische Authentifizierung kann als eine der Alternativen zur Verwendung von Anmeldeinformationen verwendet werden. Dynamische Authentifizierung bedeutet, ein Mapping so zu entwerfen, dass Benutzername und Passwort darin als Input-Parameter akzeptiert werden. Nähere Informationen zum Konfigurieren einer dynamischen Authentifizierung finden Sie in der MapForce-Dokumentation (<https://www.altova.com/de/documentation>).

Wenn Sie das Mapping für eine dynamische Authentifizierung konfiguriert haben, müssen der entsprechende Benutzername und das entsprechende Passwort zur Mapping-Laufzeit genau wie bei der Bereitstellung anderer Parameterarten als Parameter bereitgestellt werden. Wenn Sie z.B. MapForce Server über die Befehlszeile aufrufen, lautet die Syntax für ein Mapping wie das oben beschriebene folgendermaßen:

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

Wobei:

- **<exec>** der Pfad zur ausführbaren MapForce Server-Datei ist. Dabei kann es sich entweder um einen absoluten Pfad handeln oder Sie können einfach nur den Namen der ausführbaren Datei angeben, wenn das aktuelle Verzeichnis dasselbe wie das der ausführbaren Datei ist.
- **username** und **password** die Namen der entsprechenden Input-Parameter im MapForce Mapping sind.

Bei Aufruf der MapForce Server API können Sie das Mapping durch Aufruf der Methode **AddParameter** authentifizieren, bevor Sie die Methode **Run** aufrufen. Der Code dafür in C#, könnte folgendermaßen aussehen:

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

Auch hier müssen **username** und **password** (das erste Argument für die Methode **AddParameter**) mit den Namen der entsprechenden Input-Parameter im MapForce-Mapping identisch sein.

4 MapForce Server-Befehlszeile

MapForce Server bietet eine Befehlszeilenschnittstelle, über die Sie Verwaltungsaufgaben wie die Lizenzierung durchführen und auch Mapping-Ausführungsdateien (.mfx) ausführen können. Es stehen die folgenden Befehle zur Verfügung:

- [accepteula](#)⁷¹: (nur Linux) Akzeptiert die Endbenutzer-Lizenzvereinbarung für MapForce Server.
- [assignlicense](#)⁶⁷: (nur Windows) Lädt eine Lizenz in LicenseServer hoch und weist sie MapForce Server zu.
- [exportresourcestrings](#)⁶⁸: Exportiert alle Applikationsressourcenstrings in eine XML-Datei.
- [help](#)⁷⁰: Zeigt Informationen zu dem im Argument angegebenen Befehl - oder wenn kein Befehl angegeben ist, zu allen Befehlen - an.
- [licenseserver](#)⁷¹: Registriert MapForce Server auf einem LicenseServer im lokalen Netzwerk.
- [run](#)⁷²: Führt eine mit MapForce kompilierte Mapping-Ausführungsdatei (.mfx) aus.
- [setdeflang](#)⁷⁶: Definiert die Standardsprache von MapForce Server.
- [verifylicense](#)⁷⁷: (Nur Windows) Überprüft, ob der aktuelle MapForce Server lizenziert ist; optional dazu können Sie überprüfen, ob ein bestimmter Lizenzschlüssel zugewiesen ist.
- [version](#)⁷⁸: Zeigt die Versionsnummer von MapForce Server an.

Um MapForce Server über die Befehlszeile aufzurufen, müssen Sie den jeweils für Ihr Betriebssystem geltenden Pfad zur ausführbaren Datei kennen. Standardmäßig befindet sich die ausführbare MapForce Server-Datei unter dem folgenden Pfad:

<i>Linux</i>	/opt/Altova/MapForceServer2025/bin/mapforceserver
<i>macOS</i>	/usr/local/Altova/MapForceServer2025/bin/mapforceserver
<i>Windows</i>	C:\Programme\Altova\MapForceServer2025\bin\MapForceServer.exe

Anmerkung: Wenn MapForce Server 32-Bit unter Windows 64-Bit installiert ist, ändern Sie **C:\Programme** in **C:\Programme (x86)**.

Bei der Beschreibung eines Befehls wird in dieser Dokumentation konventionsgemäß anstelle des vollständigen Pfads der ausführbaren Datei **mapforceserver** verwendet, z.B:

```
mapforceserver help
```

Hierbei ist **mapforceserver** der Pfad oder Name der ausführbaren Datei. Beachten Sie Folgendes: Wenn Sie einen absoluten Pfad verwenden, können Sie Befehle unabhängig vom aktuellen Verzeichnis, in dem sich Ihr Eingabeaufforderungsfenster (Terminal) befindet, ausführen. Wenn Sie die ausführbare Datei jedoch nur durch Eingabe ihres Namens aufrufen möchten, dann wählen Sie eine der nachstehenden Methoden:

- Das aktuelle Verzeichnis im Terminal muss das MapForce Server-Installationsverzeichnis sein.
- Fügen Sie das Verzeichnis, in dem sich die ausführbare Datei befindet, zur Umgebungsvariablen PATH hinzu.

Beide Methoden werden weiter unten näher beschrieben.

Tipps und Tricks

Wenn Sie mit der Befehlszeile nicht vertraut sind, hier einige Tipps und Tricks:

- Um das aktuelle Verzeichnis, in dem sich das Befehlszeilenfenster gerade befindet, zu ermitteln, geben Sie auf Linux- und macOS-Systemen **pwd** ein. Geben Sie auf Windows-Systemen **echo %CD%** ein.
- Durch Drücken der **Tabulatortaste** können Sie schnell verschiedene Datei- oder Verzeichnispfade eingeben, ohne diese vollständig eintippen zu müssen. Wenn Sie z.B. **cd c:\prog** in die Befehlszeile eingeben und anschließend die **Tabulatortaste** drücken, wird automatisch **C:\Program Files** (oder eventuell ein anderes Verzeichnis unter C:\, dessen Name mit "Prog" beginnt) ergänzt.
- Setzen Sie bei Eingabe von Pfaden, die Leerzeichen enthalten, wie z.B. unter Windows **C:\Program Files**, diese Leerzeichen in Anführungszeichen.
- Wenn eine Meldung angezeigt wird, dass "dieser Befehl nicht als interner oder externer Befehl, ausführbares Programm oder Batch-Datei erkannt wurde", so haben Sie sich bei der Eingabe eines Pfads oder Befehls wahrscheinlich vertippt.
- Unter Linux muss bei Eingabe von Datei- oder Verzeichnisnamen die Groß- und Kleinschreibung beachtet werden. Wenn Sie z.B. einen Pfad wie **/home/nikita/downloads** eingeben, wird ein Fehler angezeigt, wenn der Verzeichnisname in Wirklichkeit **/home/nikita/Downloads** lautet.
- Verwenden Sie unter Linux oder macOS in Pfadnamen anstelle des umgekehrten Schrägstrichs den Schrägstrich vorwärts.

Ausführen eines Befehls

1. Öffnen Sie ein Eingabeaufforderungsfenster.
 - a. Drücken Sie dazu unter **Windows** die Windows-Taste und beginnen Sie mit der Eingabe von **cmd**. Klicken Sie auf den angezeigten Vorschlag **Command Prompt**.
 - b. Um auf Mac einen Terminal zu öffnen, klicken Sie auf das **Such**-Symbol und wählen Sie im Menü anschließend den Befehl **Go > Utilities**. Doppelklicken Sie im Fenster "Utilities" auf das **Terminal**-Symbol.
 - c. Wenn Sie Linux über eine grafische Benutzeroberfläche ausführen, navigieren Sie zum jeweiligen **Terminal**-Befehl Ihrer Linux-Distribution und führen Sie ihn aus. Ignorieren Sie diesen Schritt, wenn Sie Linux über eine Befehlszeilenschnittstelle ausführen.
2. Geben Sie den vollständigen Pfad zur ausführbaren Datei, gefolgt vom gewünschten Befehl ein. Mit den nachstehenden Befehl können Sie z.B. die Hilfe in der Befehlszeile aufrufen.

<i>Linux</i>	/opt/Altova/MapForceServer2025/bin/mapforceserver help
<i>macOS</i>	/usr/local/Altova/MapForceServer2025/bin/mapforceserver help
<i>Windows</i>	C:\Programme (x86)\Altova\MapForceServer2025\bin\MapForceServer.exe help

Im obigen Beispiel wurde der Befehl **help** ohne Optionen oder Argumente ausgeführt. Andere Befehle haben manchmal Argumente und Optionen, die obligatorisch oder optional sein können. So hat z.B. der Befehl **run** ein obligatorisches Argument, mit dem Sie den Pfad oder Namen der auszuführenden .mfx-Datei angeben können. Nähere Informationen zu den einzelnen Befehlen finden Sie im Abschnitt "Referenz".

Aufruf von MapForce Server im Installationsverzeichnis

Um die ausführbare Datei aufzurufen, ohne den vollständigen Pfad eingeben zu müssen, wechseln Sie in das Verzeichnis, in dem die ausführbare MapForce Server-Datei installiert wurde, z.B:

<i>Linux</i>	<code>cd /opt/Altova/MapForceServer2025/bin</code>
<i>macOS</i>	<code>cd /usr/local/Altova/MapForceServer2025/bin</code>
<i>Windows</i>	<code>cd C:\Programme (x86)\Altova\MapForceServer2025\bin</code>

Sie können jeden beliebigen Befehl nun einfach durch Eingabe des Namens der ausführbaren Datei ausführen, z.B:

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

Anmerkung: Auf Linux- und macOS-Systemen gibt das Präfix `./` an, dass sich die ausführbare Datei im aktuellen Verzeichnis befindet.

Aufruf von MapForce Server von jedem beliebigen Verzeichnis aus

Um die ausführbare Datei von jedem beliebigen Verzeichnis aus aufzurufen, geben Sie den absoluten Pfad ein. Wenn Sie das Programm alternativ dazu nur durch Eingabe des Namens der ausführbaren Datei aufrufen möchten, können Sie die Umgebungsvariable PATH Ihres Betriebssystems bearbeiten, sodass diese den vollständigen Pfad zum MapForce Server-Installationsverzeichnis enthält. Wie Sie die PATH-Umgebungsvariable ändern können, finden Sie in der Dokumentation zu Ihrem Betriebssystem.

Anmerkung: Nachdem Sie die PATH-Umgebungsvariable geändert haben, müssen Sie das Terminal-Fenster unter Umständen schließen und ein neues öffnen, damit die Änderungen wirksam werden.

4.1 accepteula (nur Linux)

Syntax und Beschreibung

Um MapForce Server ausführen zu können, muss die Endbenutzer-Lizenzvereinbarung (EULA) akzeptiert werden. Sie können die EULA der Applikation mit dem Befehl **accepteula** akzeptieren.

Dieser Befehl ist z.B. nützlich, wenn Sie MapForce Server direkt über automatisierte Prozesse unter Verwendung von Skripts lizenzieren und ausführen möchten.

mapforceserver accepteula [options]

- Der Befehl funktioniert nur bei Altova Server-Produkten, die auf Linux-Rechnern installiert wurden.
- Mit Hilfe der Option **--h, --help** können Sie Informationen über den Befehl anzeigen.
- Schreiben Sie den Befehl `mapforceserver` in Kleinbuchstaben.
- Verwenden Sie auf Linux-Systemen Vorwärts-Schrägstriche.

Beispiele

Beispiele für den Befehl **accepteula**:

mapforceserver accepteula

Optionen

Mit Hilfe der Option **--h, --help** können Sie Informationen über den Befehl anzeigen.

4.2 assignlicense (nur Windows)

Syntax und Beschreibung

Mit dem Befehl **assignlicense** wird eine Lizenzdatei auf den Altova LicenseServer, auf dem MapForce Server registriert ist (siehe Befehl **licenseserver**) hochgeladen und die Lizenz wird MapForce Server zugewiesen. Der Befehl erhält den Pfad einer Lizenzdatei als Argument. Außerdem können Sie mit dem Befehl die Gültigkeit einer Lizenz überprüfen.

mapforceserver assignlicense [options] FILE

- Das Argument *FILE* erhält den Pfad der Lizenzdatei.
- Mit der Option **--test-only** wird die Lizenzdatei auf LicenseServer hochgeladen und auf ihre Gültigkeit überprüft. Sie wird jedoch MapForce Server nicht zugewiesen.

Nähere Informationen zur Lizenzierung finden Sie in der Dokumentation zu LicenseServer (<https://www.altova.com/manual/de/licenseserver/3.17/>).

Beispiele

Beispiele für den Befehl **assignlicense**:

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

- Mit dem ersten Befehl oben wird die angegebene Lizenzdatei auf LicenseServer hochgeladen und MapForce Server zugewiesen.
- Mit dem letzten Befehl wird die angegebene Lizenz auf LicenseServer hochgeladen und auf ihre Gültigkeit überprüft, ohne sie MapForce Server zuzuweisen.

Optionen

Optionen werden in ihrer kurzen Form (falls verfügbar) und in ihrer langen Form aufgelistet. Für die kurze und die lange Form können ein oder zwei Bindestriche verwendet werden. Eine Option kann, muss aber keinen Wert erhalten. Eine Option, die einen Wert erhält, wird folgendermaßen geschrieben: **--option=wert**. Werte können außer in zwei Fällen ohne Anführungszeichen definiert werden: (i) wenn der Wertestring Leerzeichen enthält oder (ii) wenn in der Beschreibung der Option explizit erwähnt ist, dass Anführungszeichen zwingend erforderlich sind. Wenn eine Option einen Booleschen Wert erhält und kein Wert definiert ist, so ist der Standardwert der Option **TRUE**. Mit Hilfe der Option **--h, --help** können Sie Informationen über den Befehl anzeigen.

▼ test-only [t]

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

Die Werte sind **true|false**. Bei **true** wird die Lizenzdatei auf LicenseServer hochgeladen und auf ihre Gültigkeit überprüft, aber nicht zugewiesen.

4.3 exportresourcestrings

Syntax und Beschreibung

Der Befehl **exportresourcestrings** gibt eine XML-Datei aus, die die Ressourcenstrings der MapForce Server-Applikation in der definierten Sprache enthält. Als Exportsprachen stehen Englisch (en), Deutsch (de), Spanisch (es), Französisch (fr) und Japanisch (ja) zur Verfügung.

mapforceserver exportresourcestrings [options] LanguageCode XMLOutputFile

- Das Argument **LanguageCode** gibt die Sprache der Ressourcenstrings in der XML-Ausgabedatei an; dies ist die *Exportsprache*. Derzeit unterstützte Exportsprachen sind (mit den Sprachcodes in Klammern): Englisch (en), Deutsch, (de), Spanisch (es), Französisch (fr) und Japanisch (ja).
- Das Argument **XMLOutputFile** definiert den Namen und Pfad der XML-Ausgabedatei.

Eine Anleitung zum Erstellen von Lokalisierungen finden Sie weiter unten.

Beispiele

Beispiele für den Befehl **exportresourcestrings**:

mapforceserver exportresourcestrings en c:\Strings.xml

- Mit dem Befehl oben wird unter c:\ eine Datei namens Strings.xml erstellt, die alle Ressourcenstrings der MapForce Server Applikation in englischer Sprache enthält.

Erstellen lokalisierter Versionen von MapForce Server

Sie können für jede Sprache Ihrer Wahl eine lokalisierte Version von MapForce Server erstellen. Im Ordner C:\Programme (x86)\Altova\MapForceServer2025\bin stehen fünf lokalisierte Versionen (Englisch, Deutsch, Spanisch, Französisch und Japanisch) bereits zur Verfügung. Eine Lokalisierung für diese Sprache ist daher nicht mehr notwendig.

Folgendermaßen können Sie eine lokalisierte Version erstellen:

1. Generieren Sie mit Hilfe des Befehls **exportresourcestrings** (*siehe Befehlssyntax oben*) eine XML-Datei, die die Ressourcenstrings enthält. Die Ressourcenstrings in dieser XML-Datei sind in einer der fünf unterstützten Sprachen: je nachdem, welches **LanguageCode**-Argument mit dem Befehl verwendet wird, in Englisch (en), Deutsch (de), Spanisch (es), Französisch (fr) oder Japanisch (ja).
2. Übersetzen Sie die Ressourcenstrings aus einer der fünf unterstützten Sprachen in die Zielsprache. Die Ressourcenstrings bilden den Inhalt der <string> Elemente in der XML-Datei. Übersetzen Sie keine Variablen in geschweiften Klammern wie z.B. {option} oder {product}.
3. Wenden Sie sich an den [Altova Support](#), um anhand Ihrer übersetzten XML-Datei eine lokalisierte MapForce Server DLL-Datei zu generieren.
4. Nachdem Sie Ihre lokalisierte DLL-Datei vom [Altova Support](#) erhalten haben, speichern Sie diese unter C:\Programme (x86)\Altova\MapForceServer2025\bin. Ihre DLL-Datei wird einen Namen in der Form MapForceServer2025_lc.dll haben. Der lc Teil des Namens enthält den Sprachencode. So steht z.B. in MapForceServer2025_de.dll der Teil de für Deutsch.
5. Führen Sie den Befehl **setdeflang** aus, um Ihre lokalisierte DLL als die zu verwendende MapForce Server Applikation zu definieren. Verwenden Sie den Sprachencode, der Teil des DLL-Namens ist, als Argument des Befehls **setdeflang**.

Anmerkung: Altova MapForce Server ist mit Unterstützung für fünf Sprachen erhältlich: Englisch, Deutsch, Spanisch, Französisch und Japanisch. Sie müssen daher keine lokalisierte Version dieser Sprachen erstellen. Um eine dieser Sprachen als Standardsprache festzulegen, verwenden Sie den MapForce Server Befehl `setdeflang`.

4.4 help

Syntax und Beschreibung

Der Befehl **help** hat ein einziges Argument (Command): den Namen des Befehls, zu dem die Hilfe benötigt wird. Er zeigt die korrekte Syntax des Befehls, seine Optionen sowie andere relevante Informationen an. Wenn das Argument Command nicht angegeben wird, werden alle Befehle der ausführbaren Datei aufgelistet, wobei zu jedem eine kurze Textbeschreibung angezeigt wird.

mapforceserver help Command

Beispiel

Beispiel für den Befehl **help**, um Informationen über den Befehl **licenseserver** anzuzeigen:

mapforceserver help licenseserver

Die Option --help

Die Hilfe zu einem Befehl kann auch über die Option **--help** im Anschluss an diesen Befehl aufgerufen werden. Mit den beiden unten stehenden Befehlen erhalten Sie dasselbe Ergebnis:

mapforceserver licenseserver --help

Im obigen Befehl wird die Option **--help** des Befehls **licenseserver** verwendet.

mapforceserver help licenseserver

Der Befehl **help** erhält **licenseserver** als Argument.

In beiden Fällen wird die Hilfe zum Befehl **licenseserver** angezeigt.

4.5 licenseserver

Syntax und Beschreibung

Bei Ausführung des Befehls **licenseserver** wird MapForce Server auf dem durch das Argument *Server-Or-IP-Address* definierten LicenseServer registriert. Damit der Befehl **licenseserver** erfolgreich ausgeführt werden kann, müssen sich die beiden Server (MapForce Server und LicenseServer) im selben Netzwerk befinden und LicenseServer muss ausgeführt werden. Außerdem benötigen Sie zum Registrieren von MapForce Server auf dem LicenseServer Administratorrechte.

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

- Das Argument *Server-Or-IP-Address* erhält den Namen oder die IP-Adresse des LicenseServer-Rechners.

Sobald MapForce Server erfolgreich auf dem LicenseServer registriert wurde, erhalten Sie eine entsprechende Meldung. Darin wird auch die URL des LicenseServer angezeigt. Sie können nun zu LicenseServer wechseln und MapForce Server eine Lizenz zuweisen. Nähere Informationen zur Lizenzierung finden Sie in der Dokumentation zu LicenseServer (<https://www.altova.com/manual/de/licenseserver/3.17/>).

Beispiele

Beispiele für den Befehl **licenseserver**:

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

Die Befehle oben definieren den Rechner namens `DOC.altova.com` und den Rechner des Benutzers (`localhost` bzw. `127.0.0.1`) als den Rechner, auf dem Altova LicenseServer ausgeführt wird. In jedem dieser Fälle wird MapForce Server auf dem LicenseServer auf dem angegebenen Rechner registriert. Mit dem letzten Befehl wird zum Ausführen des Befehls die ausführbare Server-Datei aufgerufen.

Optionen

Optionen werden in ihrer kurzen Form (falls verfügbar) und in ihrer langen Form aufgelistet. Für die kurze und die lange Form können ein oder zwei Bindestriche verwendet werden. Eine Option kann, muss aber keinen Wert erhalten. Eine Option, die einen Wert erhält, wird folgendermaßen geschrieben: **--option=wert**. Werte können außer in zwei Fällen ohne Anführungszeichen definiert werden: (i) wenn der Wertestring Leerzeichen enthält oder (ii) wenn in der Beschreibung der Option explizit erwähnt ist, dass Anführungszeichen zwingend erforderlich sind. Wenn eine Option einen Booleschen Wert erhält und kein Wert definiert ist, so ist der Standardwert der Option `TRUE`. Mit Hilfe der Option **--h, --help** können Sie Informationen über den Befehl anzeigen.

▼ json [j]

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

Die Werte sind `true|false`. Bei `true` wird das Ergebnis des Registrierungsversuchs als JSON-Objekt ausgegeben, das von Rechnern geparkt werden kann.

4.6 run

Der Befehl `run` führt eine als Argument bereitgestellte MapForce Server-Ausführungsdatei (.mfx-Datei) aus. Die MapForce Server-Ausführungsdatei wird mit MapForce erstellt. Es handelt sich dabei im Grunde um ein für die Server-Ausführung kompiliertes Mapping.

Für das Mapping erforderliche Input-Dateien müssen sich unter dem Pfad befinden, der zum Zeitpunkt des Mapping-Designs in MapForce definiert wurde. Wenn MapForce Server nicht auf demselben Betriebssystem wie MapForce ausgeführt wird, müssen die für das Mapping erforderlichen Input-Dateien zusammen mit der .mfx.-Datei auf den Zielrechner kopiert und über einen relativen Pfad referenziert werden. Informationen zum Konfigurieren eines Mappings im Hinblick auf relative oder absolute Pfade finden Sie in der MapForce-Dokumentation (<https://www.altova.com/de/documentation#mapforce>). Je nach Mapping-Design müssen auch andere Voraussetzungen erfüllt werden, siehe [Vorbereiten von Mappings für die Server-Ausführung](#)³⁴.

Wenn das Ergebnis des Mappings ein einfacher Wert wie z.B. ein String ist, wird diese Ausgabe in den `stdout` (Standard Output) Stream geschrieben. Die Erfolgs- und Fehlermeldungen wiederum stehen im `stderr` (Standard Error) Stream zur Verfügung. Wenn Sie nicht möchten, dass der Standardausgabe-Stream zusammen mit den Erfolgs- oder Fehlermeldungen auf dem Bildschirm angezeigt wird, leiten Sie entweder den Standardausgabe- oder den Standard Error Stream (oder beide) in Dateien um. Wenn weder der `stdout` noch der `stderr` Stream umgeleitet wird, werden beide kombiniert auf dem Bildschirm angezeigt.

Um z.B. den Standardausgabe-Stream in eine Datei umzuleiten, verwenden Sie den folgenden Befehl:

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

Um den Standard Error-Stream in eine Datei umzuleiten, verwenden Sie den Befehl:

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

Um beide Streams gleichzeitig umzuleiten, verwenden Sie den Befehl:

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

Nähere Informationen zum Umleiten von Streams finden Sie in der Dokumentation zur Befehls-Shell Ihres Betriebssystems.

Syntax

Windows	<code>MapForceServer run [options] MfxFile</code>
Linux	<code>mapforceserver run [options] MfxFile</code>
Mac	<code>mapforceserver run [options] MfxFile</code>

Befehloptionen

	<code>--catalog</code>	Definiert den absoluten Pfad zu einer Root-Katalog-Datei, die nicht die installierte Root-Katalog-Datei ist. Der
--	------------------------	--

		Standardwert ist der absolute Pfad zur installierten Root-Katalog-Datei, siehe Katalogdateien ¹⁸⁰ . Form: --catalog=FILE
--cert	-certificatespath	Dieser Befehl gilt für MapForce Server, wenn er auf Linux ausgeführt wird. Damit wird der Pfad zu dem Verzeichnis definiert, in dem für das Mapping benötigte Zertifikatdateien gespeichert sind. Form: --certificatespath=DIRECTORY Siehe auch Verwaltung digitaler Zertifikate ¹⁴² .
--cred	--credential=KEY:VALUE	Diese Option ist anwendbar, wenn das Mapping Anmeldeinformationen enthält, die Sie zum Zeitpunkt des Designs des Mappings in MapForce definiert haben. KEY ist der Namen der Anmeldeinformationen, wie er in MapForce definiert wurde. VALUE besteht aus einem oder mehreren Name-Wert-Paaren, z.B: name1=value1&name2=value2 Hierbei ist name einer der folgenden: • username • Passwort • oauth:token Wenn Sie mehrere durch das Kaufmännische Und-Zeichen getrennte Name-Wert-Paare verwenden, setzen Sie das KEY:VALUE-Paar in Anführungszeichen, siehe Anmeldeinformationen ⁴⁵ .
--gc	--globalresourceconfig	Diese Option ist anwendbar, wenn im Mapping globale Ressourcen ⁴⁰ verwendet werden. Sie definiert den Namen der Konfiguration für globale Ressourcen. Diese Option muss zusammen mit der Option --globalresourcefile verwendet werden. Form: --gc=VALUE
--gr	--globalresourcefile	Diese Option ist anwendbar, wenn im Mapping globale Ressourcen ⁴⁰ verwendet werden. Sie definiert den Pfad zur Definitionsdatei für globale Ressourcen. Diese Option muss zusammen mit der Option --globalresourceconfig verwendet werden. Form: --gr=FILE.
--l	--lang	Die Sprache, in der Meldungen angezeigt werden. Form: --lang=VALUE (en, de, ja, es)
--p	--param	Diese Option ist anwendbar, wenn für das Mapping Input-Parameter verwendet werden. Damit wird einem im Mapping definierten Parameter ein Wert zugewiesen. Form: --param=ParamName:ParamValue. Vor jedem Parameter

		muss die Option <code>--param</code> verwendet werden. Verwenden Sie Anführungszeichen, wenn <i>ParamName</i> oder <i>ParamValue</i> ein Leerzeichen enthält. Zum Beispiel: <code>--p=company:"Nanonull Inc"</code>. Nähere Informationen zu Mappings, die Input-Parameter erhalten, finden Sie in der MapForce-Dokumentation (https://www.altova.com/de/documentation#mapforce).
	<code>--taxonomy-package</code>	Definiert den absoluten Pfad zu einem zusätzlichen in der Taxonomy Packages 1.0 Recommendation beschriebenen XBRL-Taxonomiepaket. Der Wert FILE gibt den Pfad des Taxonomiepakets an. Fügen Sie die Option mehrmals hinzu, um mehr als ein Taxonomiepaket zu definieren. Form: <code>--taxonomy-package=FILE</code>
	<code>--taxonomy-packages-config-file</code>	Definiert den Pfad zu einer Konfigurationsdatei namens TaxonomyPackagesConfig.json, anhand welcher XBRL-Taxonomiepakete geladen werden. Diese Konfigurationsdatei wird jedes Mal, wenn Sie XBRL-Taxonomiepakete über die grafische Benutzeroberfläche von XMLSpy, MapForce oder StyleVision hinzufügen, entfernen, aktivieren oder deaktivieren, aktualisiert. Wenn Sie über eines der oben erwähnten Produkte benutzerdefinierte XBRL-Taxonomiepakete hinzugefügt haben, so befindet sich die Datei unter C: \Benutzer\<Benutzername>\Documents\Altova. Form: <code>--taxonomy-packages-config-file=FILE</code>

Beispiele

In diesem Beispiel wird gezeigt, wie Sie eine Mapping-Ausführungsdatei (.mfx) mit MapForce Server unter Windows ausführen. In dem in diesem Beispiel beschriebenen Mapping wird die Input-Datei **Employees.xml** ausgelesen und es werden zwei Ausgabedateien erzeugt (**PersonList.xml** und **Contacts.xml**).

Zuerst wird die MapForce Server-Ausführungsdatei (.mfx) folgendermaßen generiert:

1. Starten Sie Altova MapForce und öffnen Sie die folgende MapForce Design-Datei (.mfd-Datei): **C:\Benutzer\<Benutzer>\Dokumente\Altova\MapForce2025\MapForceExamples\ChainedPersonList.mfd**.
2. Klicken Sie im Menü **Datei** auf den Befehl **Zu MapForce Server-Ausführungsdatei kompilieren**.
3. Wenn Sie dazu aufgefordert werden, speichern Sie die .mfx-Datei im Verzeichnis **C:\temp**. Dieses Verzeichnis wird bei der Ausführung durch MapForce Server als Arbeitsverzeichnis verwendet.

Öffnen Sie als nächstes eine Eingabeaufforderung und ändern Sie das Arbeitsverzeichnis in **C:\temp**.

```
cd C:\temp
```

Führen Sie schließlich den folgenden Befehl aus, um **ChainedPersonList.mfx** auszuführen. In diesem Beispiel wird MapForce Server über einen absoluten Pfad aufgerufen. (Um das Programm über einen relativen

Pfad aufzurufen, fügen Sie den Pfad zur ausführbaren Datei zu PATH-Umgebungsvariablen Ihres Systems hinzu).

```
"C:\Programme (x86)\Altova\MapForceServer2025\bin\MapForceServer.exe" run  
ChainedPersonList.mfx
```

Die beiden Ausgabedateien (**PersonList.xml** und **Contacts.xml**) werden im Arbeitsverzeichnis generiert. Beachten Sie, dass in diesem Mapping die Verwendung von absoluten Pfaden konfiguriert wurde. Aus diesem Grund konnte das Mapping erfolgreich ausgeführt werden und die Input-Datei **Employees.xml** musste sich nicht im Arbeitsverzeichnis befinden. Die Datei **Employees.xml** befindet sich im oben erwähnten Ordner MapForce Examples und wird über einen absoluten Pfad referenziert. Um anzugeben, ob Pfade als absolute oder relative Pfad behandelt werden sollen, klicken Sie in MapForce mit der rechten Maustaste in das Mapping, wählen Sie **Mapping-Einstellungen** und aktivieren bzw. deaktivieren Sie anschließend das Kontrollkästchen **Pfade im generierten Code absolut machen**. Immer, wenn Sie die Mapping-Einstellungen ändern, müssen Sie das Mapping erneut zu einer .mfx-Datei kompilieren. Nähere Informationen dazu finden Sie unter [Vorbereiten von Mappings für die Server-Ausführung](#)³⁴.

4.7 setdeflang

Syntax und Beschreibung

Der Befehl **setdeflang** (Kurzform ist **sd1**) definiert die Standardsprache von MapForce Server. Verfügbare Sprachen sind Englisch (en), Deutsch (de), Spanisch (es), Französisch (fr) und Japanisch (ja). Der Befehl erhält ein obligatorisches Argument *LanguageCode*.

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

- Das Argument *LanguageCode* definiert die Standardsprache von MapForce Server. Die entsprechenden Werte sind: en, de, es, fr, ja.
- Mit Hilfe der Option **--h, --help** können Sie Informationen über den Befehl anzeigen.

Beispiele

Beispiele für den Befehl **setdeflang** (**sd1**):

```
mapforceserver sd1 en
mapforceserver setdeflang es
```

- Mit dem ersten Befehl wird als Standardsprache von MapForce Server Englisch definiert.
- Mit dem zweiten Befehl wird als Standardsprache von MapForce Server Spanisch definiert.

Optionen

Mit Hilfe der Option **--h, --help** können Sie Informationen über den Befehl anzeigen.

4.8 verifylicense

Syntax und Beschreibung

Mit dem Befehl **verifylicense** wird überprüft, ob das aktuelle Produkt lizenziert ist. Zusätzlich können Sie mit der Option `--license-key` überprüfen, ob dem Produkt bereits ein bestimmter Lizenzschlüssel zugewiesen wurde.

mapforceserver verifylicense [options]

- Um zu überprüfen, ob MapForce Server eine bestimmte Lizenz zugewiesen ist, geben Sie den Lizenzschlüssel als Wert der Option `--license-key` an.

Nähere Informationen zur Lizenzierung finden Sie in der Dokumentation zu LicenseServer (<https://www.altova.com/manual/de/licenseserver/3.17/>).

Beispiele

Beispiele für den Befehl **verifylicense**:

```
mapforceserver verifylicense
mapforceserver verifylicense --license-key=ABCD123-ABCD123-ABCD123-ABCD123-ABCD123-ABCD123
```

- Mit dem ersten Befehl wird überprüft, ob MapForce Server lizenziert ist.
- Mit dem zweiten Befehl wird überprüft, ob MapForce Server mit dem in der Option `--license-key` definierten Lizenzschlüssel lizenziert ist.

Optionen

Optionen werden in ihrer kurzen Form (falls verfügbar) und in ihrer langen Form aufgelistet. Für die kurze und die lange Form können ein oder zwei Bindestriche verwendet werden. Eine Option kann, muss aber keinen Wert erhalten. Eine Option, die einen Wert erhält, wird folgendermaßen geschrieben: `--option=wert`. Werte können außer in zwei Fällen ohne Anführungszeichen definiert werden: (i) wenn der Wertestring Leerzeichen enthält oder (ii) wenn in der Beschreibung der Option explizit erwähnt ist, dass Anführungszeichen zwingend erforderlich sind. Wenn eine Option einen Booleschen Wert erhält und kein Wert definiert ist, so ist der Standardwert der Option TRUE. Mit Hilfe der Option `--h`, `--help` können Sie Informationen über den Befehl anzeigen.

▼ license-key [l]

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

Überprüft, ob MapForce Server mit dem als Wert dieser Option definierten Lizenzschlüssel lizenziert ist.

4.9 version

Syntax und Beschreibung

Mit dem Befehl **version** wird die Versionsnummer von MapForce Server angezeigt.

```
mapforceserver version
```

Beispiel

Beispiel für den Befehl **version**:

```
mapforceserver version
```

5 MapForce Server API

MapForce Server bietet eine API (Application Programming Interface), die Sie programmatisch über Ihren .NET-, Java- oder COM-basierten Code aufrufen können.

In den folgenden Kapiteln finden Sie eine Einführung in die einzelnen Plattformen:

- [.NET-Schnittstelle](#) ⁸⁰
- [COM-Schnittstelle](#) ⁸⁸
- [Java-Schnittstelle](#) ⁹⁷

In den folgenden Kapiteln finden Sie eine technische Beschreibung der API:

- [API-Referenz \(COM, .NET\)](#) ¹¹⁰
- [API-Referenz \(Java\)](#) ¹²⁸

5.1 .NET-Schnittstelle

Die .NET-Schnittstelle ist als Wrapper rund um die COM-Schnittstelle gebaut. Sie dient als primäre von Altova signierte Interop-Assembly und verwendet den Namespace `Altova.MapForceServer`. MapForce Server wird bei der Installation automatisch als COM-Serverobjekt registriert und muss daher nicht manuell registriert werden.

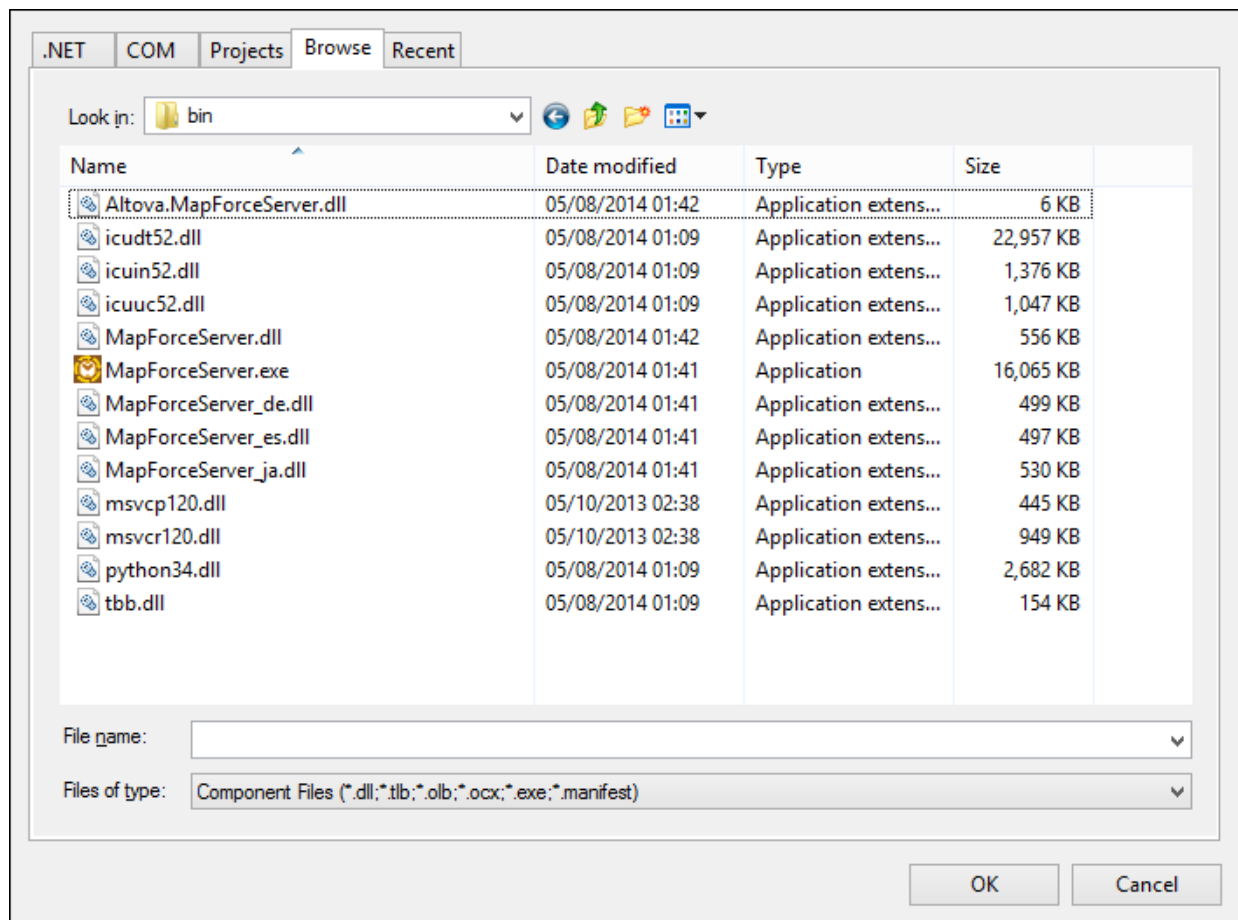
Um MapForce Server in Ihrem .NET-Projekt verwenden zu können, fügen Sie, wie unten gezeigt, eine Referenz zur Datei `Altova.MapForceServer.dll` hinzu. Die Datei `Altova.MapForceServer.dll` befindet sich im Ordner `bin` des MapForce Server-Installationsordners. Diese .dll-Datei wird bei der Installation von MapForce Server automatisch zum Global Assembly Cache (GAC) hinzugefügt.

Anmerkung: Wenn Sie eine 64-Bit-Version von MapForce Server installiert haben, so befinden sich die Dateien der 32-Bit-Version von `Altova.MapForceServer.dll` im Ordner `bin\API_32bit`. Ebenso, wenn Sie eine 32-Bit-Version von MapForce Server installiert haben, so befinden sich die Dateien der 64-Bit-Version von `Altova.MapForceServer.dll` im Ordner `bin\API_64bit`.

Anmerkung: Vor der .NET Framework 4.0-Version befand sich der GAC im Verzeichnis `%windir%\assembly`. Ab der .NET Framework 4.0-Version befindet sich der GAC im Verzeichnis `%windir%\Microsoft.NET\assembly`. Der Teil `%windir%` steht für das Windows-Betriebssystemverzeichnis, normalerweise `C:\Windows`.

So fügen Sie in einem Visual Studio .NET-Projekt eine Referenz zur MapForce Server DLL hinzu

1. Klicken Sie bei geöffnetem .NET-Projekt auf **Project | Add Reference**.



2. Navigieren Sie auf dem Register "Browse" zum Ordner: *<MapForceServer application folder>/bin*, wählen Sie die Datei *Altova.MapForceServer.dll* aus und klicken Sie auf **OK**.

Mit dem Visual Studio Object Browser sehen Sie die Struktur der *Altova.MapForceServer* Assembly (klicken Sie dazu im Menü **View** auf den Befehl **Object Browser**).

5.1.1 C#-Beispiel

Im folgenden Beispiel wird gezeigt, wie Sie eine Mapping-Ausführungsdatei (.mfx) über C#-Code ausführen. Unter Windows stehen die Beispieldateien unter dem folgenden Pfad zur Verfügung: **C:\Programme\Altova\MapForceServer2025\etc\Examples**.

Voraussetzungen

- MapForce Server ist installiert und lizenziert.
- Wenn Sie ein neues Visual Studio-Projekt erstellen, fügen Sie eine Referenz zur MapForce Server Assembly hinzu (siehe [Informationen zur .NET-Schnittstelle](#)⁸⁰). Sie können diesen Schritt überspringen, wenn Sie das vorhandene MapForce Server API-Beispiel ausführen, da die MapForce Server Assembly darin bereits referenziert wird.
- Klicken Sie im Menü **Build** von Visual Studio auf **Configuration Manager** und definieren Sie eine korrekte Build-Plattform, z.B. **Debug | x86** (oder gegebenenfalls **Debug | x64**). Verwenden Sie nicht "Any CPU" als Plattform.

- Wenn Sie eine MapForce Server 64-Bit-Version installiert haben, so muss auch die Applikation, die die API aufruft (wie z.B. die unten gezeigte Beispielapplikation) für die 64-Bit-Plattform in Visual Studio erstellt sein. Außerdem muss der Pfad zur ausführbaren MapForce Server-Datei im Code entsprechend angepasst werden.

Die Beispiellösung befindet sich im Verzeichnis "Programme", für das Sie Administratorrechte benötigen. Starten Sie Visual Studio entweder als Administrator oder kopieren Sie die Lösung in einen anderen Ordner, für den Sie keine Administratorrechte benötigen.

Ausführen des Mapping-Codes

Mit dem nachstehenden Code werden drei Server-Ausführungsdateien (.mfx) ausgeführt. Die Tabelle unten enthält eine Liste der von den einzelnen .mfx.-Dateien erwarteten Input-Dateien sowie der nach der Ausführung erstellten Ausgabedateien.

Ausführungsdatei (.mfx)	Input	Augabe
TokenizeString.mfx	AltovaTools.xml	AltovaToolsFeatures.csv
SimpleTotal.mfx	ipo.xml	<i>String</i>
ClassifyTemperatures.mfx	Temperatures.xml	Temperatures_out.xml

Wenn Sie Altova MapForce installiert haben, haben Sie die Möglichkeit, sich zum besseren Verständnis die Original-Mappings, anhand derer die .mfx-Dateien kompiliert wurden, anzusehen. Diese haben die Namen **TokenizeString1.mfd**, **SimpleTotal.mfd** bzw. **ClassifyTemperatures.mfd**. Sie finden die Mapping im folgenden Verzeichnis: **C:\Benutzer\<Benutzer>\Altova\MapForce2025\MapForceExamples**.

In diesem Beispiel geschieht Folgendes:

- Es wird eine neue Instanz von `Altova.MapForceServer.Server` erstellt. Dies ist das Objekt, mit dem Sie später arbeiten werden.
- Darin wird ein Arbeitsverzeichnis konfiguriert, in dem die Dateien ausgeführt werden. Die Input-Dateien sollten in diesem Verzeichnis vorhanden sein, falls Sie diese über einen relativen Pfad referenziert haben. Auch die Ausgabedateien werden in diesem Verzeichnis erstellt (siehe Tabelle oben).
- **TokenizeString.mfx** wird ausgeführt. Der Dateipfad wird als Argument für die Methode `Run` bereitgestellt (Beachten Sie, dass der Pfad relativ zum zuvor definierten Arbeitsverzeichnis ist). Nach erfolgreicher Ausführung wird im Arbeitsverzeichnis eine .csv-Datei, die die Mapping-Ausgabe darstellt, erstellt.
- **SimpleTotal.mfx** wird ausgeführt. Auch hier ist der Dateipfad wieder relativ zum Arbeitsverzeichnis. Mit diesem Mapping wird eine String-Ausgabe erzeugt, daher wird die Methode `GetOutputParameter` aufgerufen, um die String-Ausgabe zu erhalten.
- **ClassifyTemperatures.mfx** wird ausgeführt. In diesem Mapping wird ein Parameter als Input erwartet, der mit Hilfe der Methode `AddParameter` bereitgestellt wurde.

```
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.
        // The ServerPath should be set before any other properties because it
        re-initializes the server and resets all properties.
        // 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-Beispiel

Im folgenden Beispiel wird gezeigt, wie Sie eine Mapping-Ausführungsdatei (.mfx) über VB .NET-Code ausführen. Unter Windows stehen die Beispieldateien unter dem folgenden Pfad zur Verfügung: **C:\Programme\Altova\MapForceServer2025\etc\Examples**.

Voraussetzungen

- MapForce Server ist installiert und lizenziert.
- Wenn Sie ein neues Visual Studio-Projekt erstellen, fügen Sie eine Referenz zur MapForce Server Assembly hinzu (siehe [Informationen zur .NET-Schnittstelle](#)⁸⁰). Sie können diesen Schritt überspringen, wenn Sie das vorhandene MapForce Server API-Beispiel ausführen, da die MapForce Server Assembly darin bereits referenziert wird.
- Klicken Sie im Menü **Build** von Visual Studio auf **Configuration Manager** und definieren Sie eine korrekte Build-Plattform, z.B. **Debug | x86** (oder gegebenenfalls **Debug | x64**). Verwenden Sie nicht "Any CPU" als Plattform.
- Wenn Sie eine MapForce Server 64-Bit-Version installiert haben, so muss auch die Applikation, die die API aufruft (wie z.B. die unten gezeigte Beispielapplikation) für die 64-Bit-Plattform in Visual Studio erstellt sein. Außerdem muss der Pfad zur ausführbaren MapForce Server-Datei im Code entsprechend angepasst werden.

Die Beispiellösung befindet sich im Verzeichnis "Programme", für das Sie Administratorrechte benötigen. Starten Sie Visual Studio entweder als Administrator oder kopieren Sie die Lösung in einen anderen Ordner, für den Sie keine Administratorrechte benötigen.

Ausführen des Mapping-Codes

Mit dem nachstehenden Code werden drei Server-Ausführungsdateien (.mfx) ausgeführt. Die Tabelle unten enthält eine Liste der von den einzelnen .mfx-Dateien erwarteten Input-Dateien sowie der nach der Ausführung erstellten Ausgabedateien.

Ausführungsdatei (.mfx)	Input	Augabe
TokenizeString.mfx	AltovaTools.xml	AltovaToolsFeatures.csv
SimpleTotal.mfx	ipo.xml	<i>String</i>
ClassifyTemperatures.mfx	Temperatures.xml	Temperatures_out.xml

Wenn Sie Altova MapForce installiert haben, haben Sie die Möglichkeit, sich zum besseren Verständnis die Original-Mappings, anhand derer die .mfx-Dateien kompiliert wurden, anzusehen. Diese haben die Namen **TokenizeString1.mfd**, **SimpleTotal.mfd** bzw. **ClassifyTemperatures.mfd**. Sie finden die Mapping im folgenden Verzeichnis: **C:\Benutzer\<Benutzer>\Altova\MapForce2025\MapForceExamples**.

In diesem Beispiel geschieht Folgendes:

- Es wird eine neue Instanz von `Altova.MapForceServer.Server` erstellt. Dies ist das Objekt, mit dem Sie später arbeiten werden.
- Darin wird ein Arbeitsverzeichnis konfiguriert, in dem die Dateien ausgeführt werden. Die Input-Dateien sollten in diesem Verzeichnis vorhanden sein, falls Sie diese über einen relativen Pfad referenziert haben. Auch die Ausgabedateien werden in diesem Verzeichnis erstellt (siehe Tabelle oben).
- **TokenizeString.mfx** wird ausgeführt. Der Dateipfad wird als Argument für die Methode `Run` bereitgestellt (Beachten Sie, dass der Pfad relativ zum zuvor definierten Arbeitsverzeichnis ist). Nach erfolgreicher Ausführung wird im Arbeitsverzeichnis eine .csv-Datei, die die Mapping-Ausgabe darstellt, erstellt.
- **SimpleTotal.mfx** wird ausgeführt. Auch hier ist der Dateipfad wieder relativ zum Arbeitsverzeichnis. Mit diesem Mapping wird eine String-Ausgabe erzeugt, daher wird die Methode **`GetOutputParameter`** aufgerufen, um die String-Ausgabe zu erhalten.
- **ClassifyTemperatures.mfx** wird ausgeführt. In diesem Mapping wird ein Parameter als Input erwartet, der mit Hilfe der Methode **`AddParameter`** bereitgestellt wurde.

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.
        'The ServerPath should be set before any other properties because it re-
initializes the server and resets all properties.
        'objMFS.ServerPath = "C:\Program Files (x86)
\Altova\MapForceServer2025\bin\MapForceServer.exe"
        'objMFS.ServerPath = "C:\Program
Files\Altova\MapForceServer2025\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-Schnittstelle

MapForce Server wird bei der Installation automatisch als COM-Serverobjekt registriert. Um zu überprüfen, ob die Registrierung erfolgreich war, öffnen Sie den Registrierungseditor (z.B. durch Eingabe von `regedit.exe` in die Befehlszeile). Wenn die Registrierung erfolgreich war, enthält die Registrierungsdatei die folgenden Klassen.

- **MapForce.Server** (für 32-Bit MapForce Server)
- **MapForce_x64.Server** (für 64-Bit MapForce Server)

Diese Klassen befinden sich unter `HKEY_LOCAL_MACHINE\SOFTWARE\Classes`.

Nach Registrierung des COM-Serverobjekts können Sie es von Applikationen und Skriptsprachen aus mit Unterstützung für COM-Aufrufe aufrufen. Wenn Sie den Pfad des MapForce Server Installationspakets ändern möchten, sollten Sie MapForce Server am besten zuerst deinstallieren und anschließend im gewünschten Ordner installieren. Auf diese Art wird die Aufhebung der Registrierung und die erneute Registrierung automatisch während der Installation vorgenommen.

5.2.1 C++-Beispiel

Im folgenden Beispiel wird gezeigt, wie Sie eine Mapping-Ausführungsdatei (.mfx) über C++-Code ausführen. Unter Windows stehen die Beispieldateien unter dem folgenden Pfad zur Verfügung: **C:\Programme\Altova\MapForceServer2025\etc\Examples**.

Voraussetzungen

Bevor Sie den nachstehenden Code ausführen, stellen Sie sicher, dass die folgenden Voraussetzungen erfüllt werden:

- MapForce Server ist installiert und lizenziert.
- MapForce Server steht als COM-Serverobjekt zur Verfügung (normalerweise geschieht dies automatisch bei der Installation von MapForce Server; Informationen, wie Sie überprüfen, ob die Registrierung erfolgreich war, finden Sie unter [Informationen zur COM-Schnittstelle](#)⁸⁸).

Ausführen des Mapping-Codes

Mit dem nachstehenden Code werden drei Server-Ausführungsdateien (.mfx) ausgeführt. Die Tabelle unten enthält eine Liste der von den einzelnen .mfx-Dateien erwarteten Input-Dateien sowie der nach der Ausführung erstellten Ausgabedateien.

Ausführungsdatei (.mfx)	Input	Augabe
TokenizeString.mfx	AltovaTools.xml	AltovaToolsFeatures.csv
SimpleTotal.mfx	ipo.xml	<i>String</i>
ClassifyTemperatures.mfx	Temperatures.xml	Temperatures_out.xml

Wenn Sie Altova MapForce installiert haben, haben Sie die Möglichkeit, sich zum besseren Verständnis die Original-Mappings, anhand derer die .mfx-Dateien kompiliert wurden, anzusehen. Diese haben die Namen

TokenizeString1.mfd, **SimpleTotal.mfd** bzw. **ClassifyTemperatures.mfd**. Sie finden die Mapping im folgenden Verzeichnis: **C:\Benutzer\<Benutzer>\Altova\MapForce2025\MapForceExamples**.

In diesem Beispiel geschieht Folgendes:

- Es wird eine neue Instanz von `Altova.MapForceServer.Server` erstellt. Dies ist das Objekt, mit dem Sie später arbeiten werden.
- Darin wird ein Arbeitsverzeichnis konfiguriert, in dem die Dateien ausgeführt werden. Die Input-Dateien sollten in diesem Verzeichnis vorhanden sein, falls Sie diese über einen relativen Pfad referenziert haben. Auch die Ausgabedateien werden in diesem Verzeichnis erstellt (siehe Tabelle oben).
- **TokenizeString.mfx** wird ausgeführt. Der Dateipfad wird als Argument für die Methode `Run` bereitgestellt (Beachten Sie, dass der Pfad relativ zum zuvor definierten Arbeitsverzeichnis ist). Nach erfolgreicher Ausführung wird im Arbeitsverzeichnis eine .csv-Datei, die die Mapping-Ausgabe darstellt, erstellt.
- **SimpleTotal.mfx** wird ausgeführt. Auch hier ist der Dateipfad wieder relativ zum Arbeitsverzeichnis. Mit diesem Mapping wird eine String-Ausgabe erzeugt, daher wird die Methode `GetOutputParameter` aufgerufen, um die String-Ausgabe zu erhalten.
- **ClassifyTemperatures.mfx** wird ausgeführt. In diesem Mapping wird ein Parameter als Input erwartet, der mit Hilfe der Methode `AddParameter` bereitgestellt wurde.

```
// 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.
```

```

// The ServerPath should be set before any other properties because it re-
initializes the server and resets all properties.
// pMFS.ServerPath = "C:\\Program Files (x86)\\Altova\\MapForceServer2025\\bin\\
\\MapForceServer.exe";
// pMFS.ServerPath = "C:\\Program Files\\Altova\\MapForceServer2025\\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-Beispiel

Im folgenden Beispiel wird gezeigt, wie Sie eine Mapping-Ausführungsdatei (.mfx) über VBScript-Code ausführen. Unter Windows stehen die Beispieldateien unter dem folgenden Pfad zur Verfügung: **C:\Programme\Altova\MapForceServer2025\etc\Examples**.

Bevor Sie den nachstehenden Code ausführen, stellen Sie sicher, dass die folgenden Voraussetzungen erfüllt werden:

- MapForce Server ist installiert und lizenziert
- MapForce Server steht als COM-Serverobjekt zur Verfügung (normalerweise geschieht dies automatisch bei der Installation von MapForce Server; Informationen, wie Sie überprüfen, ob die Registrierung erfolgreich war, finden Sie unter [Informationen zur COM-Schnittstelle](#)⁸⁸).

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\SysOW64\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.
'The ServerPath should be set before any other properties because it re-initializes the
server and resets all properties.
'objMFS.ServerPath = "C:\Program Files (x86)
\Altova\MapForceServer2025\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 direcorý
' 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-Beispiel

Microsoft Visual Basic for Applications (VBA) dient in erster Linie zum Automatisieren von Aufgaben in Microsoft Office. Sie können aber auch die MapForce Server API von VBA aus aufrufen und Mappings ausführen. Die folgende Anleitung wurde auf MapForce Server und VBA für Microsoft Office 2013 getestet. Bei Verwendung einer anderen VBA-Umgebung sind die einzelnen Schritte eventuell anders.

Voraussetzungen

Bevor Sie die MapForce Server API-Funktionen von Ihrem VBA-Projekt aus aufrufen können, benötigen Sie die folgenden Voraussetzungen:

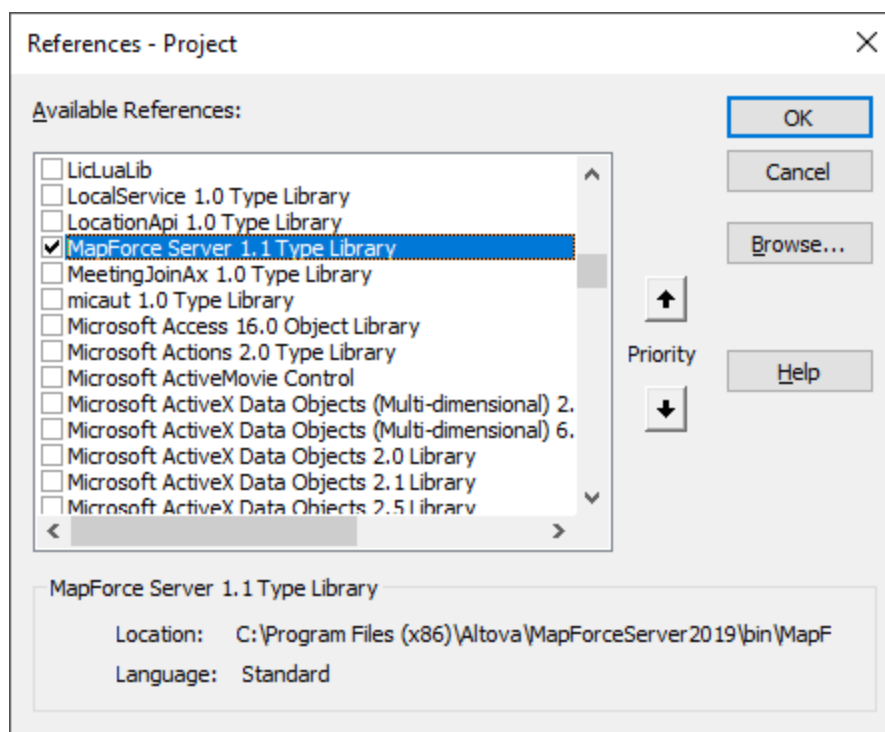
1. Microsoft Office und MapForce Server müssen auf demselben Rechner installiert sein.
2. Die Architektur von MapForce Server (32-Bit oder 64-Bit) muss mit der von Microsoft Office übereinstimmen. Wenn Sie z.B. VBA auf Microsoft Office 32-Bit ausführen, stellen Sie sicher, dass Sie die MapForce Server 32-Bit-Version verwenden. Um herauszufinden, ob Ihr Office-Produkt auf 64-Bit läuft, klicken Sie auf das Register **Datei**, dann auf **Konto** und anschließend auf "Über Excel" (oder "Über Word")
3. Die MapForce Server-Bibliothek muss von Ihrem VBA-Projekt aus referenziert werden (siehe Anleitung unten).

So fügen Sie in Ihrem VBA-Projekt eine Referenz zur MapForce Server-Bibliothek hinzu

1. Klicken Sie in einem Microsoft Office-Dokument (.docm, .xlsm), in dem Makros aktiviert sind, auf dem Register **Entwicklertools** auf **Visual Basic**.

Standardmäßig ist das Register **Entwicklertools** in Microsoft Office nicht aktiviert. Um das Register **Entwicklertools** in einem Office 2013-Programm zu aktivieren, klicken Sie mit der rechten Maustaste auf das Menüband und wählen Sie im Kontextmenü den Befehl **Menüband anpassen**. Aktivieren Sie anschließend im Dialogfeld "Optionen" unter "Hauptregisterkarten" das Kontrollkästchen **Entwicklertools**.

2. Klicken Sie in der VBA-Entwicklungsumgebung im Menü **Extras** auf **Verweise**.



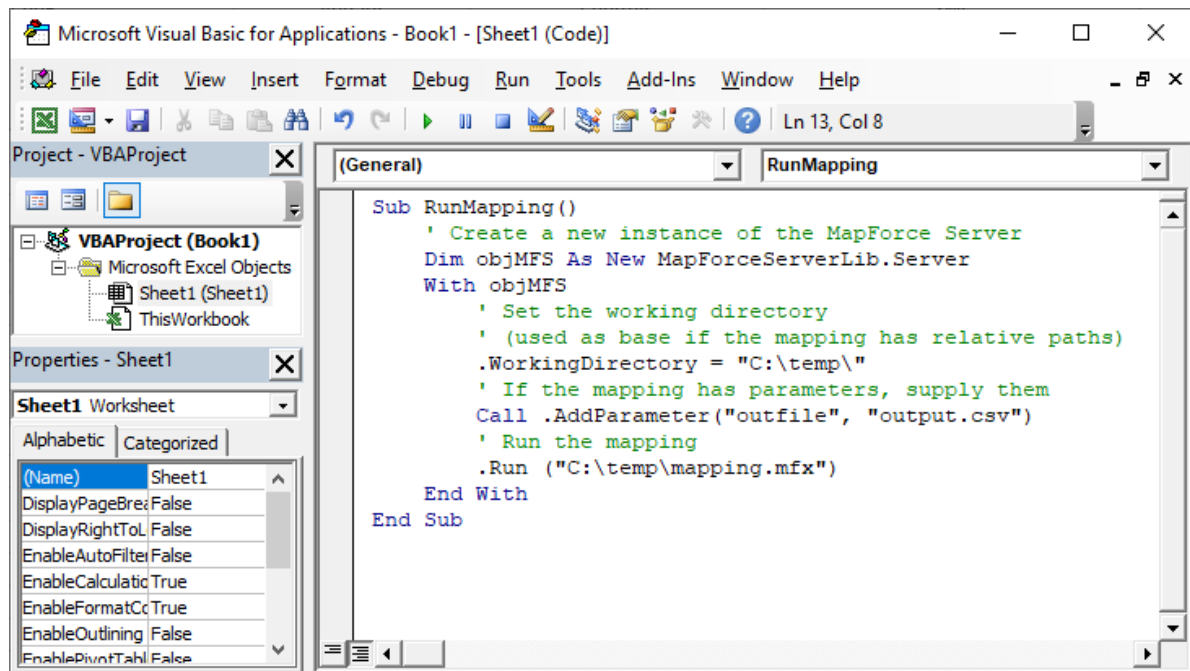
3. Klicken Sie, um die **MapForce Server-Typbibliothek** auszuwählen.

So rufen Sie die MapForce Server API auf

Nachdem Sie in Ihrem VBA-Projekt eine Referenz zur MapForce Server-Bibliothek hinzugefügt haben, können Sie den VBA-Code im Code-Editor-Fenster eingeben. Mit dem folgenden Beispielcode wird z.B. MapForce Server aufgerufen und eine ausführbare Mapping-Datei (**mapping.mfx**), die einen Input-Parameter namens "outfile" mit dem Wert "output.csv" erhält, ausgeführt.

```
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
```

Drücken Sie **F5**, um den VBA-Code zu debuggen und das Mapping auszuführen.



5.3 Java-Schnittstelle

Die API besteht aus einer JAR-Datei (**MapForceServer.jar**) und einer JNI-Datei (**MapForceServer.dll**). Beide Dateien sowie andere damit in Zusammenhang stehende API-Dateien befinden sich im Ordner `bin` des MapForce Server Installationsordners. Sie können diese Dateien entweder über ihren Originalordner referenzieren oder sie in einen anderen Ordner kopieren, falls Ihr Projekt dies erfordert. (Auf Windows-Systemen benötigen Sie Administratorrechte, um das Programm von seinem Originalordner aus zu starten).

Anmerkung: Wenn Sie eine 64-Bit-Version von MapForce Server installiert haben, so befinden sich die Dateien der 32-Bit-Version von **MapForceServer.jar** und **MapForceServer.dll** im Ordner `bin\API_32bit` des MapForce Server Installationsverzeichnisses. Sie benötigen diese Dateien, wenn Sie eine 32-Bit-Java-Version verwenden. Ebenso, wenn Sie eine 32-Bit-Version von MapForce Server installiert haben, so befinden sich die Dateien der 64-Bit-Version von **MapForceServer.jar** und **MapForceServer.dll** im Ordner `bin\API_64bit`. Sie benötigen diese Dateien, wenn Sie eine 64-Bit-Java-Version verwenden.

Um die MapForce Server API über Java-Code aufrufen zu können, fügen Sie die folgenden Referenzen zur `.classpath`-Datei Ihres Java-Projekts hinzu.

<code>MapForceServer.jar</code>	Die Bibliothek, die mit MapForce Server kommuniziert
<code>MapForceServer_JavaDoc.zip</code>	Die Dokumentation zur MapForce Server API

Des Weiteren muss der `java.library.path` den Ordner enthalten, in dem sich die JNI-Bibliotheksddatei (**MapForceServer.dll**) befindet.

Wenn Sie Ihr Projekt auf einem Applikationsserver bereitstellen, stellen Sie sicher, dass **MapForceServer.jar** und **MapForceServer.dll** auf dem Server korrekt mit Java konfiguriert sind.

Ein Beispiel dafür, wie die API-Bibliotheksddateien verwendet werden, sehen Sie in der Beispiel-Batch-Datei **buildAndRun.bat** (siehe unten), im Ordner `etc\Examples\Java` Ihres MapForce Server-Installationsverzeichnisses.

Erstellen und Ausführen eines Java-Programms über die API

Ein Beispiel dafür, wie Sie ein Java-Programm, das die MapForce Server-API verwendet, erstellen und ausführen können, finden Sie in der Beispiel-Batch-Datei **buildAndRun.bat**. Sie können diese Datei für Ihre eigenen Java-Programme anpassen und wiederverwenden.

Starten Sie die Batch-Datei mit dem folgenden Befehl über eine Befehlszeilenschnittstelle:

```
buildAndRun "path_to_Java_bin_folder"
```

Anmerkung: Um zu überprüfen, ob sich Java in Ihrem Classpath befindet, führen Sie den Befehl `java --version` aus. Wenn sich Java nicht in Ihrem Classpath befindet, müssen Sie den Pfad als Parameter des **buildAndRun**-Befehls angeben. Wenn der Pfad Leerzeichen enthält, setzen Sie ihn in Anführungszeichen.

 Codefragment von `buildAndRun.bat`

```
@echo off
if %1.==. goto error
```

```
REM The location of the JAVA API binaries, the JAR file and the JNI library.
REM Adapt to your needs.
SETLOCAL
Set JavaAPIBinPath=%PROGRAMFILES%\Altova\MapForceServer2025\bin

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

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

@echo off
goto end

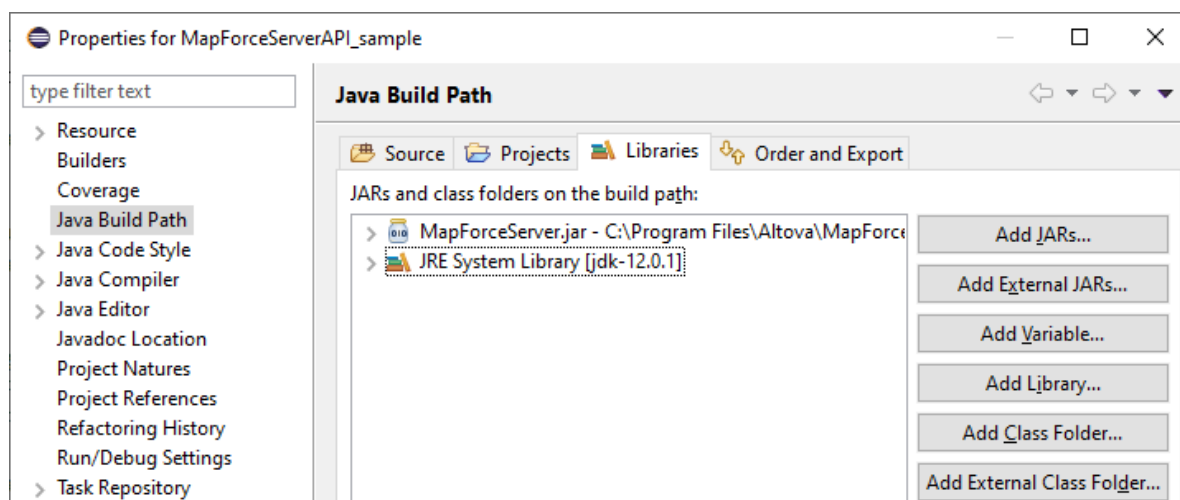
:error
echo Usage: buildAndRun "<path_to_java_bin_folder>"

:end
```

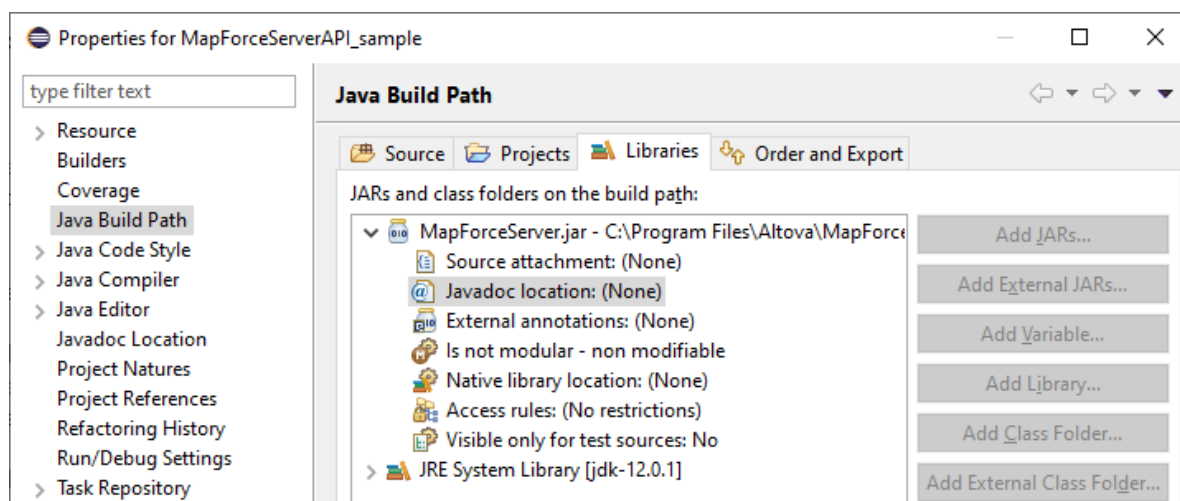
Hinzufügen der Bibliotheksreferenzen in Eclipse

Wenn Sie als Java-Entwicklungsumgebung Eclipse verwenden, können Sie die benötigte Bibliotheksreferenz durch Bearbeiten der Eigenschaften des Java-Projekts zum CLASSPATH hinzufügen (siehe unten). Optional können Sie auch Dokumentation im JavaDoc-Format an die .jar-Bibliothek anhängen. Die JavaDoc befindet sich im Ordner **bin** des MapForce Server-Installationsordners; in der Anleitung unten wird beschrieben, wie Sie die JavaDoc-Dokumentation von Eclipse aus anzeigen können.

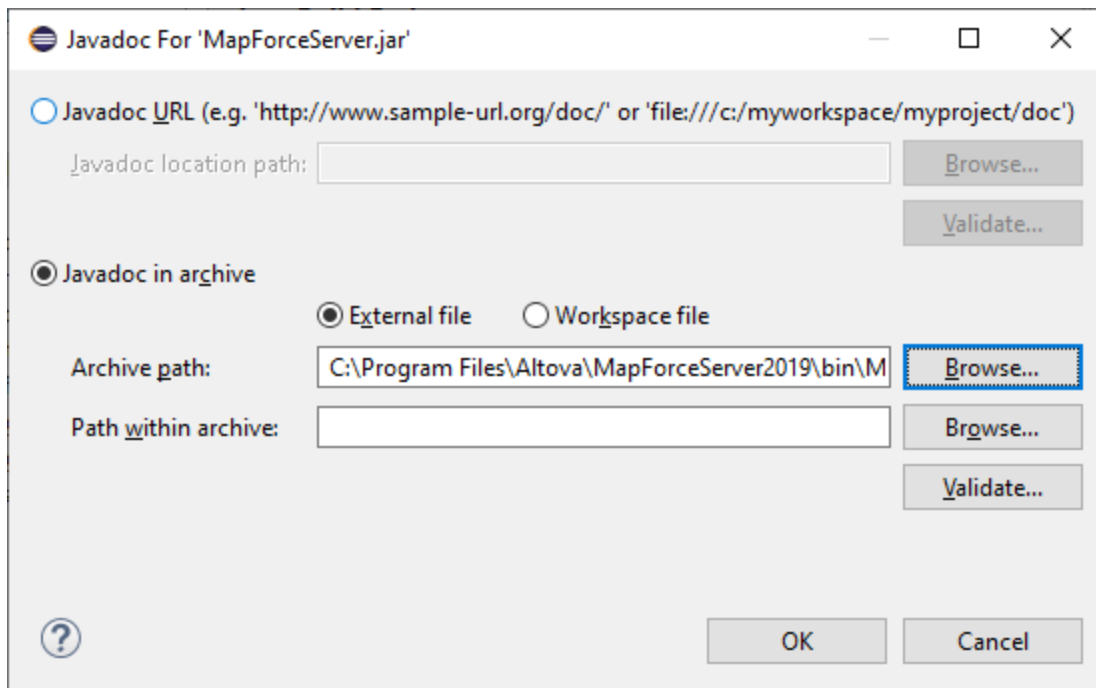
1. Klicken Sie, während das Projekt in Eclipse geöffnet ist im Menü **Project** auf **Properties**.
2. Klicken Sie auf **Java Build Path**.
3. Klicken Sie auf dem Register **Libraries** auf **Add External JARs** und navigieren Sie zur Datei MapForceServer.jar im MapForce Server-Installationsordner.



4. Fügen Sie optional das Javadoc-Archiv hinzu, erweitern Sie den **MapForceServer.jar**-Eintrag und doppelklicken Sie anschließend auf den Eintrag **Javadoc location: (None)**.



5. Stellen Sie sicher, dass die Optionen **Javadoc in archive** und **External file** aktiviert sind und navigieren Sie anschließend zur Datei MapForceServer_JavaDoc.zip im MapForce Server-Installationsordner.



6. Klicken Sie auf OK.

Unten sehen Sie ein Beispiel für eine Eclipse .classpath-Datei, wenn Sie die Dateien vom Originalinstallationsordner auf einem 64-Bit-Windows-System mit einer 64-Bit-MapForce Server Edition referenzieren (die entsprechenden Zeilen sind gelb markiert):

```
<?xml version="1.0" encoding="UTF-8"?>
<classpath>
  <classpathentry kind="src" path=""/>
  <classpathentry kind="lib" path="C:/Program
Files/Altova/MapForceServer2025/bin/MapForceServer.jar">
    <attributes>
      <attribute name="javadoc_location" value="jar:file:/C:/Program%
20Files/Altova/MapForceServer2025/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-Beispiel

Im folgenden Beispiel sehen Sie, wie Sie eine Mapping-Ausführungsdatei (.mfx) von Java-Code aus ausführen. Unter Windows stehen die Beispieldateien unter dem folgenden Pfad zur Verfügung: **C:**

\\Programme\\Altova\\MapForceServer2025\\etc\\Examples. Folgende Beispieldateien stehen zur Verfügung:

- **TokenizeString.mfx** - eine MapForce Server-Ausführungsdatei, die vom Java-Programm ausgeführt wird.
- **AltovaTools.xml** - eine XML-Datei, die die Input-Daten für das Mapping-Programm bereitstellt.

Auf Linux- und Mac-Plattformen werden keine Beispiele vorinstalliert, Sie können jedoch eine ausführbare .mfx-Datei wie die in diesem Beispiel gezeigte folgendermaßen vorbereiten:

1. Öffnen Sie die gewünschte Mapping-Design-Datei (.mfd) mit MapForce unter Windows.
2. Wählen Sie im Menü **Datei** den Befehl **Mapping-Einstellungen** und deaktivieren Sie das Kontrollkästchen **Pfade im generierten Code absolut machen**, falls es aktiviert ist.
3. Öffnen Sie für jede Mapping-Komponente das Dialogfeld **Eigenschaften** (z.B. durch Doppelklick auf die Titelleiste) und ändern Sie alle Dateipfade von absoluten in relative um. Aktivieren Sie außerdem das Kontrollkästchen **Alle Dateipfade relativ zur MFD-Datei speichern**. Sie können alle Input-Dateien und Schemas aus praktischen Gründen in denselben Ordner wie das Mapping kopieren und diese einfach über den Dateinamen referenzieren. Nähere Informationen zum Arbeiten mit relativen und absoluten Pfaden bei der Erstellung eines Mappings finden Sie in der Dokumentation zu MapForce.
4. Wählen Sie im Menü **Datei** den Befehl **Zu MapForce Server-Ausführungsdatei kompilieren**. Dadurch wird die .mfx-Datei generiert, die Sie in der Folge mit MapForce Server ausführen werden, wie im Codefragment unten gezeigt.

Voraussetzungen

Bevor Sie den nachstehenden Code ausführen, stellen Sie sicher, dass die folgenden Voraussetzungen erfüllt werden:

- MapForce Server ist installiert und lizenziert.
- Die java-CLASSPATH-Datei enthält eine Referenz zur **MapForceServer.jar**-Bibliothek (Beispiel siehe [Informationen zur Java-Schnittstelle](#)⁹⁷).

Wenn Sie, wie unten gezeigt, eine benutzerdefinierte .mfx-Datei verwenden, variieren die Voraussetzungen eventuell je nach Art der vom Mapping verarbeiteten Daten. Nähere Informationen dazu finden Sie unter [Vorbereiten von Mappings für die Server-Ausführung](#)³⁴.

Das Java-Beispielprojekt befindet sich unter Windows im Verzeichnis "Programme", für das Sie Administratorrechte benötigen. Starten Sie Ihre Java-Entwicklungsumgebung (z.B. Eclipse) entweder als Administrator oder kopieren Sie das Beispiel in einen anderen Ordner, für den Sie keine Administratorrechte benötigen.

Ausführen des Java-Programms

Mit dem nachstehenden Codefragment wird zuerst ein MapForce Server-Objekt erstellt. Als nächstes wird das Arbeitsverzeichnis, in dem die Applikation nach Input-Dateien für das Mapping suchen soll und in dem die Mapping-Ausgabedateien erzeugt werden sollen, definiert. Wie oben erwähnt, werden in der Beispiel-Mapping-

Datei Daten aus einer XML-Quelldatei ausgelesen. Stellen Sie also sicher, dass sich sowohl die XML-Datei als auch das dazugehörige Schema im Arbeitsverzeichnis befindet.

Die Methode `setServerPath` definiert den Pfad zur ausführbaren MapForce Server-Datei. Auf Ubuntu wäre der Pfad **`/opt/Altova/MapForceServer2025/bin/mapforceserver`**. Sie können `setServerPath` weglassen, wenn Sie die **`MapForceServer.jar`** -Datei nicht aus ihrem Standardordner verschoben haben.

Mit der Methode `run` wird schließlich ein Mapping (`.mfx`-Datei) ausgeführt, das mit MapForce kompiliert wurde (z.B. **`TokenizeString.mfx`**). Bei Erfolg generiert das unten stehende Programm eine CSV- und eine XML-Datei im Arbeitsverzeichnis sowie Ausgabertext in der Befehlszeile. Bei einem Fehler versucht das Programm die letzte von MapForce Server generierte Ausführungsmeldung zu drucken.

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

public class Program {

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

            // The default location of the server is the directory that the java native
library is in.
            // With the following line you could select a different server binary.
            // objMFS.setServerPath(strServerPath);
            // The ServerPath should be set before any other properties because it re-
initializes the server and resets all properties.

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

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

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

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

```
//
-----
// An example creating a simple output so that we can retrieve the result
explicitly
    System.out.println("\nExecuting SimpleTotal.mfx...");
    if (objMFS.run("SimpleTotal.mfx"))
        System.out.println("Mapping result is: " +
objMFS.getOutputParameter("total"));
    else
    {
        // execution failed (e.g. somebody deleted file ipo.xml)
        System.out.println(objMFS.getLastExecutionMessage());
    }
//
-----
// An example with parameterized input
// the default of 'lower = 5' gets changed to the value '10'
// mfx reads file Temperatures.xml and writes its output to
Temperatures_out.xml.
    System.out.println("\nExecuting ClassifyTemperatures.mfx with parameter
'lower' set to '10' ...");
    objMFS.addParameter("lower", "10");
    if (objMFS.run("ClassifyTemperatures.mfx"))
        System.out.println("File Temperatures_out.xml has been written
successfully.");
    else
    {
        // execution failed. maybe no write permissions in working directory? Run
this program as administrator.
        System.out.println(objMFS.getLastExecutionMessage());
    }

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

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

}

}
```

5.4 Beispiel: Ausführen eines Mappings mit Parametern

In diesem Beispiel wird gezeigt, wie Sie ein MapForce-Mapping zu einer MapForce Server-Ausführungsdatei (.mfx) kompilieren und diese über die MapForce API ausführen. Im Beispiel wird ein Szenario beschrieben, in dem der Name der Input-Datei im Mapping als Parameter verwendet wird. Siehe dazu auch die vorherigen Beispiele zu [C#](#)⁸¹, [C++](#)⁸⁸, [VB.NET](#)⁸⁴, [VBScript](#)⁹¹ oder [Java](#)¹⁰¹.

In diesem Beispiel wird zur Anzeige und zum Verständnis des ursprünglichen Mapping-Designs MapForce verwendet. Das Mapping wird mit MapForce zu einer MapForce Server Ausführungsdatei (.mfx) kompiliert. Auch Einstellungen wie relative anstelle von absoluten Pfaden werden in MapForce vorgenommen.

Die in diesem Beispiel verwendete Server-Plattform ist Windows. Es könnte sich hierbei entweder um denselben Rechner handeln, auf dem auch MapForce installiert ist, oder um einen anderen. Sie können dieses Beispiel auch auf einem Linux- oder Mac-Rechner (in Java) ausführen, vorausgesetzt, Sie passen die Windows-Pfade an Ihre jeweilige Plattform an.

Voraussetzungen

Für die Ausführung dieses Mappings gelten dieselben Voraussetzungen wie für die zuvor beschriebenen Beispiele für [C#](#)⁸¹, [C++](#)⁸⁸, [VB.NET](#)⁸⁴, [VBScript](#)⁹¹ oder [Java](#)¹⁰¹.

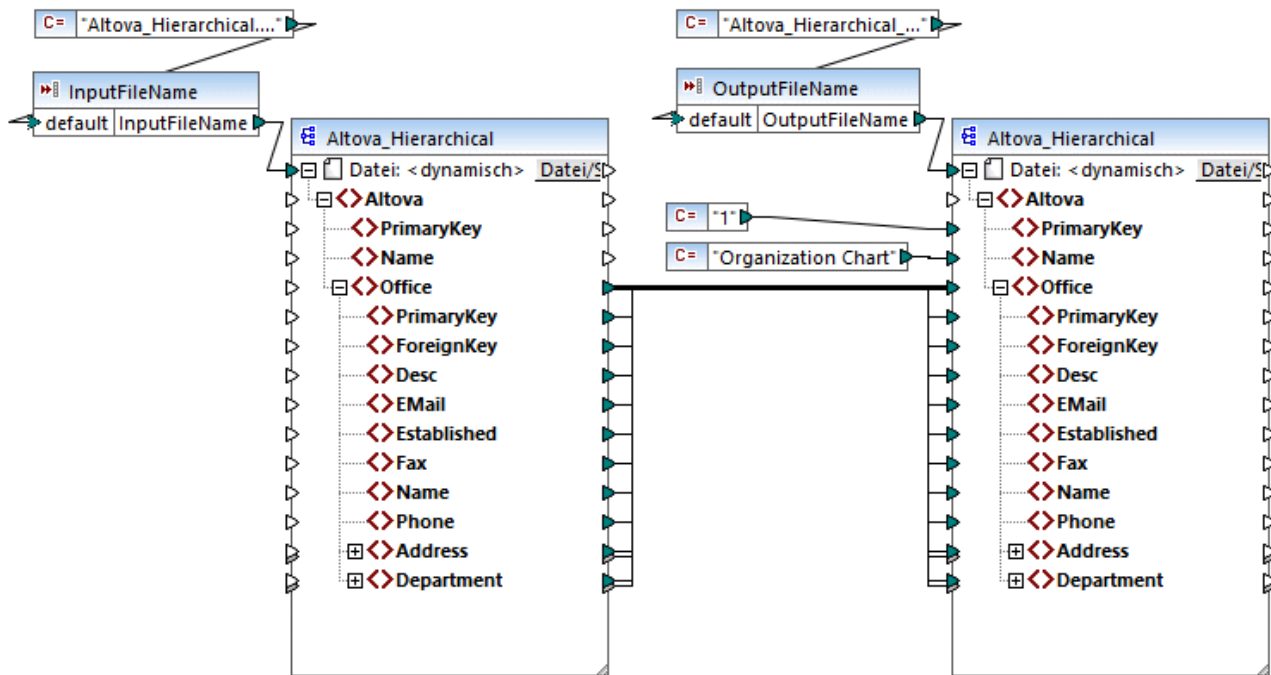
Vorbereiten von Mappings für die Server-Ausführung

Nachdem Sie MapForce installiert und zum ersten Mal gestartet haben, steht eine Reihe von Demo-Mapping-Design-Dateien unter dem folgenden Pfad zur Verfügung:

C:\Benutzer\<Benutzername>\Dokumente\Altova\MapForce2025\MapForce Examples

Der obige Pfad muss entsprechend geändert werden, wenn Sie eine andere Version vom MapForce haben.

Das in diesem Beispiel verwendete Mapping-Design hat den Namen **FileNamesAsParameters.mfd**. Wie Sie in der Abbildung unten sehen, handelt es sich hierbei um ein einfaches Mapping, in dem Daten beinahe unverändert aus einer XML-Quelldatei in eine XML-Zielfeile kopiert werden. Nur die Felder **PrimaryKey** und **Name** der XML-Zielfeile werden mit Konstanten aus dem Mapping befüllt.

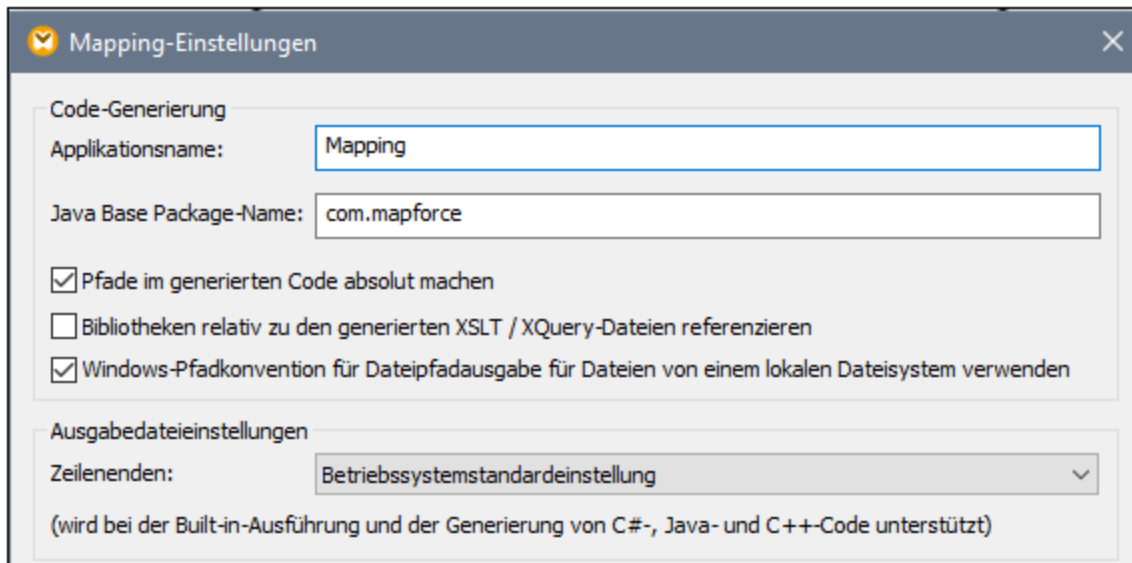


FileNamesAsParameters.mfd

Das Mapping hat zwei Input-Parameter: **InputFileName** und **OutputFileName**. Damit in MapForce eine Vorschau auf das Mapping angezeigt werden kann, werden die Parameterwerte durch zwei Konstanten bereitgestellt. Wie weiter unten beschrieben, haben Sie die Möglichkeit, die Parameterwerte bei Ausführung des Mappings mit MapForce Server durch Ihre eigenen zu ersetzen.

Beachten Sie, dass der Dateiname laut Konfiguration sowohl für die Quell- als auch für die Zielkomponente des Mappings dynamisch aus dem Mapping abgerufen wird. In der obersten Zeile der Komponente steht dazu ein *Datei <dynamisch>* Input-Konnektor zur Verfügung, über den Daten aus dem entsprechenden Mapping-Parameter (**InputFileName** oder **OutputFileName**) ausgelesen werden. Sie können diese Option in MapForce durch Klicken auf die Schaltfläche **Datei/String** in der rechten oberen Ecke der Komponente definieren oder ändern. Beachten Sie, dass der als Parameter bereitgestellte Input-Dateiname eine physische XML-Datei, die die Input-Daten für das Mapping enthält (in diesem Beispiel **Altova_Hierarchichal.xml**, aus demselben Ordner wie die Mapping-Design-Datei) referenzieren muss. Der Ausgabedateiname kann jeder gültige Name sein, z.B. **Output.xml**.

Bevor Sie das Mapping zu einer MapForce Server Ausführungsdatei (.mfx) kompilieren, sollten Sie normalerweise die Mapping-Einstellungen überprüfen. Klicken Sie mit der rechten Maustaste in einen leeren Bereich des Mappings und wählen Sie im Kontextmenü den Befehl **Mapping-Einstellungen**.



Ändern Sie die Einstellungen für dieses Beispiel wie oben gezeigt. Beachten Sie vor allem, dass alle absoluten Pfade zu im Mapping verwendeten Input- oder Output-Dateien relativ zur Mapping-Design-Datei (.mfd) aufgelöst werden, wenn Sie das Kontrollkästchen **Pfade im generierten Code absolut machen** deaktivieren.

MapForce Server sucht zur Laufzeit im Arbeitsverzeichnis des Programms nach diesen Pfaden. Welches Standard-Arbeitsverzeichnis verwendet wird, hängt von der Plattform ab, über die Sie die MapForce Server API aufrufen. Im Fall einer C#-Applikation ist es dasselbe Verzeichnis wie das der ausführbaren Datei. Außerdem können Sie, wie weiter unten gezeigt, das Arbeitsverzeichnis mit Hilfe einer API-Eigenschaft namens [WorkingDirectory](#)⁽¹¹⁹⁾ (oder gegebenenfalls der Java-Methode [setWorkingDirectory](#)⁽¹⁴⁰⁾) ändern.

Anmerkungen:

- Im Dialogfeld "Mapping-Einstellungen" sind die einzigen Einstellungen, die sich direkt auf die Kompilierung der .mfx-Datei auswirken **Pfade im generierten Code absolut machen** und **Zeilenenden**.
- In diesem Mapping war es nicht notwendig, die einzelnen Mapping-Komponenten in MapForce zu bearbeiten, um absolute Pfade in relative zu ändern, da alle Pfade bereits relativ waren. Falls dies nicht der Fall ist, müssten Sie diesen zusätzlichen Schritt ebenfalls durchführen, siehe [Vorbereiten von Mappings für die Server-Ausführung](#)⁽³⁴⁾.

In diesem Beispiel wird das Verzeichnis **C:\MappingExample** als Arbeitsverzeichnis verwendet. Kopieren Sie daher die vom Mapping referenzierte Input-Datei **Altova_Hierarchical.xml** von **C:\Benutzer\<Benutzername>\Dokumente\Altova\MapForce2025\MapForceExamples** in das Arbeitsverzeichnis.

Sowohl bei der Quell- als auch bei der Zieldatei in diesem Beispiel handelt es sich um XML-Dateien. Die XML-Schema-Datei ist auf dem Server nicht vonnöten, da die Informationen daraus bei der Kompilierung in die .mfx-Datei eingebettet wurden. Wenn in Ihrem Mapping andere Komponententypen (z.B. Datenbanken) verwendet werden, gelten allenfalls zusätzliche Voraussetzungen, wie unter [Vorbereiten von Mappings für die Server-Ausführung](#)⁽³⁴⁾ beschrieben.

Um das Mapping schließlich zu einer Server-Ausführungsdatei zu kompilieren, gehen Sie folgendermaßen vor:

- Klicken Sie im Menü **Datei** auf den Befehl **Zu MapForce Server-Ausführungsdatei kompilieren** und wählen Sie ein Zielverzeichnis aus. Das Zielverzeichnis ist in diesem Beispiel dasselbe wie das Arbeitsverzeichnis, nämlich **C:\MappingExample**.

In den folgenden Codefragmenten wird gezeigt, wie Sie das Mapping über die MapForce Server API von verschiedenen Umgebungen aus ausführen. Beachten Sie, dass beide Parameter in den unten stehenden Codefragmenten durch denselben Namen, den sie auch im Mapping-Design haben, referenziert werden. Außerdem hat der Parameterwert denselben Datentyp, der auch im Mapping definiert ist (in diesem Fall String).

Ausführen des Mappings über C++

Sie können die .mfx-Datei nun mit C++-Code, wie dem unten gezeigten, ausführen. Um diesen Code erfolgreich ausführen zu können, stellen Sie sicher, dass MapForce Server eine gültige Lizenz hat, siehe auch CLI-Befehl [verifylicense](#)⁷⁷.

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;

}

```

Ausführen des Mappings über C#

Sie können die .mfx-Datei nun mit C#-Code, wie dem unten gezeigten, ausführen. Um diesen Code erfolgreich ausführen zu können, fügen Sie zuerst in Visual Studio eine Referenz zur MapForce Server DLL hinzu, wie unter [.NET-Schnittstelle](#)⁸⁰ beschrieben, und vergewissern Sie sich, dass MapForce Server eine gültige Lizenz hat, siehe auch CLI-Befehl [verifylicense](#)⁷⁷.

C#

```

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

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

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

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

```

Ausführen des Mappings über Java

Sie können die .mfx-Datei nun mit Java-Code, wie dem unten gezeigten, ausführen. Um dieses Codefragment erfolgreich ausführen zu können, stellen Sie sicher, dass:

- die Java-CLASSPATH-Umgebungsvariable eine Referenz zur **MapForceServer.jar**-Bibliothek enthält, wie unter [Informationen zur Java-Schnittstelle](#)⁹⁷ beschrieben.
- MapForce Server eine gültige Lizenz hat, siehe auch CLI-Befehl [verifylicense](#)⁷⁷.

Java

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

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

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

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

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

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

5.5 API-Referenz (COM, .NET)

Dieser Abschnitt enthält allgemeine Informationen über die Elemente der MapForce Server API (wie z.B. Schnittstellen und Methoden), die für Code, der für die COM- oder .NET-Plattform geschrieben wurde, verwendet werden können.

5.5.1 Schnittstellen

5.5.1.1 IServer

Die IServer-Schnittstelle erstellt eine neue MapForce Server-Objektinstanz und bietet Zugriff auf den MapForce Server-Prozessor.

Wenn Sie C++ unter der COM-Plattform verwenden, lautet der Name der Hauptschnittstelle IServer. Wenn Sie eine .NET-Sprache wie z.B. C# oder VB.NET verwenden, lautet der Name der Hauptschnittstelle Server.

Eigenschaften

Name	Beschreibung
APIMajorVersion ¹¹³	Schreibgeschützt. Ruft die Hauptversion der MapForce Server API ab. Diese stimmt unter Umständen nicht mit der Produktversion überein, wenn die API mit einem anderen Server verbunden ist.
APIMinorVersion ¹¹⁴	Schreibgeschützt. Ruft die Nebenversion der MapForce Server API ab. Diese stimmt unter Umständen nicht mit der Produktversion überein, wenn die API mit einem anderen Server verbunden ist.
APIServicePackVersion ¹¹⁴	Schreibgeschützt. Ruft die Service Pack-Version der MapForce Server API ab. Diese stimmt unter Umständen nicht mit der Produktversion überein, wenn die API mit einem anderen Server verbunden ist.
Is64Bit ¹¹⁵	Schreibgeschützt. Gibt true zurück, wenn der MapForce Server-Prozessor eine ausführbare 64-Bit-Datei ist.
LastExecutionMessage ¹¹⁵	Schreibgeschützt. Ruft die beim letzten Run -Befehl erhaltene Meldung ab.
MajorVersion ¹¹⁶	Schreibgeschützt. Ruft die Hauptversion des Produkts als Anzahl von Jahren seit 1998 ab (z.B. "20" für Altova MapForce Server 2018).
MinorVersion ¹¹⁶	Schreibgeschützt. Ruft die Nebenversion des Produkts ab (z.B. "2" für Altova MapForce Server 2018 r2).

Name	Beschreibung
ProductName ¹¹⁷	Schreibgeschützt. Ruft den Namen des Produkts ab (z.B. "Altova MapForce Server").
ProductNameAndVersion ¹¹⁷	Schreibgeschützt. Ruft den vollständigen Namen und die Version des Produkts ab (z.B. "Altova MapForce Server 2018 r2 sp1 (x64)").
ServerPath ¹¹⁸	Ruft den Pfad der ausführbaren MapForce Server-Datei ab bzw. definiert ihn.
ServicePackVersion ¹¹⁸	Schreibgeschützt. Ruft die Service Pack-Version des Produkts ab (z.B. "1" für Altova MapForce Server 2018 r2 sp1).
WorkingDirectory ¹¹⁹	Ruft das aktuelle Verzeichnis für ausgeführte Aufträge ab bzw. definiert es (relative Pfade werden anhand des Arbeitsverzeichnisses aufgelöst).

Methoden

Name	Beschreibung
AddCredentialProperty ¹¹⁹	Fügt eine Eigenschaft zu den aktuellen Anmeldeinformationen hinzu (z.B. den Benutzernamen, das Passwort oder beides). Das erste Argument definiert den Namen der Eigenschaft, das zweite Argument den Wert der Eigenschaft. Gültige Eigenschaftsnamen: username, password. In der MapForce Server Advanced Edition wird zusätzlich dazu der Eigenschaftsname oauth:token unterstützt. Diese Methode muss nach BeginCredential() und vor EndCredential() aufgerufen werden.
AddParameter ¹²⁰	Weist einem im Mapping definierten Parameter einen Wert zu. Das erste Argument definiert den Namen des Parameters, wie er im Mapping definiert wurde; das zweite Argument definiert den Parameterwert.
BeginCredential ¹²¹	Erstellt neue Anmeldeinformationen mit dem als Argument bereitgestellten Namen. Wenn Sie diese Methode aufrufen, müssen Sie mittels AddCredentialProperty() auch Eigenschaften zu dieser hinzufügen und die Anmeldeinformationen schließlich durch Aufruf von EndCredential() schließen.
ClearCredentialList ¹²²	Löscht die zuvor definierte Liste der Anmeldeinformationen. Alle Anmeldeinformationen sind während der Lebensdauer des Objekts gültig. Rufen Sie diese Methode auf, wenn Sie alle zuvor gesetzten Anmeldeinformationen explizit löschen möchten.

Name	Beschreibung
ClearOptions ¹²²	Löscht die Liste der zuvor mit der Methode <code>SetOption</code> definierten Optionen. Alle mit der <code>SetOption</code> -Methode definierten Optionen sind während der Lebensdauer des Objekts gültig. Rufen Sie diese Methode auf, wenn Sie alle zuvor gesetzten Optionen explizit löschen möchten.
ClearParameterList ¹²³	Löscht die Liste der zuvor mit der Methode <code>AddParameter</code> definierten Parameter.
EndCredential ¹²³	Schließt ein zuvor mit der Methode <code>BeginCredential</code> definiertes Anmeldeinformationen-Objekt.
GetOutputParameter ¹²⁴	Ruft den mit dem letzten <code>run</code> -Befehl generierten String Output ab. Gibt Null zurück, wenn kein Output generiert wurde. Für diese Funktion ist ein String-Parameter, der den Namen der Output-Komponente in MapForce definiert, erforderlich.
Run ¹²⁵	Führt eine MapForce Server-Ausführungsdatei (.mfx-Datei) aus. Gibt bei Erfolg true zurück, andernfalls false .
SetOption ¹²⁵	Definiert vor Ausführung des Mappings eine Option. Das erste Argument definiert den Namen der Option, das zweite Argument definiert den Optionswert. Diese Methode ist besonders dann nützlich, wenn in einem Mapping globale Ressourcen verwendet werden (siehe Globale Altova-Ressourcen ⁴⁰). Derzeit werden die folgenden Optionen unterstützt: • globalresourcefile (oder gr) - Der Pfad zu einer globalen Ressourcendatei. (Wenn diese Option definiert ist, muss auch ein Name für die Konfiguration der globalen Ressource definiert werden, siehe nächster Eintrag). • globalresourcefile (oder gr) - Name einer globalen Ressourcenkonfiguration. (Wenn diese Option definiert ist, muss ein Dateipfad zu einer globalen Ressourcen-Datei definiert werden, siehe vorheriger Eintrag). • catalog - Pfad zu einer benutzerdefinierten RootCatalog.xml-Datei. Mit Hilfe dieser Option können Sie eine benutzerdefinierte Katalogdatei für die Auflösung von im Mapping verwendeten URLs definieren. Der Standardkatalog befindet sich im Unterverzeichnis "etc" des Programminstallationsverzeichnis. • taxonomy-package - Der Pfad zu einem benutzerdefinierten XBRL-Taxonomiepaket, falls eines für das Mapping erforderlich ist. • taxonomy-packages-config-file - Der Pfad zu einer benutzerdefinierten XBRL-Taxonomiepaketkonfiguration, falls eine für das Mapping erforderlich ist.

Name	Beschreibung
	Alle definierten Optionen sind während der Lebensdauer des Objekts gültig. Wenn Sie eine Option mit demselben Namen zwei Mal definieren, wird die vorherige Option außer Kraft gesetzt. Um alle Optionen explizit zu löschen, rufen Sie die Methode <code>ClearOptions()</code> auf.
StopServerProcess ¹²⁷	Mit dieser Methode wird der mit dem COM-Objekt verbundene Prozess <i>explizit</i> beendet, ohne dass das Objekt freigegeben wird. Der Prozess wird implizit beendet, wenn das COM-Objekt freigegeben wird.

Beispiele

In den folgenden Beispielen wird gezeigt, wie Sie MapForce Server in verschiedenen Sprachen initialisieren und ausführen:

- [C++ -Beispiel](#) ⁸⁸
- [C#-Beispiel](#) ⁸¹
- [VBScript-Beispiel](#) ⁹¹
- [VB.NET-Beispiel](#) ⁸⁴.

5.5.1.1.1 Eigenschaften

5.5.1.1.1.1 *APIMajorVersion*

Ruft die Hauptversion der MapForce Server API ab. Diese stimmt unter Umständen nicht mit der Produktversion überein, wenn die API mit einem anderen Server verbunden ist.

Signatur

APIMajorVersion : **Integer**

Allgemeine Signatur

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

C#

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

C++

```
ReadOnly Property APIMajorVersion As Integer
```

VB.NET

5.5.1.1.1.2 *APIMinorVersion*

Ruft die Nebenversion der MapForce Server API ab. Diese stimmt unter Umständen nicht mit der Produktversion überein, wenn die API mit einem anderen Server verbunden ist.

Signatur

```
APIMinorVersion : Integer
```

Allgemeine Signatur

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

C#

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

C++

```
ReadOnly Property APIMinorVersion As Integer
```

VB.NET

5.5.1.1.1.3 *APIServicePackVersion*

Ruft die Service Pack-Version der MapForce Server API ab. Diese stimmt unter Umständen nicht mit der Produktversion überein, wenn die API mit einem anderen Server verbunden ist.

Signatur

```
APIServicePackVersion : Integer
```

Allgemeine Signatur

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

C#

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

C++

```
ReadOnly Property APIServicePackVersion As Integer
```

VB.NET

5.5.1.1.1.4 Is64Bit

Gibt **true** zurück, wenn der MapForce Server-Prozessor eine ausführbare 64-Bit-Datei ist.

Signatur

```
Is64Bit : Boolean
```

Allgemeine Signatur

```
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

Ruft die beim letzten **Run**-Befehl erhaltene Meldung ab.

Signatur

```
LastExecutionMessage : String
```

Allgemeine Signatur

```
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*

Ruft die Hauptversion des Produkts als Anzahl von Jahren seit 1998 ab (z.B. "20" für Altova MapForce Server 2018).

Signatur

```
MajorVersion : Integer
```

Allgemeine Signatur

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

C#

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

C++

```
ReadOnly Property MajorVersion As Integer
```

VB.NET

5.5.1.1.1.7 *MinorVersion*

Ruft die Nebenversion des Produkts ab (z.B. "2" für Altova MapForce Server 2018 r2).

Signatur

```
MinorVersion : Integer
```

Allgemeine Signatur

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

C#

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

C++

```
ReadOnly Property MinorVersion As Integer
```

VB.NET

5.5.1.1.1.8 *ProductName*

Ruft den Namen des Produkts ab (z.B. "Altova MapForce Server").

Signatur

```
ProductName : String
```

Allgemeine Signatur

```
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*

Ruft den vollständigen Namen und die Version des Produkts ab (z.B. "Altova MapForce Server 2018 r2 sp1 (x64)").

Signatur

```
ProductNameAndVersion : String
```

Allgemeine Signatur

```
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

Ruft den Pfad der ausführbaren MapForce Server-Datei ab bzw. definiert ihn.

Signatur

```
ServerPath : String
```

Allgemeine Signatur

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

C#

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

C++

```
Property ServerPath As String
```

VB.NET

5.5.1.1.1.11 ServicePackVersion

Ruft die Service Pack-Version des Produkts ab (z.B. "1" für Altova MapForce Server 2018 r2 sp1).

Signatur

```
ServicePackVersion : Integer
```

Allgemeine Signatur

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

C#

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

C++

```
ReadOnly Property ServicePackVersion As Integer
```

VB.NET

5.5.1.1.1.12 WorkingDirectory

Ruft das aktuelle Verzeichnis für ausgeführte Aufträge ab bzw. definiert es (relative Pfade werden anhand des Arbeitsverzeichnisses aufgelöst).

Signatur

```
WorkingDirectory : String
```

Allgemeine Signatur

```
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 Methoden

5.5.1.1.2.1 AddCredentialProperty

Fügt eine Eigenschaft zu den aktuellen Anmeldeinformationen hinzu (z.B. den Benutzernamen, das Passwort oder beides). Das erste Argument definiert den Namen der Eigenschaft, das zweite Argument den Wert der Eigenschaft. Gültige Eigenschaftsnamen: username, password. In der MapForce Server Advanced Edition wird zusätzlich dazu der Eigenschaftsname oauth:token unterstützt.

Diese Methode muss nach BeginCredential() und vor EndCredential() aufgerufen werden.

Signatur

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

Allgemeine Signatur

```
void AddParameter(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

Parameter

Name	Typ	Beschreibung
bstrName	System.String	Definiert den Namen der Anmeldeinformationen-Eigenschaft.
bstrValue	System.String	Definiert den Wert der Anmeldeinformationen-Eigenschaft.

Beispiele

Im folgenden Codefragment wird gezeigt, wie Sie Anmeldeinformationen namens "mycredential" in C# deklarieren. Der Name der Anmeldeinformationen muss derjenige sein, der den Anmeldeinformationen in MapForce zum Zeitpunkt des Designs gegeben wurde.

```
//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

Weist einem im Mapping definierten Parameter einen Wert zu. Das erste Argument definiert den Namen des Parameters, wie er im Mapping definiert wurde; das zweite Argument definiert den Parameterwert.

Signatur

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

Allgemeine Signatur

```
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

Parameter

Name	Typ	Beschreibung
bstrName	String	Definiert den Parameternamen.
bstrValue	String	Definiert den Parameterwert.

5.5.1.1.2.3 BeginCredential

Erstellt neue Anmeldeinformationen mit dem als Argument bereitgestellten Namen. Wenn Sie diese Methode aufrufen, müssen Sie mittels `AddCredentialProperty()` auch Eigenschaften zu dieser hinzufügen und die Anmeldeinformationen schließlich durch Aufruf von `EndCredential()` schließen.

Signatur

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

Allgemeine Signatur

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

C#

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

C++

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

VB.NET

Parameter

Name	Typ	Beschreibung
bstrCredentialName	String	Definiert den Namen der Anmeldeinformationen, wie er in MapForce definiert wurde.

5.5.1.1.2.4 ClearCredentialList

Löscht die zuvor definierte Liste der Anmeldeinformationen. Alle Anmeldeinformationen sind während der Lebensdauer des Objekts gültig. Rufen Sie diese Methode auf, wenn Sie alle zuvor gesetzten Anmeldeinformationen explizit löschen möchten.

Signatur

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

Allgemeine Signatur

```
void ClearCredentialList()
```

C#

```
HRESULT ClearCredentialList();
```

C++

```
Sub ClearCredentialList()
```

VB.NET

5.5.1.1.2.5 ClearOptions

Löscht die Liste der zuvor mit der Methode SetOption definierten Optionen. Alle mit der SetOption-Methode definierten Optionen sind während der Lebensdauer des Objekts gültig. Rufen Sie diese Methode auf, wenn Sie alle zuvor gesetzten Optionen explizit löschen möchten.

Signatur

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

Allgemeine Signatur

```
void ClearOptions()
```

C#

```
HRESULT ClearOptions();
```

C++

```
Sub ClearOptions()
```

VB.NET

5.5.1.1.2.6 *ClearParameterList*

Löscht die Liste der zuvor mit der Methode `AddParameter` definierten Parameter.

Signatur

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

Allgemeine Signatur

```
void ClearParameterList()
```

C#

```
HRESULT ClearParameterList();
```

C++

```
Sub ClearParameterList()
```

VB.NET

5.5.1.1.2.7 *EndCredential*

Schließt ein zuvor mit der Methode `BeginCredential` definiertes Anmeldeinformationen-Objekt.

Signatur

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

Allgemeine Signatur

```
void EndCredential()
```

C#

```
HRESULT EndCredential();
```

C++

```
Sub EndCredential()
```

VB.NET

5.5.1.1.2.8 *GetOutputParameter*

Ruft den mit dem letzten run-Befehl generierten String Output ab. Gibt Null zurück, wenn kein Output generiert wurde. Für diese Funktion ist ein String-Parameter, der den Namen der Output-Komponente in MapForce definiert, erforderlich.

Signatur

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

Allgemeine Signatur

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

C#

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

C++

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

VB.NET

Parameter

Name	Typ	Beschreibung
bstrName	String	Definiert den Namen der Output-Komponente, wie er in MapForce angezeigt wird. Dieser Name wird in der Titelleiste jeder Komponente des Mappings angezeigt (oder wenn Sie mit der rechten Maustaste auf die Komponentenüberschrift klicken)

Name	Typ	Beschreibung
		und den Befehl Eigenschaften wählen.)

5.5.1.1.2.9 Run

Führt eine MapForce Server-Ausführungsdatei (.mfx-Datei) aus. Gibt bei Erfolg **true** zurück, andernfalls **false**.

Signatur

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

Allgemeine Signatur

```
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

Parameter

Name	Typ	Beschreibung
bstrMappingPath	String	Definiert den Pfad zur MapForce Server-Ausführungsdatei (.mfx)-Datei. Wenn Sie einen relativen Pfad definieren, wird dieser anhand des Arbeitsverzeichnisses aufgelöst. Sie können das Arbeitsverzeichnis über die Eigenschaft <code>workingDirectory</code> definieren.

5.5.1.1.2.10 SetOption

Definiert vor Ausführung des Mappings eine Option. Das erste Argument definiert den Namen der Option, das zweite Argument definiert den Optionswert. Diese Methode ist besonders dann nützlich, wenn in einem

Mapping globale Ressourcen verwendet werden (siehe [Globale Altova-Ressourcen](#)⁴⁰). Derzeit werden die folgenden Optionen unterstützt:

- **globalresourcefile (oder gr)** - Der Pfad zu einer globalen Ressourcendatei. (Wenn diese Option definiert ist, muss auch ein Name für die Konfiguration der globalen Ressource definiert werden, siehe nächster Eintrag).
- **globalresourcefile (oder gr)** - Name einer globalen Ressourcenkonfiguration. (Wenn diese Option definiert ist, muss ein Dateipfad zu einer globalen Ressourcen-Datei definiert werden, siehe vorheriger Eintrag).
- **catalog** - Pfad zu einer benutzerdefinierten **RootCatalog.xml**-Datei. Mit Hilfe dieser Option können Sie eine benutzerdefinierte Katalogdatei für die Auflösung von im Mapping verwendeten URLs definieren. Der Standardkatalog befindet sich im Unterverzeichnis "etc" des Programminstallationsverzeichnisses.
- **taxonomy-package** - Der Pfad zu einem benutzerdefinierten XBRL-Taxonomiepaket, falls eines für das Mapping erforderlich ist.
- **taxonomy-packages-config-file** - Der Pfad zu einer benutzerdefinierten XBRL-Taxonomiepaketkonfiguration, falls eine für das Mapping erforderlich ist.

Alle definierten Optionen sind während der Lebensdauer des Objekts gültig. Wenn Sie eine Option mit demselben Namen zwei Mal definieren, wird die vorherige Option außer Kraft gesetzt. Um alle Optionen explizit zu löschen, rufen Sie die Methode `ClearOptions()` auf.

Signatur

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

Allgemeine Signatur

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

C#

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

C++

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

VB.NET

Parameter

Name	Typ	Beschreibung
bstrName	String	Definiert den Namen der zu setzenden Option.
bstrValue	String	Definiert den Wert der zu setzenden Option.

5.5.1.1.2.11 StopServerProcess

Mit dieser Methode wird der mit dem COM-Objekt verbundene Prozess *explizit* beendet, ohne dass das Objekt freigegeben wird. Der Prozess wird implizit beendet, wenn das COM-Objekt freigegeben wird.

Signatur

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

Allgemeine Signatur

```
void StopServerProcess()
```

C#

```
Sub StopServerProcess()
```

VB.NET

5.6 API-Referenz (Java)

Dieser Abschnitt enthält allgemeine Informationen über die Elemente der MapForce Server API (wie z.B. Klassen und Methoden), die in Code, der für die Java-Plattform geschrieben wurde, verwendet werden können.

5.6.1 Klassen

5.6.1.1 MapForceServer

Die MapForceServer-Klasse erstellt eine MapForce Server-Objektinstanz und ermöglicht den Zugriff auf den MapForce Server-Prozessor.

Methoden

Name	Beschreibung
addCredentialPropertiesFromMap ¹³¹	Fügt zu den aktuellen Anmeldeinformationen Eigenschaften aus einer Anmeldeinformationen-Eigenschaftszuordnung hinzu. Diese Methode erhält als Argument eine Anmeldeinformationen-Eigenschaftszuordnung (<i>property_name</i> , <i>property_value</i>). Diese Methode muss nach <code>beginCredential()</code> und vor <code>endCredential()</code> aufgerufen werden. Alternativ dazu können Sie auch <code>AddCredentialProperty()</code> aufrufen.
addCredentialProperty ¹³²	Fügt eine Eigenschaft zu den aktuellen Anmeldeinformationen hinzu (z.B. den Benutzernamen, das Passwort oder beides). Das erste Argument definiert den Namen der Eigenschaft, das zweite Argument den Wert der Eigenschaft. Gültige Eigenschaftsnamen: <code>username</code> , <code>password</code> . In der MapForce Server Advanced Edition wird zusätzlich dazu der Eigenschaftsname <code>oauth:token</code> unterstützt. Diese Methode muss nach <code>beginCredential()</code> und vor <code>endCredential()</code> aufgerufen werden. Alternativ dazu können Sie auch <code>addCredentialPropertiesFromMap()</code> aufrufen.
addParameter ¹³³	Weist einem im Mapping definierten Parameter einen Wert zu.
beginCredential ¹³³	Erstellt neue Anmeldeinformationen mit dem als Argument bereitgestellten Namen. Wenn Sie diese Methode aufrufen, müssen Sie mit Hilfe von <code>addCredentialProperty()</code> oder <code>addCredentialPropertiesFromMap()</code> auch Eigenschaften dazu hinzufügen und die Anmeldeinformationen schließlich durch Aufruf von <code>endCredential()</code> schließen.
clearCredentialList ¹³³	Löscht die zuvor definierte Liste der Anmeldeinformationen. Alle Anmeldeinformationen sind während der Lebensdauer des Objekts gültig. Rufen Sie diese Methode auf, wenn Sie alle

Name	Beschreibung
	zuvor gesetzten Anmeldeinformationen explizit löschen möchten.
clearOptions ¹³⁴	Löscht die Liste der zuvor mit der Methode <code>setOption</code> definierten Optionen. Alle mit der <code>setOption</code> -Methode definierten Optionen sind während der Lebensdauer des Objekts gültig. Rufen Sie diese Methode auf, wenn Sie alle zuvor gesetzten Optionen explizit löschen möchten.
clearParameterList ¹³⁴	Löscht die Liste der zuvor mit der Methode <code>addParameter</code> definierten Parameter.
endCredential ¹³⁴	Schließt ein zuvor mit der Methode <code>beginCredential</code> definiertes Anmeldeinformationen-Objekt.
getAPIMajorVersion ¹³⁵	Ruft die Hauptversion der MapForce Server API ab. Diese stimmt unter Umständen nicht mit der Produktversion überein, wenn die API mit einem anderen Server verbunden ist.
getAPIMinorVersion ¹³⁵	Ruft die Nebenversion der MapForce Server API ab. Diese stimmt unter Umständen nicht mit der Produktversion überein, wenn die API mit einem anderen Server verbunden ist.
getAPIServicePackVersion ¹³⁵	Ruft die Service Pack-Version der MapForce Server API ab. Diese stimmt unter Umständen nicht mit der Produktversion überein, wenn die API mit einem anderen Server verbunden ist.
getLastExecutionMessage ¹³⁶	Ruft die beim letzten <code>run</code> -Befehl erhaltene Meldung ab.
getMajorVersion ¹³⁶	Ruft die Hauptversion des Produkts als Anzahl von Jahren seit 1998 ab (z.B. "20" für Altova MapForce Server 2018).
getMinorVersion ¹³⁶	Ruft die Nebenversion des Produkts ab (z.B. "2" für Altova MapForce Server 2018 r2).
getOutputParameter ¹³⁶	Ruft den mit dem letzten <code>run</code> -Befehl generierten String Output ab. Gibt Null zurück, wenn kein Output generiert wurde. Für diese Funktion ist ein String-Parameter, der den Namen der Output-Komponente in MapForce definiert, erforderlich.
getProductName ¹³⁷	Ruft den Namen des Produkts ab (z.B. "Altova MapForce Server").
getProductNameAndVersion ¹³⁷	Ruft den vollständigen Namen und die Version des Produkts ab (z.B. "Altova MapForce Server 2018 r2 sp1 (x64)").
getServerPath ¹³⁸	Ruft den Pfad zur ausführbaren Binärdatei des Servers ab.
getServicePackVersion ¹³⁸	Ruft die Service Pack-Version des Produkts ab bzw. definiert sie (z.B. "1" für Altova MapForce Server 2018 r2 sp1 (x64)).
getWorkingDirectory ¹³⁸	Ruft das aktuelle Arbeitsverzeichnis ab.

Name	Beschreibung
is64bit ¹³⁸	Gibt true zurück, wenn der MapForce Server-Prozessor eine ausführbare 64-Bit-Datei ist.
run ¹³⁹	Führt eine MapForce Server-Ausführungsdatei (.mfx-Datei) aus. Gibt bei Erfolg true zurück, andernfalls false .
setOption ¹³⁹	Definiert vor Ausführung des Mappings eine Option. Das erste Argument definiert den Namen der Option, das zweite Argument definiert den Optionswert. Diese Methode ist besonders dann nützlich, wenn in einem Mapping globale Ressourcen verwendet werden (siehe Globale Altova-Ressourcen⁴⁰). Derzeit werden die folgenden Optionen unterstützt: • globalresourcefile (oder gr) - Der Pfad zu einer globalen Ressourcendatei. (Wenn diese Option definiert ist, muss auch ein Name für die Konfiguration der globalen Ressource definiert werden, siehe nächster Eintrag). • globalresourcefile (oder gr) - Name einer globalen Ressourcenkonfiguration. (Wenn diese Option definiert ist, muss ein Dateipfad zu einer globalen Ressourcendatei definiert werden, siehe vorheriger Eintrag). • catalog - Pfad zu einer benutzerdefinierten RootCatalog.xml-Datei. Mit Hilfe dieser Option können Sie eine benutzerdefinierte Katalogdatei für die Auflösung von im Mapping verwendeten URLs definieren. Der Standardkatalog befindet sich im Unterverzeichnis "etc" des Programminstallationsverzeichnis. • taxonomy-package - Der Pfad zu einem benutzerdefinierten XBRL-Taxonomiepaket, falls eines für das Mapping erforderlich ist. • taxonomy-packages-config-file - Der Pfad zu einer benutzerdefinierten XBRL-Taxonomiepaketkonfiguration, falls eine für das Mapping erforderlich ist. Alle definierten Optionen sind während der Lebensdauer des Objekts gültig. Wenn Sie eine Option mit demselben Namen zwei Mal definieren, wird die vorherige Option außer Kraft gesetzt. Um alle Optionen explizit zu löschen, rufen Sie die Methode <code>clearOptions()</code> auf.
setServerPath ¹⁴⁰	Definiert den Pfad der ausführbaren MapForce Server-Datei.
setWorkingDirectory ¹⁴⁰	Definiert das aktuelle Verzeichnis für ausgeführte Aufträge (relative Pfade werden anhand des Arbeitsverzeichnisses aufgelöst).
stop ¹⁴¹	Beendet den Server-Vorgang.

Beispiele

Ein Beispiel zur Erstellung einer neuen Instanz von MapForceServer in Java finden Sie im [Java-Beispiel](#) ¹⁰¹.

5.6.1.1.1 Methoden

5.6.1.1.1.1 addCredentialPropertiesFromMap

Fügt zu den aktuellen Anmeldeinformationen Eigenschaften aus einer Anmeldeinformationen-Eigenschaftszuordnung hinzu. Diese Methode erhält als Argument eine Anmeldeinformationen-Eigenschaftszuordnung (*property_name*, *property_value*). Diese Methode muss nach `beginCredential()` und vor `endCredential()` aufgerufen werden. Alternativ dazu können Sie auch `AddCredentialProperty()` aufrufen.

Signatur

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

Allgemeine Signatur

Parameter

Name	Typ	Beschreibung
arg0	Map	Eine Zuordnung, die den Namen der Anmeldeinformationen-Eigenschaft und ihren Wert bereitstellt.

Beispiele

Im folgenden Codefragment wird gezeigt, wie Sie Anmeldeinformationen namens "mycredential" mit Hilfe der Methode `addCredentialPropertiesFromMap` zum aktuellen Kontext hinzufügen.

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

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

5.6.1.1.1.2 addCredentialProperty

Fügt eine Eigenschaft zu den aktuellen Anmeldeinformationen hinzu (z.B. den Benutzernamen, das Passwort oder beides). Das erste Argument definiert den Namen der Eigenschaft, das zweite Argument den Wert der Eigenschaft. Gültige Eigenschaftsnamen: username, password. In der MapForce Server Advanced Edition wird zusätzlich dazu der Eigenschaftsname oauth:token unterstützt.

Diese Methode muss nach beginCredential() und vor endCredential() aufgerufen werden. Alternativ dazu können Sie auch addCredentialPropertiesFromMap() aufrufen.

Signatur

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

Allgemeine Signatur

Parameter

Name	Typ	Beschreibung
arg0	String	Der Name der Anmeldeinformationen-Eigenschaft (z.B. "username" oder "password").
arg1	String	Der Wert der Anmeldeinformationen-Eigenschaft.

Beispiele

Im folgenden Codefragment wird gezeigt, wie Sie Anmeldeinformationen namens "mycredential" deklarieren. Der Name der Anmeldeinformationen muss derjenige sein, der den Anmeldeinformationen in MapForce zum Zeitpunkt des Designs gegeben wurde.

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

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

5.6.1.1.1.3 *addParameter*

Weist einem im Mapping definierten Parameter einen Wert zu.

Signatur

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

Allgemeine Signatur

Parameter

Name	Typ	Beschreibung
arg0	String	Definiert den Parameternamen.
arg1	String	Definiert den Parameterwert.

5.6.1.1.1.4 *beginCredential*

Erstellt neue Anmeldeinformationen mit dem als Argument bereitgestellten Namen. Wenn Sie diese Methode aufrufen, müssen Sie mit Hilfe von `addCredentialProperty()` oder `addCredentialPropertiesFromMap()` auch Eigenschaften dazu hinzufügen und die Anmeldeinformationen schließlich durch Aufruf von `endCredential()` schließen.

Signatur

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

Allgemeine Signatur

Parameter

Name	Typ	Beschreibung
arg0	String	Der Name der Anmeldeinformationen, wie er in MapForce definiert wurde.

5.6.1.1.1.5 *clearCredentialList*

Löscht die zuvor definierte Liste der Anmeldeinformationen. Alle Anmeldeinformationen sind während der Lebensdauer des Objekts gültig. Rufen Sie diese Methode auf, wenn Sie alle zuvor gesetzten Anmeldeinformationen explizit löschen möchten.

Signatur

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

Allgemeine Signatur

5.6.1.1.1.6 *clearOptions*

Löscht die Liste der zuvor mit der Methode `setOption` definierten Optionen. Alle mit der `setOption`-Methode definierten Optionen sind während der Lebensdauer des Objekts gültig. Rufen Sie diese Methode auf, wenn Sie alle zuvor gesetzten Optionen explizit löschen möchten.

Signatur

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

Allgemeine Signatur

5.6.1.1.1.7 *clearParameterList*

Löscht die Liste der zuvor mit der Methode `addParameter` definierten Parameter.

Signatur

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

Allgemeine Signatur

5.6.1.1.1.8 *endCredential*

Schließt ein zuvor mit der Methode `beginCredential` definiertes Anmeldeinformationen-Objekt.

Signatur

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

Allgemeine Signatur

5.6.1.1.1.9 *getAPIMajorVersion*

Ruft die Hauptversion der MapForce Server API ab. Diese stimmt unter Umständen nicht mit der Produktversion überein, wenn die API mit einem anderen Server verbunden ist.

Signatur

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

Allgemeine Signatur

5.6.1.1.1.10 *getAPIMinorVersion*

Ruft die Nebenversion der MapForce Server API ab. Diese stimmt unter Umständen nicht mit der Produktversion überein, wenn die API mit einem anderen Server verbunden ist.

Signatur

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

Allgemeine Signatur

5.6.1.1.1.11 *getAPIServicePackVersion*

Ruft die Service Pack-Version der MapForce Server API ab. Diese stimmt unter Umständen nicht mit der Produktversion überein, wenn die API mit einem anderen Server verbunden ist.

Signatur

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

Allgemeine Signatur

5.6.1.1.1.12 *getLastExecutionMessage*

Ruft die beim letzten run-Befehl erhaltene Meldung ab.

Signatur

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

Allgemeine Signatur

5.6.1.1.1.13 *getMajorVersion*

Ruft die Hauptversion des Produkts als Anzahl von Jahren seit 1998 ab (z.B. "20" für Altova MapForce Server 2018).

Signatur

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

Allgemeine Signatur

5.6.1.1.1.14 *getMinorVersion*

Ruft die Nebenversion des Produkts ab (z.B. "2" für Altova MapForce Server 2018 r2).

Signatur

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

Allgemeine Signatur

5.6.1.1.1.15 *getOutputParameter*

Ruft den mit dem letzten run-Befehl generierten String Output ab. Gibt Null zurück, wenn kein Output generiert wurde. Für diese Funktion ist ein String-Parameter, der den Namen der Output-Komponente in MapForce definiert, erforderlich.

Signatur

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


Allgemeine Signatur

Parameter

Name	Typ	Beschreibung
arg0	<code>String</code>	Definiert den Namen der Output-Komponente, wie er in MapForce angezeigt wird. Dieser Name wird in der Titelleiste jeder Komponente des Mappings angezeigt (oder wenn Sie mit der rechten Maustaste auf die Komponentenüberschrift klicken und den Befehl Eigenschaften wählen.)

5.6.1.1.1.16 *getProductName*

Ruft den Namen des Produkts ab (z.B. "Altova MapForce Server").

Signatur

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

*Allgemeine Signatur***5.6.1.1.1.17** *getProductNameAndVersion*

Ruft den vollständigen Namen und die Version des Produkts ab (z.B. "Altova MapForce Server 2018 r2 sp1 (x64)").

Signatur

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

Allgemeine Signatur

5.6.1.1.1.18 *getServerPath*

Ruft den Pfad zur ausführbaren Binärdatei des Servers ab.

Signatur

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

Generische Signatur

5.6.1.1.1.19 *getServicePackVersion*

Ruft die Service Pack-Version des Produkts ab bzw. definiert sie (z.B. "1" für Altova MapForce Server 2018 r2 sp1 (x64)).

Signatur

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

Allgemeine Signatur

5.6.1.1.1.20 *getWorkingDirectory*

Ruft das aktuelle Arbeitsverzeichnis ab.

Signatur

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

Generische Signatur

5.6.1.1.1.21 *is64bit*

Gibt **true** zurück, wenn der MapForce Server-Prozessor eine ausführbare 64-Bit-Datei ist.

Signatur

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

Allgemeine Signatur

5.6.1.1.1.22 *run*

Führt eine MapForce Server-Ausführungsdatei (.mfx-Datei) aus. Gibt bei Erfolg **true** zurück, andernfalls **false**.

Signatur

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

Allgemeine Signatur

Parameter

Name	Typ	Beschreibung
arg0	String	Definiert den Pfad zur MapForce Server-Ausführungsdatei (.mfx)-Datei. Wenn Sie einen relativen Pfad definieren, wird dieser anhand des Arbeitsverzeichnisses aufgelöst. Sie können das Arbeitsverzeichnis durch Aufruf der Methode <code>setWorkingDirectory</code> definieren.

5.6.1.1.1.23 *setOption*

Definiert vor Ausführung des Mappings eine Option. Das erste Argument definiert den Namen der Option, das zweite Argument definiert den Optionswert. Diese Methode ist besonders dann nützlich, wenn in einem Mapping globale Ressourcen verwendet werden (siehe [Globale Altova-Ressourcen](#)⁴⁰). Derzeit werden die folgenden Optionen unterstützt:

- **globalresourcefile (oder gr)** - Der Pfad zu einer globalen Ressourcendatei. (Wenn diese Option definiert ist, muss auch ein Name für die Konfiguration der globalen Ressource definiert werden, siehe nächster Eintrag).
- **globalresourcefile (oder gr)** - Name einer globalen Ressourcenkonfiguration. (Wenn diese Option definiert ist, muss ein Dateipfad zu einer globalen Ressourcen-Datei definiert werden, siehe vorheriger Eintrag).
- **catalog** - Pfad zu einer benutzerdefinierten **RootCatalog.xml**-Datei. Mit Hilfe dieser Option können Sie eine benutzerdefinierte Katalogdatei für die Auflösung von im Mapping verwendeten URLs definieren. Der Standardkatalog befindet sich im Unterverzeichnis "etc" des Programminstallationsverzeichnisses.
- **taxonomy-package** - Der Pfad zu einem benutzerdefinierten XBRL-Taxonomiepaket, falls eines für das Mapping erforderlich ist.
- **taxonomy-packages-config-file** - Der Pfad zu einer benutzerdefinierten XBRL-Taxonomiepaketkonfiguration, falls eine für das Mapping erforderlich ist.

Alle definierten Optionen sind während der Lebensdauer des Objekts gültig. Wenn Sie eine Option mit demselben Namen zwei Mal definieren, wird die vorherige Option außer Kraft gesetzt. Um alle Optionen explizit zu löschen, rufen Sie die Methode `clearOptions()` auf.

Signatur

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

Allgemeine Signatur

Parameter

Name	Typ	Beschreibung
arg0	String	Definiert den Namen der zu setzenden Option.
arg1	String	Definiert den Wert der zu setzenden Option.

5.6.1.1.1.24 setServerPath

Definiert den Pfad der ausführbaren MapForce Server-Datei.

Signatur

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

Allgemeine Signatur

Parameter

Name	Typ	Beschreibung
arg0	String	Definiert den Pfad der ausführbaren MapForce Server-Datei.

5.6.1.1.1.25 setWorkingDirectory

Definiert das aktuelle Verzeichnis für ausgeführte Aufträge (relative Pfade werden anhand des Arbeitsverzeichnisses aufgelöst).

Signatur

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

Allgemeine Signatur

Parameter

Name	Typ	Beschreibung
arg0	String	Definiert den Pfad zum Arbeitsverzeichnis.

5.6.1.1.1.26 stop

Beendet den Server-Prozess.

Signatur

```
stop() -> void
```

Generische Signatur

5.6.1.2 MapForceServerException

Die Klasse MapForceServerException bietet programmatischen Zugriff auf von der Klasse MapForceServer ausgelöste Ausnahmen.

```
public class MapForceServerException extends Exception
```

6 Verwaltung digitaler Zertifikate

Die Verwaltung digitaler Zertifikate spielt eine wichtige Rolle beim sicheren Austausch von Daten zwischen einem Client-Rechner und einem Webserver. Da Mappings nicht nur durch MapForce unter Windows ausgeführt werden können, sondern auch durch MapForce Server (entweder als Standalone-Version oder in einer FlowForce Server-Ausführung) auf einem Windows-, Linux- oder macOS-Server, wird in diesem Abschnitt die Verwaltung von HTTPS-Zertifikaten auf verschiedenen Plattformen behandelt.

Im Zusammenhang mit dem sicheren HyperText Transport Protocol (HTTPS) muss zwischen Server- und Client-Zertifikaten unterschieden werden.

Server-Zertifikate

Ein Server-Zertifikat dient dazu, einen Server für eine Client-Applikation wie z.B. MapForce als vertrauenswürdig zu erkennen. Das Server-Zertifikat kann von einer kommerziellen Zertifizierungsstelle oder von Ihrem Unternehmen selbst digital signiert sein. In beiden Fällen können Sie beim Erstellen eines Mappings in Mapforce die folgenden Einstellungen definieren:

- ob das Server-Zertifikat überprüft werden muss,
- ob mit dem Request fortgefahren werden soll, wenn das Namenszertifikat nicht mit dem Namen des Host übereinstimmt.

Diese Einstellungen werden über das MapForce-Dialogfeld "HTTP-Sicherheitseinstellungen" vorgenommen. Wenn Sie die Überprüfung der Server-Zertifikate aktivieren, beachten Sie die folgenden Punkte:

- Wenn Sie einen Webserver aufrufen, dessen Zertifikat von einer vertrauenswürdigen Zertifizierungsstelle signiert ist, erkennt Ihr Betriebssystem das Server-Zertifikat von seiner Konfiguration her meist bereits als vertrauenswürdig, sodass keine zusätzliche Konfiguration erforderlich ist.
- Wenn Sie einen Webserver aufrufen, der ein selbst signiertes Zertifikat bereitstellt (z.B. einen lokalen Netzwerkserver in Ihrem Unternehmen), so müssen Sie auch Ihr Betriebssystem so konfigurieren, dass es dieses Zertifikat als vertrauenswürdig erkennt.

In den meisten Fällen können Sie die Konfiguration der Vertrauenswürdigkeit überprüfen, indem Sie die URL des Webservice in die Adressleiste Ihres Browsers eingeben. Wenn dem Server nicht vertraut wird oder wenn Ihr Betriebssystem den Server laut Konfiguration nicht als vertrauenswürdig einstuft, so wird in Ihrem Browser eine Meldung ungefähr folgenden Inhalts angezeigt "Diese Verbindung ist nicht vertrauenswürdig" oder "Es gibt ein Problem mit dem Zertifikat der Website". Beachten Sie, dass Sie die Einstellung für die Vertrauenswürdigkeit eines Webserver nicht überprüfen können, wenn der Browser eine andere Zertifikat-Datenbank als die des Betriebssystems verwendet (z.B. Firefox auf Ubuntu).

Unter Windows können Sie die Vertrauenswürdigkeit einer Verbindung durch Befolgen der Anweisungen im Browser und Importieren oder Installieren des benötigten Zertifikats im Speicher der vertrauenswürdigen Stammzertifikate bestätigen (siehe [Vertrauenswürdige Server-Zertifikate unter Windows](#)¹⁴⁷). Auf macOS-Systemen können Sie die entsprechende Operation in Keychain Access durchführen (siehe [Vertrauenswürdige Server-Zertifikate auf macOS](#)¹⁴⁶). Eine Anleitung zu Linux-Systemen finden Sie unter [Vertrauenswürdige Server-Zertifikate unter Linux](#)¹⁴⁵.

Client-Zertifikate

Während Server-Zertifikate dazu dienen, einen Server als vertrauenswürdig zu erkennen, werden Client-Zertifikate in erste Linie dazu verwendet, den aufrufenden Client auf dem Webserver zu identifizieren. Wenn Sie

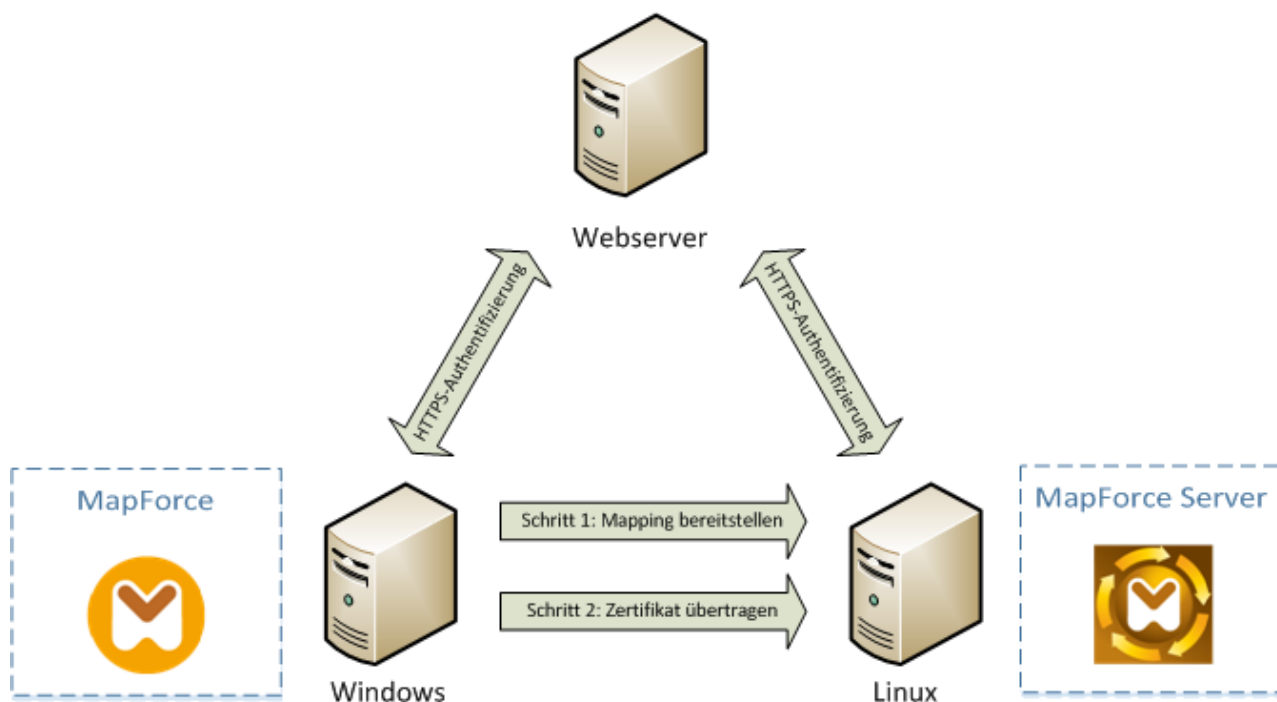
vorhaben, auf einen Webserver zuzugreifen, für den Client-Zertifikate erforderlich sind, müssen Sie möglicherweise den Administrator des Webserver bezüglich einer Client-Konfigurationsanweisung kontaktieren. So kann ein Webserver z.B. bei IIS (Internet Information Services) so konfiguriert werden, dass er HTTPS- und Client-Zertifikate auf eine der folgenden Arten behandelt:

- HTTPS ist erforderlich und Client-Zertifikat wird ignoriert
- HTTPS ist erforderlich und Client-Zertifikat wird akzeptiert
- HTTPS und Client-Zertifikat sind erforderlich

Der Erfolg eines Webservice-Request hängt von der Konfiguration des Webserver und der Client-Applikation ab. Wenn der Webserver z.B. so konfiguriert ist, dass er ein Client-Zertifikat verlangt, so muss die aufrufende Applikation ein gültiges Client-Zertifikat zur Verfügung stellen, damit der Aufruf erfolgreich ist.

Aus der MapForce-Perspektive gesehen, gilt dasselbe für Mappings, die Webservice-Aufrufe über HTTPS enthalten. Damit solche Mappings erfolgreich ausgeführt werden können, geht MapForce davon aus, dass der Webserver so konfiguriert ist, dass er das Client-Zertifikat akzeptiert oder zwingend verlangt und dass das Betriebssystem, auf dem das Mapping ausgeführt wird, das richtige Client-Zertifikat für den Webserver zur Verfügung stellt.

Im unten gezeigten Diagramm sehen Sie ein Szenario, in dem ein in MapForce verwendetes Client-Zertifikat auf einen Linux-Server transferiert wird, auf dem MapForce Server ausgeführt wird. Sobald das Zertifikat auf das Zielbetriebssystem transferiert wurde, kann MapForce Server sich mit Hilfe dieses Zertifikats beim Webserver authentifizieren und das Mapping erfolgreich ausführen.



Bereitstellen von Mappings mit Client-Zertifikaten auf einem anderen Rechner

Bei einer HTTPS-Authentifizierung in Webservice-Aufrufen kann MapForce zusätzlich zu HTTP die Transport Layer Security (TLS) verwenden. TLS ist der Nachfolger des SSL (Secure Sockets Layer)-Protokolls. Beachten Sie dass der Client bzw. Server auf SSL zurückgreift, falls die Client- bzw. Server-Implementierung

TLS nicht unterstützt.

Die jeweilige Implementierung der Zertifikatverwaltung auf den einzelnen Plattformen ist dafür verantwortlich, dass Webservice-Aufrufe mit Client-Authentifizierung durch MapForce unterstützt werden. Daher muss die Zertifikatverwaltung immer im Rahmen des zugrunde liegenden Betriebssystems erfolgen. Jedes Betriebssystem bietet unterschiedliche Unterstützung für Zertifikatverwaltung - siehe Tabelle unten.

Plattform	Zertifikatverwaltung und -implementierung
Windows	Unter Windows können Sie Zertifikate mit Hilfe des Zertifikat-Snap-in verwalten (siehe Aufrufen der Zertifikatspeicher unter Windows¹⁴⁸). TLS-Unterstützung steht über den <i>Sicheren Kanal</i> (auch als SKanal bezeichnet) zur Verfügung.
Linux	Unter Linux können Sie Zertifikate mit Hilfe des OpenSSL (openssl) Befehlszeilentools und der dazugehörigen Bibliothek verwalten. Falls auf dem Linux-Rechner, auf dem MapForce Server installiert ist, noch keine OpenSSL-Unterstützung zur Verfügung steht, müssen Sie diese Funktion herunterladen und sie installieren, bevor Sie Zertifikate verwalten können. TLS-Unterstützung steht über die OpenSSL-Bibliothek zur Verfügung (https://www.openssl.org/).
macOS	Auf macOS-Rechnern können Sie Zertifikate mit dem <i>Keychain Access Manager</i> unter Finder > Applications > Utilities verwalten. TLS-Unterstützung steht über die native <i>Secure Transport</i>-Bibliothek zur Verfügung

Wenn Sie das Mapping auf einem Windows-Betriebssystem ausführen, auf dem Sie den Webservice, den Sie aufrufen möchten, bereits erfolgreich verwendet haben, ist normalerweise keine zusätzliche Zertifikatkonfiguration erforderlich (Welche Bedingungen erfüllt werden müssen, damit das Mapping erfolgreich unter Windows ausgeführt werden kann, finden Sie unter [Client-Zertifikate unter Windows](#)¹⁵⁹). Wenn Sie allerdings auf einem Windows-Rechner Mappings mit MapForce erstellen und diese dann auf einem anderen Computer bereitstellen (der möglicherweise unter einem anderen Betriebssystem läuft), so wird das Client-Zertifikat nicht zusammen mit dem bereitgestellten Paket gespeichert bzw. kopiert. Damit der Webservice-Aufrufe (und das Mapping) erfolgreich ausgeführt werden können, muss das Client-Zertifikat auch auf dem Zielbetriebssystem vorhanden sein.

Um ein Zertifikat von einem Windows-System auf einen anderen Windows-basierten Rechner zu transferieren, exportieren Sie das erforderliche Zertifikat (mit dem privaten Schlüssel) aus dem Ausgangssystem (siehe [Exportieren von Zertifikaten aus Windows](#)¹⁴⁹). Importieren Sie anschließend dasselbe Zertifikat in den Speicher **Aktueller Benutzer\Persönlich** auf dem Zielbetriebssystem (siehe [Client-Zertifikate unter Windows](#)¹⁵⁹).

Eine Anleitung zum Transferieren von Client-Zertifikaten auf die Linux- und macOS-Plattform finden Sie unter [Client-Zertifikate unter Linux](#)¹⁵⁵ bzw. [Client-Zertifikate unter macOS](#)¹⁵⁷.

6.1 Definieren vertrauenswürdiger Zertifikate unter Linux

Auf Linux-Systemen gehen Sie vor, wie unten gezeigt, um ein vertrauenswürdiges Zertifikat in den Zertifikatspeicher des Systems zu importieren: Führen Sie die folgenden Schritte nur durch, wenn Sie sicher sind, dass das betreffende Zertifikat authentisch ist.

Befolgen Sie auf Debian und Ubuntu die folgende Anleitung:

1. Kopieren Sie die Zertifikatdatei des Webserver in das folgende Verzeichnis.

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

2. Aktualisieren Sie den Zertifikatspeicher folgendermaßen:

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

Befolgen Sie auf CentOS die folgende Anleitung:

1. Installieren Sie das Paket ca-certificates:

```
yum install ca-certificates
```

2. Aktivieren Sie die Funktion zur dynamischen Konfiguration der Zertifizierungsstelle:

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

3. Kopieren Sie das Server-Zertifikat in das folgenden Verzeichnis:

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

4. Verwenden Sie den folgenden Befehl:

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

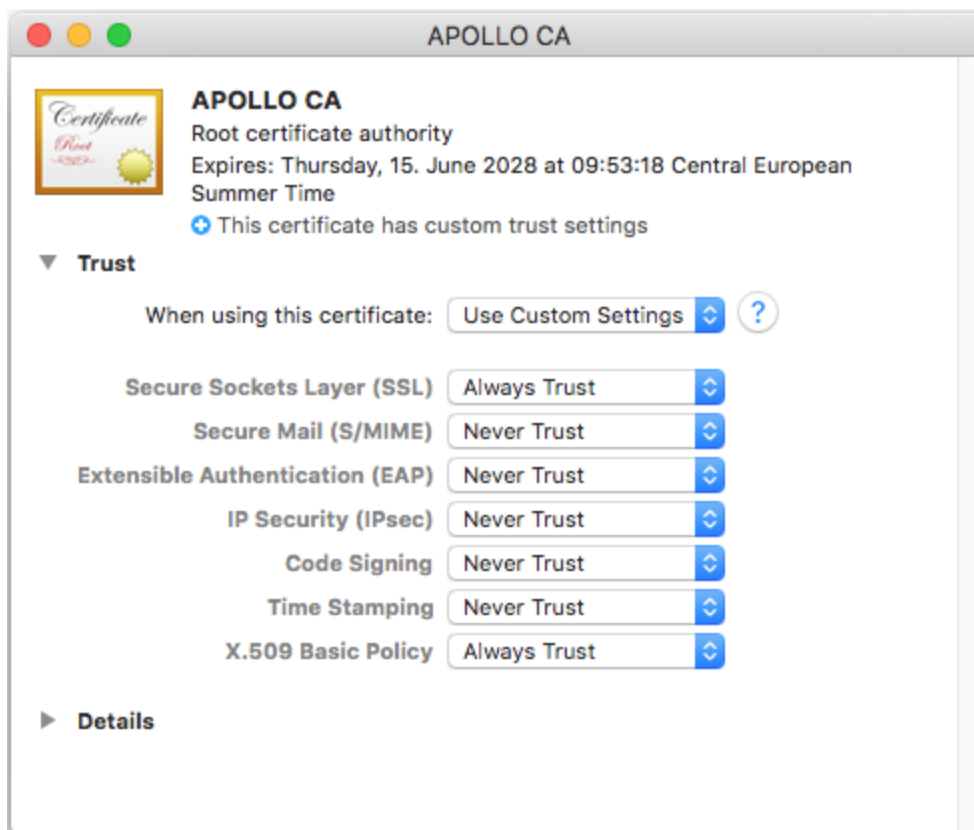
In Fällen, in denen Sie nur über den Browser auf den Server zugreifen, genügt es, das Zertifikat in den Zertifikatspeicher des Browsers zu importieren. Die genaue Vorgehensweise variiert je nach Browser. In Firefox 59.0.2 können Sie z.B. folgendermaßen vorgehen:

1. Klicken Sie unter **Options | Privacy & Security** auf **View Certificates**.
2. Klicken Sie auf dem Register **Authorities** auf **Import** und navigieren Sie zur zuvor erstellen Root-Zertifikatdatei.
3. Aktivieren Sie die Option **Trust this CA to identify websites**, wenn Sie gefragt werden und klicken Sie auf **OK**.

6.2 Definieren vertrauenswürdiger Zertifikate auf macOS-Systemen

Auf macOS-Systemen importieren Sie ein vertrauenswürdiges Zertifikat auf folgende Weise in den Keychain Access:

1. Starten Sie Keychain Access.
2. Klicken Sie auf **System** und anschließend auf **Certificates**.
3. Klicken Sie im Menü **File** auf **Import Items**.
4. Navigieren Sie zum gewünschten vertrauenswürdigen Zertifikat und klicken Sie auf **Open**.
5. Geben Sie das Keychain Access-Passwort ein, wenn Sie dazu aufgefordert werden und klicken Sie anschließend auf **Modify Keychain**.
6. Doppelklicken Sie auf das Zertifikat, erweitern Sie den Abschnitt *Trust* und aktivieren Sie die Option **Always Trust**.



6.3 Definieren vertrauenswürdiger Zertifikate unter Windows

Auf Windows-Systemen importieren Sie ein vertrauenswürdiges Zertifikat auf folgende Weise in den Zertifikatspeicher des Systems:

1. Öffnen Sie den Windows-Zertifikatspeicher für *das Computer-Konto*, siehe [Aufrufen des Windows-Zertifikatspeichers](#)¹⁴⁸.
2. Erweitern Sie den Ordner *Vertrauenswürdige Stammzertifizierungsstellen der Zertifikate (Lokaler Computer)*-Struktur, klicken Sie mit der rechten Maustaste auf **Zertifikate**, wählen Sie **Alle Aufgaben | Importieren** und befolgen Sie die Anweisungen des Zertifikatimportassistenten.

Nähere Informationen dazu finden Sie auf der Microsoft Website im Artikel [Import a Certificate](#).

6.4 Aufrufen der Zertifikatspeicher unter Windows

Auf Windows-Systemen können Sie Zertifikate über das Microsoft Management Console (MMC) Snap-in entweder für Ihr Benutzerkonto oder für das Computerkonto verwalten.

Um das Zertifikat-Snap-in für den *aktuellen Windows-Benutzer* zu öffnen, führen Sie über die Befehlszeile den folgenden Befehl aus:

```
certmgr.msc
```

Um das Zertifikat-Snap-in für das *Computerkonto* zu öffnen, gehen Sie folgendermaßen vor:

1. Führen Sie in der Befehlszeile den Befehl `mmc` aus.
2. Klicken Sie im Menü **Datei** der MMC auf **Snap-in hinzufügen/entfernen**.
3. Klicken Sie auf **Zertifikate** und anschließend auf **Hinzufügen**.
4. Wählen Sie **Computerkonto** aus und klicken Sie anschließend auf **Weiter**.
5. Wählen Sie **Lokalen Computer** und anschließend **Fertig stellen**.

6.5 Exportieren von Zertifikaten aus Windows

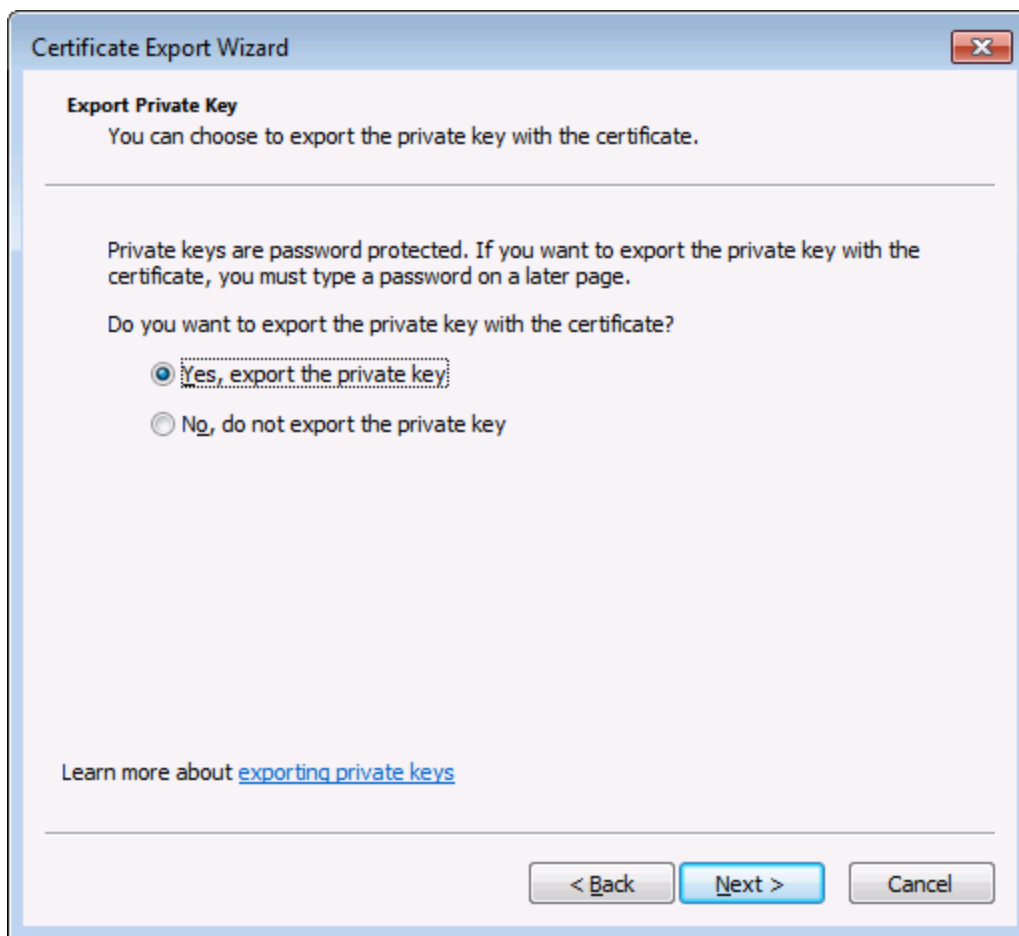
Bei Mappings, in denen Webservices über HTTPS aufgerufen werden und die auf einem macOS- oder Linux-Server, auf dem MapForce Server oder FlowForce Server ausgeführt wird, bereitgestellt werden, muss auf dem Nicht-Windows-Betriebssystem dasselbe Client-Zertifikat zur Verfügung stehen, das auch beim Design und Testen des Mappings unter Windows verwendet wurde. Um solche Mappings auf einem anderen Betriebssystem als Windows mit MapForce Server ausführen zu können, exportieren Sie das erforderliche Zertifikat mit dem privaten Schlüssel aus Windows und importieren Sie es anschließend in das Zielbetriebssystem.

So exportieren Sie ein Zertifikat mit einem privaten Schlüssel aus Windows:

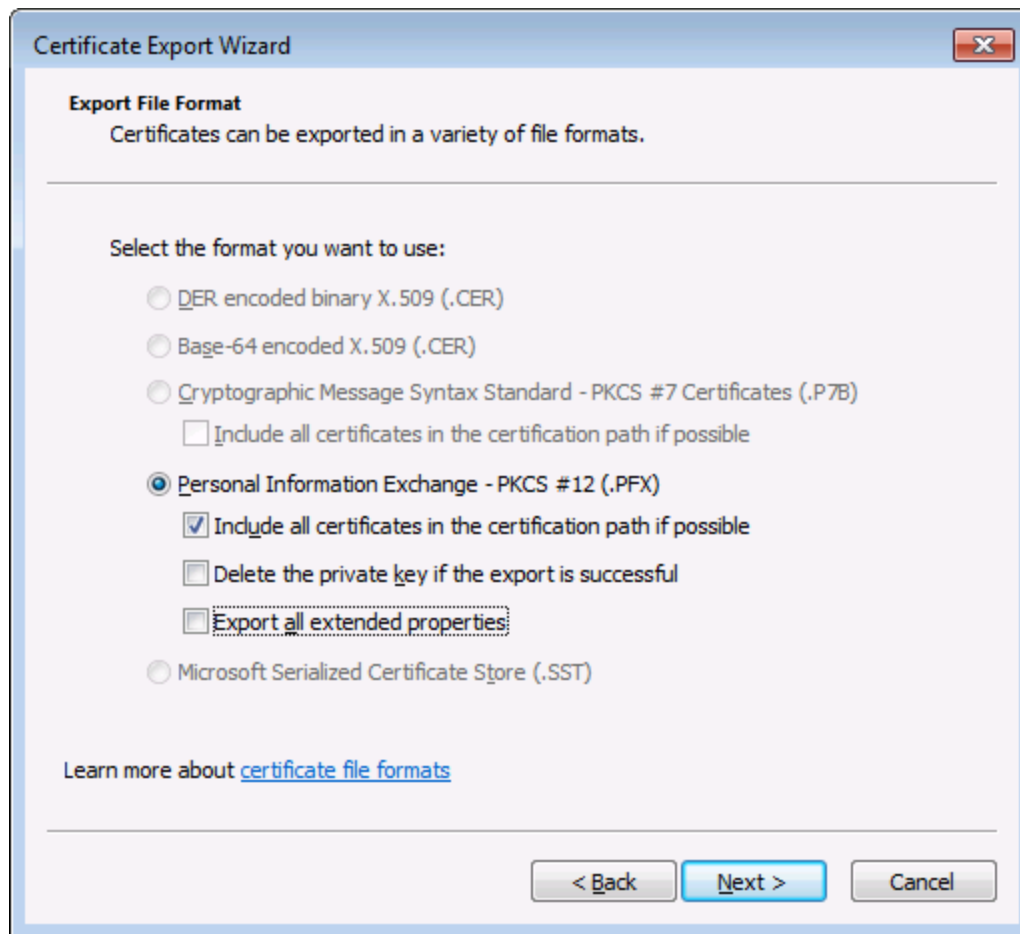
1. Öffnen Sie in Windows das Zertifikate-Snap-in (siehe [Aufrufen der Zertifikatspeicher unter Windows](#)¹⁴⁸).
2. Klicken Sie mit der rechten Maustaste auf das gewünschte Zertifikat, platzieren Sie den Mauszeiger über **Alle Aufgaben** und klicken Sie auf **Exportieren**.
3. Klicken Sie auf **Weiter**.



4. Wählen Sie die Option, dass das Zertifikat zusammen mit seinem privaten Schlüssel aus Windows exportiert werden soll und klicken Sie anschließend auf **Weiter**.

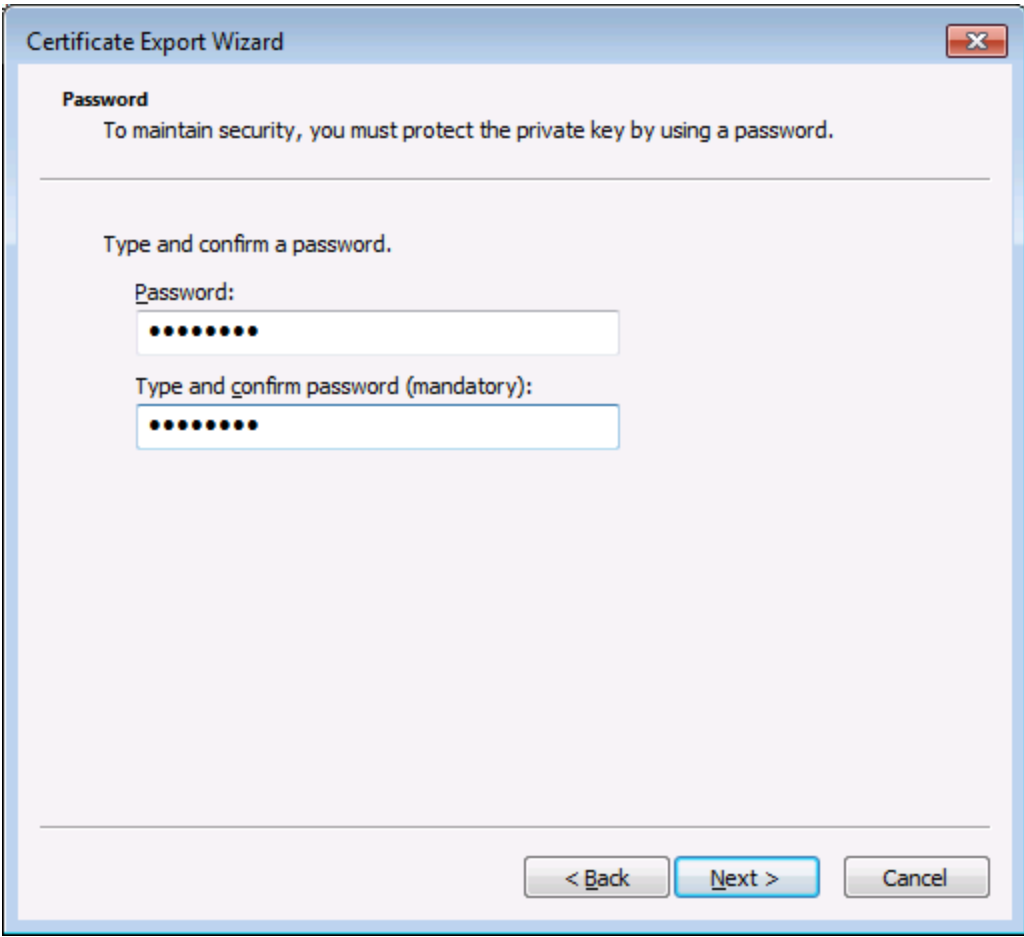


5. Wählen Sie das Dateiformat *Privater Informationsaustausch - PKCS #12 (.pfx)* aus und klicken Sie auf **Weiter**.



Anmerkung: Stellen Sie sicher, dass Sie nicht die Option **Privaten Schlüssel nach erfolgreichem Export löschen** aktiviert haben, da Sie das Zertifikat nach dem Export sonst nicht mehr verwenden können.

6. Geben Sie ein Passwort ein und klicken Sie auf **Weiter**. Sie benötigen dieses Passwort nachdem Sie das Zertifikat in das Zielbetriebssystem kopiert haben.



The image shows a Windows dialog box titled "Certificate Export Wizard". It has a standard Windows window border with a close button (X) in the top right corner. The main content area has a light blue background. At the top, the word "Password" is written in bold. Below it, a message states: "To maintain security, you must protect the private key by using a password." A horizontal line separates this message from the input fields. Below the line, the text "Type and confirm a password." is displayed. There are two input fields: the first is labeled "Password:" and the second is labeled "Type and confirm password (mandatory):". Both fields contain ten black dots, indicating that the password is masked. At the bottom of the dialog, there are three buttons: "< Back", "Next >" (which is highlighted with a blue border), and "Cancel".

Password

To maintain security, you must protect the private key by using a password.

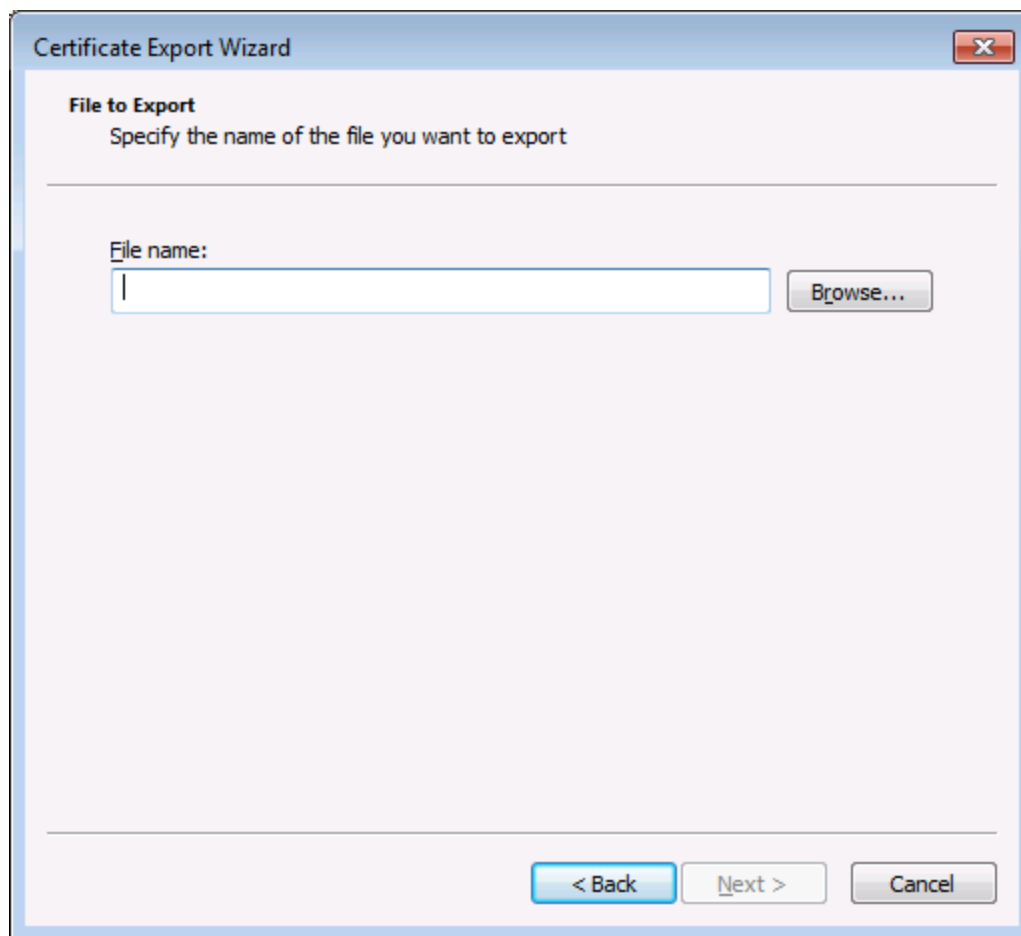
Type and confirm a password.

Password:

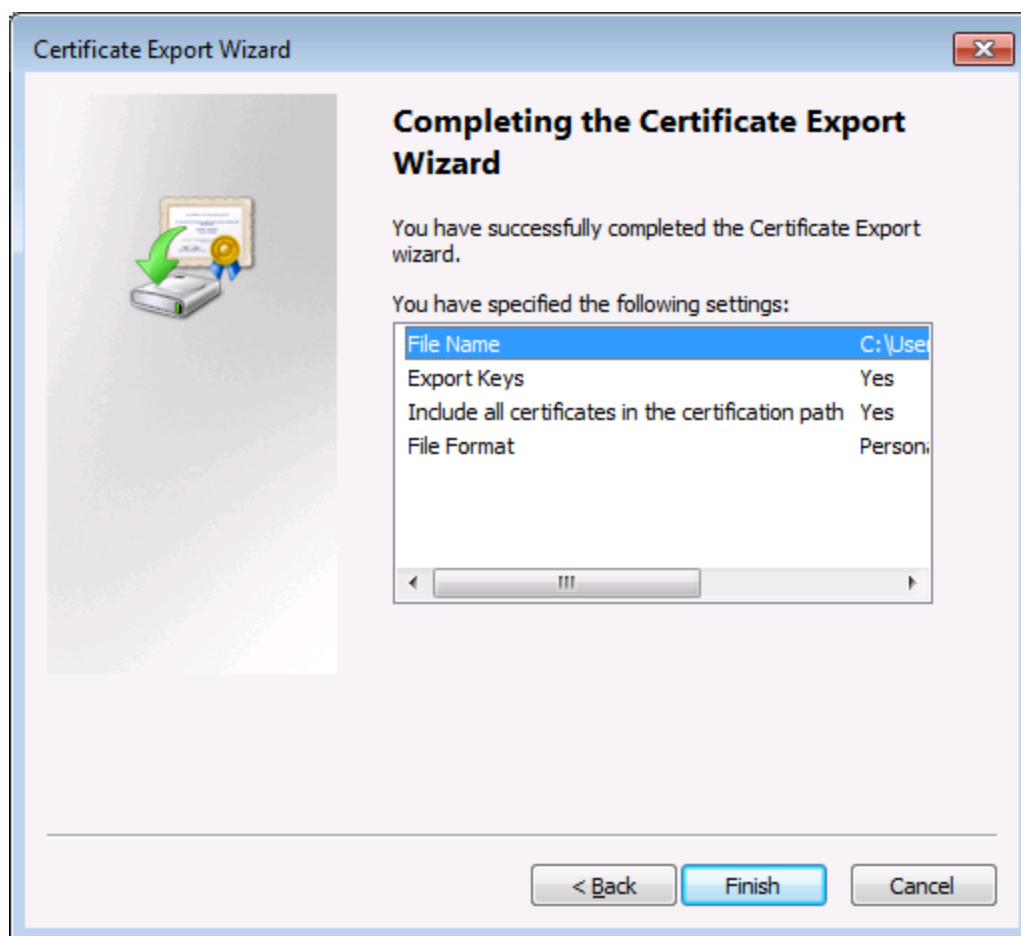
Type and confirm password (mandatory):

< Back Next > Cancel

7. Navigieren Sie zur Datei, die Sie exportieren möchten und klicken Sie auf **Weiter**.



8. Klicken Sie auf **fertig stellen**.



6.6 Client-Zertifikate unter Linux

Wenn Ihre Mappings eine Webservice-Authentifizierung mittels Client-Zertifikaten über HTTPS enthalten, befolgen Sie die folgende Anleitung, um solche Mappings auf einem Linux-Rechner, auf dem MapForce Server läuft, bereitzustellen:

1. Öffnen Sie das Mapping, das den Webservice aufruft, in MapForce.
2. Doppelklicken Sie auf die Überschrift der Webservice-Komponente. Daraufhin wird das Dialogfeld "Komponenteneinstellungen" aufgerufen.
3. Klicken Sie neben "HTTP-Sicherheitseinstellungen" auf **Bearbeiten**.
4. Klicken Sie im Dialogfeld "HTTP-Sicherheitseinstellungen" auf **Client-Zertifikat** und wählen Sie anschließend in Windows das gewünschte Zertifikat aus dem Speicher **Aktueller Benutzer\Persönlich** aus .
5. Speichern Sie das Mapping und kompilieren Sie es zu einer Mapping-Ausführungsdatei oder stellen Sie es auf FlowForce bereit.
6. Übertragen Sie die für den Webservice-Aufruf erforderlichen Client-Zertifikate auf das Zielbetriebssystem. Stellen Sie sicher, dass das Zertifikat einen privaten Schlüssel hat und dass die Eigenschaft **Erweiterte Schlüsselverwendung** des Zertifikats "Clientauthentifizierung" als Zweck enthält.

So transferieren Sie das Client-Zertifikat auf Linux:

1. Exportieren Sie das Client-Zertifikat mit dem privaten Schlüssel im Dateiformat *Privater Informationsaustausch - PKCS #12 (.pfx)* (siehe [Exportieren von Zertifikaten aus Windows](#)¹⁴⁹).
2. Kopieren Sie die Zertifikatdatei auf den Linux-Rechner.
3. Konvertieren Sie die .pfx-Datei mit dem folgenden Befehl in das .pem-Format.

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

Dieser Befehl parst die .pfx-Datei und erzeugt eine .pem-Datei, ohne dabei den privaten Schlüssel zu verschlüsseln. Bei Zertifikaten mit einem verschlüsselten privaten Schlüssel wird nach einem Passwort gefragt, was bei der Server-Ausführung nicht unterstützt wird.

Ausführen des Mappings

Damit MapForce Server die .pem-Datei als Client-Zertifikat verwendet, definieren Sie beim Ausführen des Mappings den Parameter `--certificatespath`. Der Parameter `--certificatespath` definiert den Pfad zu dem Verzeichnis, in dem alle vom aktuellen Mapping benötigten Zertifikate gespeichert sind. Wenn der Pfad zur Zertifikatdatei z.B. `/home/John/John Doe.pem` lautet, so muss `--certificatespath` auf `/home/John` gesetzt werden.

Wenn der Parameter `--certificatespath` nicht angegeben wird, sucht MapForce Server standardmäßig im Verzeichnis `$HOME/.config/altova/certificates` des aktuellen Benutzers nach Zertifikaten.

Damit das Mapping erfolgreich ausgeführt werden kann, muss die Zertifikatdatei die Erweiterung `.pem` haben und der Name der Datei muss mit dem Common Name (CN) des Zertifikats einschließlich der Leerzeichen übereinstimmen (z.B. **John Doe.pem**).

Wenn Sie vorhaben, das Mapping als FlowForce Server-Auftrag auszuführen, kopieren Sie das Zertifikat in das

Verzeichnis `$HOME/.config/altova/certificates`. FlowForce Server sucht bei Ausführung des Auftrags in diesem Verzeichnis nach allen für das Mapping erforderlichen Zertifikatdateien.

Stellen Sie aus Sicherheitsgründen sicher, dass Zertifikatdateien nicht von anderen Benutzern gelesen werden können, da sie sensible Informationen enthalten.

6.7 Client-Zertifikate auf macOS

Wenn Ihre Mappings eine Webservice-Authentifizierung mittels Client-Zertifikaten über HTTPS enthalten, befolgen Sie die folgende Anleitung, um solche Mappings auf einem macOS-System, auf dem MapForce Server läuft, bereitzustellen:

1. Öffnen Sie das Mapping, das den Webservice aufruft, in MapForce.
2. Doppelklicken Sie auf die Überschrift der Webservice-Komponente. Daraufhin wird das Dialogfeld "Komponenteneinstellungen" aufgerufen.
3. Klicken Sie neben "HTTP-Sicherheitseinstellungen" auf **Bearbeiten**.
4. Klicken Sie im Dialogfeld "HTTP-Sicherheitseinstellungen" auf **Client-Zertifikat** und wählen Sie anschließend das gewünschte Zertifikat aus .
5. Wenn der Zertifikatname nicht genau mit dem Host-Namen des Servers übereinstimmt, aktivieren Sie die Option **Diskrepanz zwischen Host-Namen im Zertifikat und Request zulassen**.
6. Speichern Sie das Mapping und stellen Sie es auf dem Zielbetriebssystem bereit.
7. Übertragen Sie die für den Webservice-Aufruf erforderlichen Client-Zertifikate auf das Zielbetriebssystem. Stellen Sie sicher, dass das Zertifikat einen privaten Schlüssel hat und dass die Eigenschaft **Erweiterte Schlüsselverwendung** des Zertifikats "Clientauthentifizierung" als Zweck enthält.

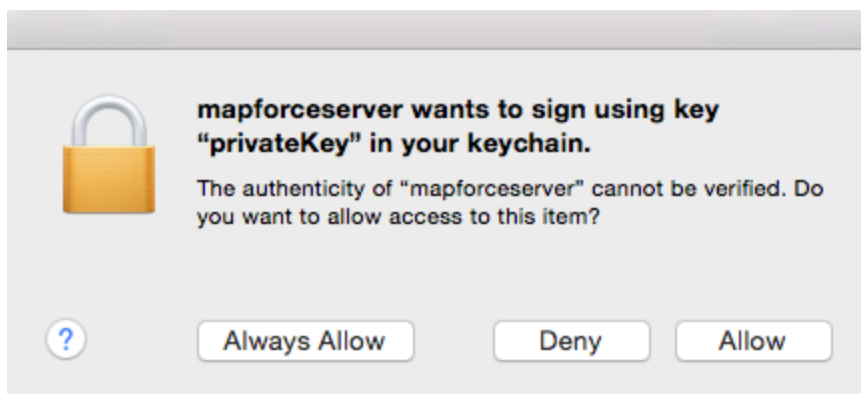
So übertragen Sie das Client-Zertifikat auf das macOS-System:

1. Exportieren Sie das Client-Zertifikat mit dem privaten Schlüssel im Dateiformat *Privater Informationsaustausch - PKCS #12 (.pfx)* (siehe [Exportieren von Zertifikaten aus Windows](#)¹⁴⁹) aus Windows und kopieren Sie die .pfx-Datei auf den macOS.
2. Stellen Sie sicher, dass das Betriebssystem das Server-Zertifikat als vertrauenswürdig einstuft (siehe [Definieren vertrauenswürdiger Zertifikate auf Mac-Systemen](#)¹⁴⁶).
3. Führen Sie über **Finder > Applications > Utilities** Keychain Access aus.
4. Klicken Sie im Menü **File** auf **Import Items**.
5. Navigieren Sie zu dem in Schritt 1 exportierten Client-Zertifikat und wählen Sie eine Ziel-Keychain aus.
6. Klicken Sie auf **Open** und geben Sie das Passwort ein, mit dem das Zertifikat verschlüsselt wurde.

Ausführen des Mappings

Sie können jetzt das Mapping mit dem MapForce Server-Befehl run ausführen. Beachten Sie dazu Folgendes:

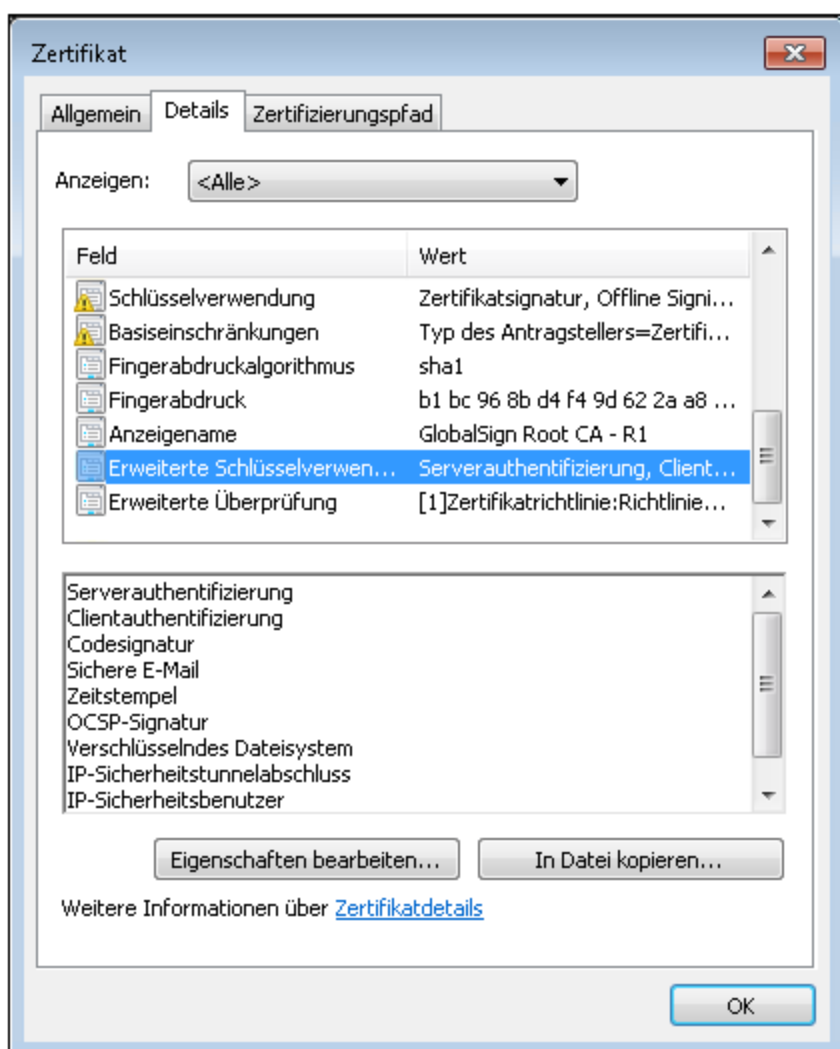
- Wenn Sie das Mapping über einen entfernten Rechner mittels SSH ausführen, entsperren Sie zuerst die Keychain mit dem Befehl `security unlock-keychain`.
- Wenn Sie das Mapping über die grafische Benutzeroberfläche von macOS ausführen, klicken Sie auf **Allow**, wenn Sie gefragt werden, ob MapForce Server Zugriff auf die Keychain erhalten soll.



6.8 Client-Zertifikate unter Windows

Wenn Sie unter Windows ein Mapping ausführen, für das Client-Zertifikate erforderlich sind, müssen die folgenden Voraussetzungen erfüllt werden, damit das Mapping erfolgreich ausgeführt werden kann:

- Das Client-Zertifikat muss im Zertifikatspeicher **Aktueller Benutzer\Persönlich** (auch als **Mein Speicher** bezeichnet) vorhanden sein. Dazu muss es mit dem Zertifikatimport-Assistenten importiert werden. Eine Anleitung dazu finden Sie unter [https://technet.microsoft.com/de-de/library/cc754489\(v=ws.11\).aspx](https://technet.microsoft.com/de-de/library/cc754489(v=ws.11).aspx).
- Das Zertifikat muss einen privaten Schlüssel haben.
- Die Eigenschaft **Erweiterte Schlüsselverwendung** des Zertifikats muss als Zweck "Clientauthentifizierung" enthalten.



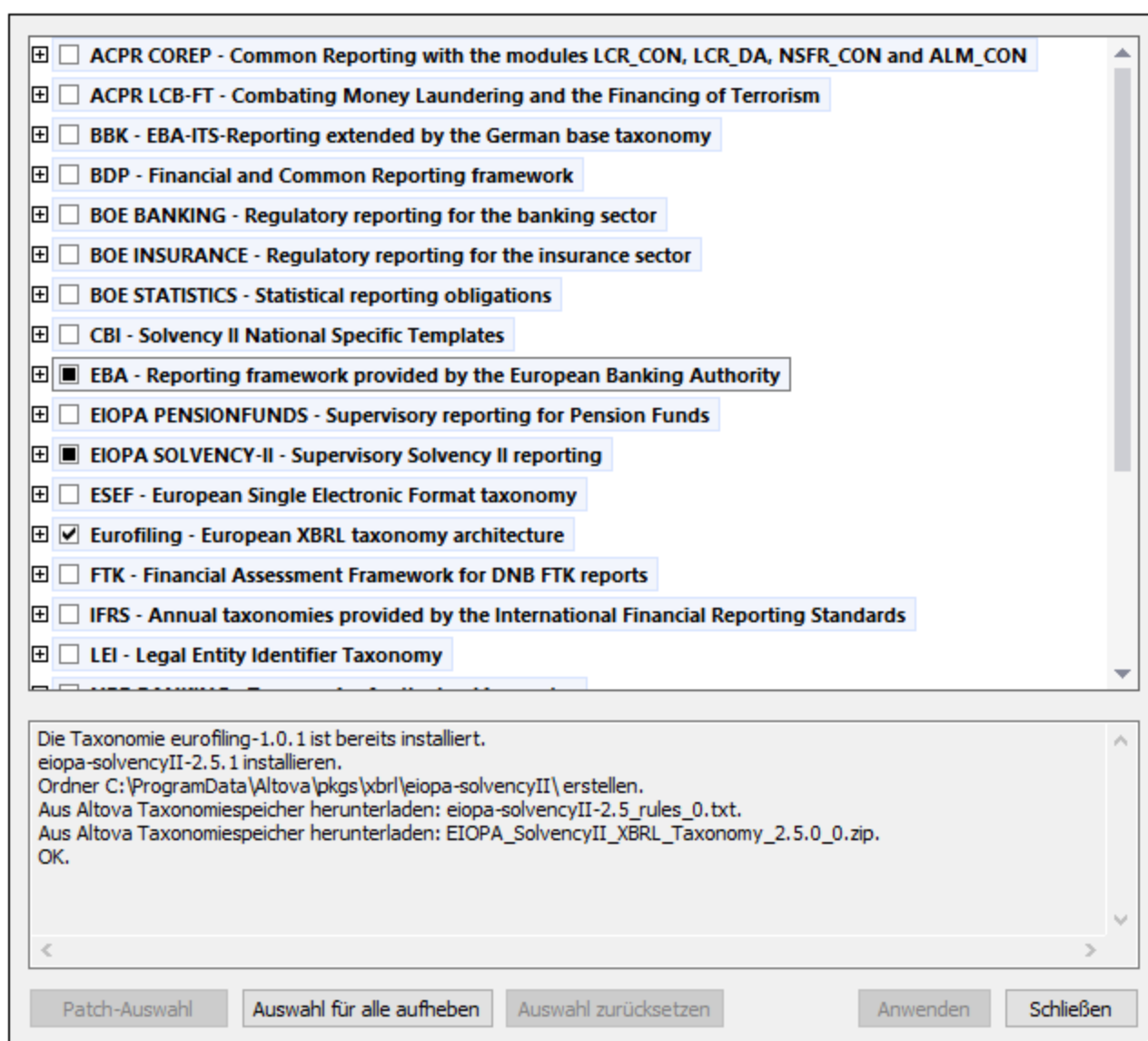
In der aktuellen Version von MapForce wählt Windows das erforderliche Zertifikat aufgrund einer Einschränkung der von MapForce verwendeten Bibliothek automatisch aus dem Zertifikatspeicher aus, wenn Sie das Mapping ausführen. Das Mapping wird erfolgreich ausgeführt, wenn der Server nach dem Filtern des Zertifikatspeichers **Aktueller Benutzer\Persönlich** ein passendes Zertifikat findet. Beachten Sie, dass die HTTPS-Authentifizierung (und die Auswahl des Zertifikats) von Windows und nicht von MapForce oder MapForce Server

gesteuert wird. Es kann vorkommen, dass im Zertifikatspeicher **Aktueller Benutzer\Persönlich** mehrere Zertifikate vorhanden sind und dass das Betriebssystem automatisch ein unpassendes Zertifikat auswählt, wodurch das Mapping fehlschlägt. Diese Situation lässt sich vermeiden, indem Sie die Anzahl der im Speicher **Aktueller Benutzer\Persönlich** verfügbaren Zertifikate einschränken.

7 Taxonomie-Manager

Der XBRL-Taxonomie-Manager ist ein Altova-Tool, mit dem Sie XBRL-Taxonomien zentral installieren und verwalten können, um diese in allen XBRL-fähigen Applikationen von Altova einschließlich MapForce Server verwenden zu können.

- Unter Windows hat der Taxonomie-Manager eine grafische Benutzeroberfläche (siehe Abbildung unten) und steht auch über die Befehlszeile zur Verfügung. (Die Desktop-Applikationen von Altova stehen nur unter Windows zur Verfügung; siehe Liste unten).
- Unter Linux und MacOS steht der Taxonomie-Manager nur über die Befehlszeile zur Verfügung. (Die Server-Applikationen von Altova stehen unter Windows, Linux und macOS zur Verfügung; siehe Liste unten).



XBRL-fähige Applikationen von Altova

Desktop-Applikationen (nur Windows)	Server-Applikationen (Windows, Linux, macOS)
Altova XBRL Add-ins für Excel (EBA, ESEF, Solvency II, WIP)	MapForce Server (Standard und Advanced Edition)
MapForce Enterprise Edition	RaptorXML+XBRL Server
StyleVision Enterprise Edition	StyleVision Server
XMLSpy Enterprise Edition	

Installation und Deinstallation des Taxonomie-Managers

Der Taxonomie-Manager wird bei der ersten Installation einer neuen Version der Altova Mission Kit Enterprise Edition oder einer der XBRL-fähigen Applikationen von Altova (*siehe Tabelle oben*) automatisch installiert.

Ebenso wird er auch automatisch entfernt, wenn Sie die letzte XBRL-fähige Applikation von Altova auf Ihrem Rechner deinstallieren.

Taxonomie-Manager-Funktionalitäten

Im Taxonomie-Manager stehen die folgenden Funktionalitäten zur Verfügung:

- Anzeigen der auf Ihrem Rechner installierten XBRL-Taxonomien und Überprüfung, ob neue Versionen zum Download zur Verfügung stehen.
- Download neuer Versionen von XBRL-Taxonomien unabhängig vom Altova Produkt-Release-Zyklus. (Die Taxonomien werden von Altova online bereitgestellt und können über den Taxonomie-Manager heruntergeladen werden).
- Installation oder Deinstallation jeder beliebigen (oder ggf. aller) der zahlreichen Versionen einer bestimmten Taxonomie.
- Eine XBRL-Taxonomie kann Abhängigkeiten von anderen Taxonomien aufweisen. Bei der Installation oder Deinstallation einer bestimmten Taxonomie informiert Sie der Taxonomie-Manager über davon abhängige Taxonomien und installiert bzw. entfernt diese ebenfalls automatisch.
- Der Taxonomie-Manager ordnet Schema-Referenzen mit Hilfe des [XML-Katalogs](#) lokalen Dateien zu. Dadurch lassen sich große XBRL-Taxonomien schneller verarbeiten, als wenn sie sich unter einem entfernten Pfad befinden.
- Alle wichtigen Taxonomien werden über den Taxonomie-Manager bereitgestellt und regelmäßig auf die jeweils neuesten Version aktualisiert. Dadurch können alle Ihre Taxonomien zentral verwaltet werden und stehen allen XBRL-fähigen Applikationen von Altova jederzeit zur Verfügung.
- Im Taxonomie-Manager vorgenommene Änderungen werden für alle auf dem Rechner installierten Altova-Produkte wirksam.

Benutzerdefinierte XBRL-Taxonomien

Wenn Sie mit benutzerdefinierten XBRL-Taxonomien, die nicht Teil des Taxonomie-Manager sind, arbeiten müssen, können Sie diese zur Gruppe der benutzerdefinierten Pakete, die MapForce Server referenzieren kann, hinzufügen. Gehen Sie dazu folgendermaßen vor:

- *In Altova-Desktop-Applikationen:* Wählen Sie den Menübefehl **Extras | Optionen** und gehen Sie anschließend zum Abschnitt *XBRL | Taxonomiepakete*. Navigieren Sie zum ZIP-Paket Ihrer

benutzerdefinierten XBRL-Taxonomie. Nähere Informationen dazu finden Sie in Ihrer Desktop-Produktdokumentation unter der Beschreibung dieses Befehls.

- *In Altova-Server-Applikationen:* Geben Sie bei der Befehlszeilenausführung von Befehlen, die benutzerdefinierte Taxonomien unterstützen, die Option **--taxonomy-package** oder **--taxonomy-package-config-file** ein. Diese Optionen werden z.B. in RaptorXML+XBRL Server von XBRL-Validierungsbefehlen wie `valxbrl` oder `valxbrltaxonomy` oder in MapForce vom Befehl `run` unterstützt.

Funktionsweise

Alle in Altova-Produkten verwendeten XBRL-Taxonomien werden von Altova online bereitgestellt. Dieser Speicher wird bei Veröffentlichung neuer Versionen der Taxonomien aktualisiert. Im Taxonomie-Manager werden sowohl bei Aufruf über die Benutzeroberfläche als auch über das CLI Informationen über die neuesten verfügbare Taxonomien angezeigt. Sie können die gewünschten Taxonomien dann über den Taxonomie-Manager installieren, aktualisieren oder deinstallieren.

Taxonomien können vom Taxonomie-Manager auch auf eine weitere Art installiert werden. Sie können eine Taxonomie und die davon abhängigen Taxonomien auf der Altova Website (<https://www.altova.com/de/taxonomy-manager>) auswählen. Daraufhin wird auf der Website eine Datei des Typs **.altova_taxonomies** mit Informationen über Ihre ausgewählte Taxonomie zum Download vorbereitet. Bei Doppelklick auf diese Datei oder bei Übergabe an den Taxonomie-Manager über das CLI als Argument des Befehls **install**¹⁷⁵ installiert der Taxonomie-Manager die ausgewählten Taxonomien.

Lokaler Cache: Überprüfung Ihrer Taxonomien

Alle Informationen über installierte Taxonomien werden in einem zentralen Cache-Verzeichnis auf Ihrem Rechner aufgezeichnet. Das Verzeichnis befindet sich hier:

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

Dieses Cache-Verzeichnis wird regelmäßig mit dem neuesten Status der Taxonomien im Online-Speicher von Altova aktualisiert. Diese Aktualisierungen finden unter den folgenden Bedingungen statt:

- bei jedem Start von Taxonomie-Manager.
- Wenn Sie MapForce Server zum ersten Mal an einem bestimmten Kalendertag starten.
- Wenn MapForce Server länger als 24 Stunden geöffnet ist, findet alle 24 Stunden eine Aktualisierung des Cache statt.
- Sie können den Cache auch durch Ausführung des **update**¹⁷⁸-Befehls über die Befehlszeilenschnittstelle aktualisieren.

Der Taxonomie-Manager kann somit Ihre installierten Taxonomien über den Cache ständig anhand der online verfügbaren Taxonomien auf der Altova Website überprüfen.

Nehmen Sie keine manuellen Änderungen am Cache vor!

Das lokale Cache-Verzeichnis wird automatisch auf Basis der installierten oder deinstallierten Taxonomien verwaltet; es sollte nicht manuell geändert oder gelöscht werden. Falls Sie den Taxonomie-Manager je in seinen Originalzustand zurücksetzen möchten, (i) führen Sie den CLI-Befehl **reset**¹⁷⁶ der Befehlszeilenschnittstelle und (ii) anschließend den Befehl **initialize**¹⁷⁴ aus. (Führen Sie alternativ dazu den Befehl **reset** mit der Option **-i** aus).

HTTP-Proxy

Sie können für Taxonomie-Manager-Verbindungen einen HTTP-Proxy verwenden. Es werden die Proxy-Einstellungen des Systems verwendet.

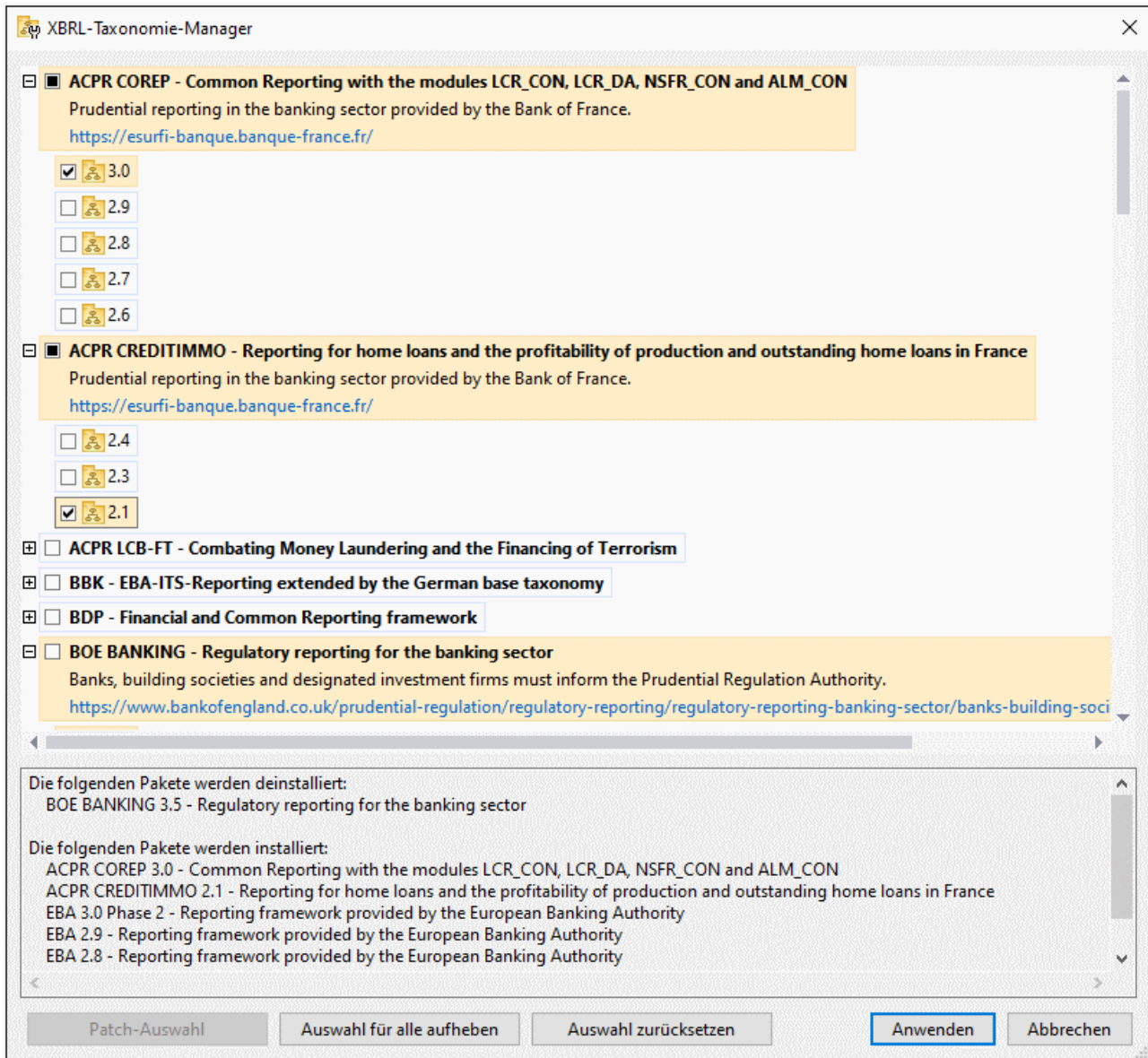
7.1 Ausführen des Taxonomie-Managers

Grafische Benutzeroberfläche

Sie können die Benutzeroberfläche des Taxonomie-Managers auf eine der folgenden Arten aufrufen:

- *Bei der Installationen von MapForce Server:* Aktivieren Sie gegen Ende der Installation das Kontrollkästchen *Altova-Taxonomie-Manager aufrufen*, wodurch Sie die Benutzeroberfläche des XBRL-Taxonomie-Managers direkt aufrufen können. Auf diese Art können Sie Taxonomien während der Installation Ihrer Altova-Applikation installieren.
- über die vom [XBRL-Taxonomie-Download-Center von Altova](#) heruntergeladene **.altova_taxonomies**-Datei: Doppelklicken Sie auf die heruntergeladene Datei, um den Taxonomie-Manager zu starten, der daraufhin die (auf der Website) ausgewählten Taxonomien installiert.

Nachdem die Benutzeroberfläche des Taxonomie-Managers geöffnet wurde (*Abbildung unten*), werden bereits installierte Taxonomien markiert angezeigt. Wenn eine zusätzliche Taxonomie installiert werden soll, aktivieren Sie diese. Wenn eine bereits installierte Taxonomie deinstalliert werden soll, deaktivieren Sie diese. Nachdem Sie Ihre Auswahl getroffen haben, können Ihre Änderungen angewendet werden. Die Taxonomien, die installiert bzw. deinstalliert werden, werden markiert und im Fenster "Meldungen" am unteren Rand des Taxonomie-Manager-Fensters (*siehe Abbildung*) erscheint eine Meldung über die bevorstehenden Änderungen.



Wenn Sie auf **Anwenden** klicken, wird der Fortschritt der Installation in einem vereinfachten Taxonomie-Manager-Dialogfeld angezeigt. Bei Auftreten eines Fehlers (z.B. eines Verbindungsfehlers) wird im Dialogfeld eine Fehlermeldung angezeigt. Klicken Sie in diesem Fall auf die Schaltfläche **Erweitert**, überprüfen Sie die Taxonomieauswahl und versuchen Sie es erneut durch Klicken auf **Anwenden**.

Befehlszeilenschnittstelle

Sie können den Taxonomie-Manager über eine Befehlszeilenschnittstelle starten, indem Sie Befehle an die ausführbare Datei **taxonomymanager.exe** senden.

Die Datei **taxonomymanager.exe** steht im folgenden Ordner zur Verfügung:

- *Unter Windows:* C:\ProgramData\Altova\SharedBetweenVersions

- *Unter Linux oder macOS (nur Server-Applikationen):* %INSTALLDIR%/bin, wobei %INSTALLDIR% das Installationsverzeichnis des Programms ist.

Anschließend können Sie jeden der im Abschnitt zur CLI-Befehlsreferenz aufgelisteten Befehle verwenden.

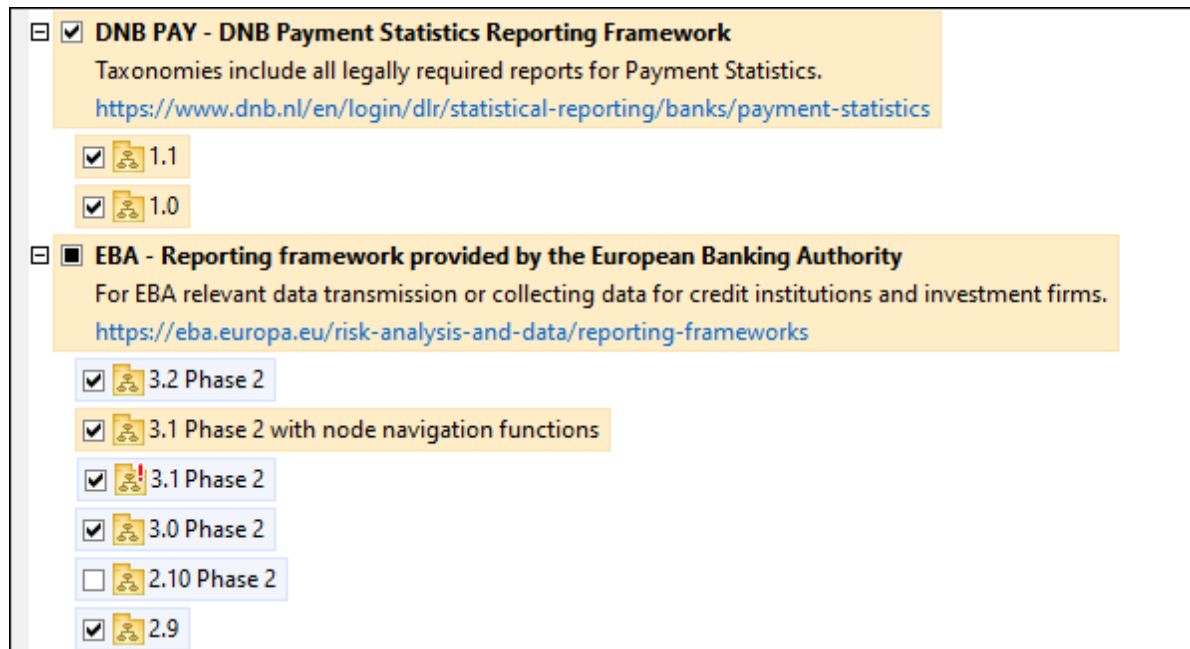
Um die Hilfe zu den Befehlen anzuzeigen, führen Sie den folgenden Befehl aus:

- *Unter Windows:* `taxonomymanager.exe --help`
- *Unter Linux oder macOS (nur Server-Applikationen):* `sudo ./taxonomymanager --help`

7.2 Statuskategorien

Der Taxonomie-Manager unterscheidet folgendermaßen zwischen den von ihm verwalteten Taxonomien:

- **Installierte Taxonomien** Diese werden auf der Benutzeroberfläche mit einem Häkchen angezeigt (in der Abbildung unten sind die mit einem Häkchen versehenen Versionen der DNB- und EBA-Taxonomie installiert). Wenn alle Versionen einer Taxonomie ausgewählt sind, wird ein Häkchen angezeigt. Wenn zumindest eine Version nicht ausgewählt ist, wird die Auswahl in Form eines gefüllten Quadrats angezeigt (siehe EBA in der Abbildung unten). Sie können die Auswahl für eine installierte Taxonomie aufheben, um sie zu **deinstallieren**.
- **Nicht installierte verfügbare Taxonomien**. Diese werden auf der Benutzeroberfläche mit einem deaktivierten Kontrollkästchen angezeigt. Sie können die Taxonomien, die **installiert** werden sollen, auswählen.



- **Taxonomien, für die ein Upgrade zur Verfügung steht** sind diejenigen, die seit ihrer Installation vom Herausgeber überarbeitet wurden. Sie werden auf der Benutzeroberfläche durch ein Symbol gekennzeichnet (siehe Abbildung oben). Sie können für eine installierte Taxonomie ein **Patch** der überarbeiteten Version installieren.

Wichtige Punkte

- In der Abbildung oben sind beide DNB-Taxonomien und einige der EBA-Taxonomien ausgewählt. Diejenigen mit einem blauen Hintergrund sind bereits installiert. Diejenige mit gelbem Hintergrund sind nicht installiert und wurden für die Installation ausgewählt. Beachten Sie, dass (i) die EBA 2.10 Phase 2-Taxonomie nicht installiert wurde und nicht für die Installation ausgewählt wurde, (ii) die EBA 3.1 Phase 2-Taxonomie installiert wurde, dafür aber seit ihrer Installation ein Patch vom Herausgeber zur Verfügung gestellt wurde, das noch nicht installiert wurde.
- Bei Ausführung des Taxonomie-Managers über die Befehlszeile wird der Befehl `list`¹⁷⁶ mit verschiedenen Optionen verwendet, um verschiedene Taxonomiekategorien aufzulisten:

taxonomymanager.exe list	Listet alle installierten und verfügbaren Taxonomien auf; auch verfügbare Upgrades werden angezeigt.
taxonomymanager.exe list -i	Listet nur installierte Taxonomien auf; auch verfügbare Upgrades werden angezeigt.
taxonomymanager.exe list -u	Listet für Taxonomien auf, für die Upgrades zur Verfügung stehen

Anmerkung: Verwenden Sie unter Linux und macOS, **sudo ./taxonomymanager list**

7.3 Anwenden eines Patch oder Installation einer Taxonomie

Anwenden eines Patch auf eine installierte Taxonomie

Von Zeit zu Zeit werden von den Herausgebern der XBRL-Taxonomien Patches (Upgrades oder Überarbeitungen) veröffentlicht. Wenn der Taxonomie-Manager erkennt, dass Patches zur Verfügung stehen, werden diese in der Taxonomieliste des Taxonomie-Managers angezeigt und Sie können diese Patches schnell installieren.

Über die Benutzeroberfläche

Patches werden mit dem Symbol  gekennzeichnet. (Siehe auch vorhergehendes Kapitel über [Statuskategorien](#)¹⁶⁸). Falls Patches zur Verfügung stehen, ist die Schaltfläche **Patch-Auswahl** aktiv. Klicken Sie darauf, um alle Patches für die Installation auszuwählen und vorzubereiten. Auf der Benutzeroberfläche ändert sich das Symbol von Taxonomien, für die ein Patch installiert wird von  in , und im Fenster "Meldungen" am unteren Rand des Dialogfelds werden die Patches, die angewendet werden, aufgelistet. Sobald Sie mit der Auswahl fertig sind, klicken Sie auf **Anwenden**. Alle Patches werden gemeinsam angewendet. Beachten Sie, dass eine für die Installation eines Patch markierte Taxonomie deinstalliert wird, wenn Sie die Auswahl für diese Taxonomie aufheben.

Über das CLI

So wenden Sie einen Patch über die Befehlszeilenschnittstelle an:

1. Führen Sie den Befehl [list -u](#)¹⁷⁶ aus. Daraufhin werden alle Taxonomien, für die Patch Upgrades zur Verfügung stehen, aufgelistet.
2. Führen Sie den Befehl [upgrade](#)¹⁷⁹ aus, um alle Patches zu installieren.

Installieren einer verfügbaren Taxonomie

Sie können Taxonomien entweder über die Benutzeroberfläche des Taxonomie-Managers oder durch Senden der Taxonomie-Manager-Installationsbefehle über die Befehlszeile installieren.

Anmerkung: Wenn die aktuelle Taxonomie andere Taxonomien referenziert, werden auch die referenzierten Taxonomien installiert.

Über die Benutzeroberfläche

Um Taxonomien über die Benutzeroberfläche des Taxonomie-Managers zu installieren, wählen Sie die gewünschten Taxonomien aus und klicken Sie auf **Anwenden**.

Sie können die gewünschten Taxonomien auch auf der [Altova Website](#) auswählen und eine herunterladbare `.altova_taxonomies`-Datei generieren. Bei Doppelklick auf diese Datei wird der Taxonomie-Manager aufgerufen, in dem die gewünschten Taxonomien bereits vorausgewählt sind. Sie müssen nur mehr auf **Anwenden** klicken.

Über das CLI

Um Taxonomien über die Befehlszeile zu installieren, rufen Sie den Befehl [install](#)¹⁷⁵ auf:

```
taxonomymanager.exe install [options] Taxonomy+
```

wobei es sich bei **Taxonomy** um die gewünschte(n) Taxonomie(n) bzw. eine **.altova_taxonomies**-Datei handelt. Eine Taxonomie wird von einem Identifier im Format **<name>-<version>** referenziert. (Die Identifier von Taxonomien werden angezeigt, wenn Sie den Befehl [list](#)¹⁷⁶ ausführen.) Sie können beliebig viele Taxonomien eingeben. Nähere Informationen dazu finden Sie unter der Beschreibung des Befehls [install](#)¹⁷⁵.

Anmerkung: Verwenden Sie unter Linux oder macOS den Befehl **sudo ./taxonomymanager**.

Installation einer benötigten Taxonomie

Wenn Sie einen XBRL-Befehl in MapForce Server ausführen und MapForce Server erkennt, dass eine zur Ausführung des Befehls erforderliche Taxonomie nicht vorhanden oder unvollständig ist, wird der Taxonomie-Manager mit Informationen über die fehlende Taxonomie aufgerufen. Sie können die gewünschten Taxonomien dann über den Taxonomie-Manager direkt installieren.

Alle bereits installierten Taxonomien können jederzeit durch Aufruf des Taxonomie-Managers über **Extras | Taxonomie-Manager** über die Benutzeroberfläche des Taxonomie-Managers angezeigt werden.

7.4 Deinstallieren einer Taxonomie, Zurücksetzen

Deinstallieren einer Taxonomie

Sie können Taxonomien entweder über die Benutzeroberfläche des Taxonomie-Managers oder durch Senden der Taxonomie-Manager-Deinstallationsanweisungen über die Befehlszeile deinstallieren.

Anmerkung: Wenn die gewünschte Taxonomie andere Taxonomien referenziert, so werden auch die referenzierten Taxonomien deinstalliert.

Über die Benutzeroberfläche

Um Taxonomien über die Benutzeroberfläche des Taxonomie-Managers zu deinstallieren, deaktivieren Sie die Kontrollkästchen der entsprechenden Taxonomien und klicken Sie auf **Anwenden**. Daraufhin werden die ausgewählten Taxonomien und die davon referenzierten Taxonomien deinstalliert.

Um alle Taxonomien zu deinstallieren, klicken Sie auf **Auswahl für alle aufheben** und anschließend auf **Anwenden**.

Über das CLI

Um Taxonomien über die Befehlszeile zu deinstallieren, rufen Sie den Befehl **uninstall** auf:

```
taxonomymanager.exe uninstall [options] Taxonomy+
```

wobei es sich beim Argument **Taxonomy** eine zu deinstallierende Taxonomie oder eine **.altova_taxonomies**-Datei handelt. Eine Taxonomie wird von einem Identifier im Format **<name>-<version>** definiert. (Die Identifier von Taxonomien werden angezeigt, wenn Sie den Befehl [list](#)¹⁷⁶ ausführen.) Sie können beliebig viele Taxonomien eingeben. Nähere Informationen dazu finden Sie unter der Beschreibung des Befehls [uninstall](#)¹⁷⁷.

Anmerkung: Verwenden Sie unter Linux oder macOS den Befehl **sudo ./taxonomymanager**.

Zurücksetzen des Taxonomie-Managers

Sie können den Taxonomie-Manager zurücksetzen.

- Klicken Sie auf der Benutzeroberfläche auf **Auswahl zurücksetzen**. Damit wird die Benutzeroberfläche zurückgesetzt, und Sie sehen, welche Taxonomien derzeit installiert sind. Jegliche Auswahl bzw. Aufhebung einer Auswahl, die der Benutzer in der aktuellen Sitzung vorgenommen hat, wird aufgehoben.
- Führen Sie über die Benutzeroberfläche den Befehl [reset](#)¹⁷⁶ aus. Damit werden alle installierten Taxonomien und das Cache-Verzeichnis entfernt.

Nachdem Sie diesen Befehl ausgeführt haben, muss der Befehl [initialize](#)¹⁷⁴ ausgeführt werden, um das Cache-Verzeichnis neu zu erstellen. Führen Sie alternativ dazu den Befehl [reset](#)¹⁷⁶ mit der Option **-i** aus.

Beachten Sie, dass mit [reset-i](#)¹⁷⁶ die Originalinstallation des Produkts wiederhergestellt wird, daher wird empfohlen, nach dem Zurücksetzen auch den Befehl [update](#)¹⁷⁸ auszuführen. Führen Sie alternativ dazu den Befehl [reset](#)¹⁷⁶ mit den Optionen **-i** und **-u** aus.

7.5 Befehlszeilenschnittstelle (CLI)

Um den Taxonomie-Manager über die Befehlszeile aufzurufen, müssen Sie den Pfad zur ausführbaren Datei kennen. Standardmäßig befindet sich die ausführbare Taxonomie-Manager-Datei hier:

<i>Windows</i>	C:\ProgramData\Altova\SharedBetweenVersions\TaxonomyManager.exe
<i>Linux</i>	/opt/Altova/MapForceServer2025/bin/taxonomymanager
<i>macOS</i>	/usr/local/Altova/MapForceServer2025/bin/taxonomymanager

Anmerkung: Nachdem Sie auf Linux- und macOS-Systemen das Verzeichnis in dasjenige, das die ausführbare Datei enthält, geändert haben, können Sie die ausführbare Datei mit **sudo ./taxonomymanager** aufrufen. Das Präfix **./** gibt an, dass sich die ausführbare Datei im aktuellen Verzeichnis befindet. Das Präfix **sudo** gibt an, dass der Befehl mit Root-Rechten ausgeführt werden muss.

Befehlszeilensyntax

Die allgemeine Syntax zur Verwendung der Befehlszeile lautet folgendermaßen:

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

Der senkrechte Balken **|** im Codefragment oben trennt eine Gruppe einander gegenseitig ausschließender Elemente. Optionale Elemente stehen innerhalb von eckigen Klammern **[]**. Im Prinzip können Sie den Pfad zur ausführbaren Datei, gefolgt von entweder **-h**, **--help** oder **--version**-Optionen oder gefolgt von einem Befehl eingeben. Jeder Befehl kann Optionen und Argumente haben. Die Liste der Befehle wird in den folgenden Abschnitten beschrieben.

7.5.1 help

Mit diesem Befehl erhalten Sie Hilfe zu Befehlen zur ausführbaren Taxonomie-Manager-Datei.

Syntax

```
<exec> help [Befehl]
```

[Befehl] ist hierbei ein optionales Argument zur Angabe jedes beliebigen gültigen Befehlsnamens.

Beachten Sie dazu Folgendes:

- Sie können die Hilfe zu einem Befehl auch durch Eingabe des Befehls, gefolgt von **-h** oder **--help** aufrufen, z.B: **<exec> list -h**
- Wenn Sie **-h** oder **--help** direkt nach dem Namen der ausführbaren Datei und vor einem Befehl eingeben, wird die allgemeine Hilfe (und nicht die Hilfe zu einem bestimmten Befehl) angezeigt, z.B: **<exec> -h list**

Beispiel

Mit dem folgenden Befehl wird Hilfe zum Befehl `list` angezeigt:

```
taxonomymanager help list
```

7.5.2 info

Mit diesem Befehl werden ausführliche Informationen über die einzelnen als Taxonomy-Argument angegebenen Taxonomien angezeigt. Darin enthalten sind Titel, Version, Beschreibung, Herausgeber der jeweils angegebenen Taxonomie und davon abhängige Taxonomien sowie die Information, ob die Taxonomie installiert ist oder nicht.

Syntax

```
<exec> info [options] Taxonomy+
```

- Das Argument **Taxonomy** ist der Name einer Taxonomie oder Teil eines Taxonomienamens. (Die Paket-ID einer Taxonomie und detaillierte Informationen über ihren Installationsstatus erhalten Sie mit dem Befehl [list](#)¹⁷⁶.)
- Mit **<exec> info -h** können Sie die Hilfe zum Befehl anzeigen.

Beispiel

Mit dem folgenden Befehl werden Informationen über die Taxonomien **eba-2.10** und **us-gaap-2020.0** angezeigt:

```
taxonomymanager info eba-2.10 us-gaap-2020.0
```

7.5.3 initialize

Mit diesem Befehl wird die Taxonomie-Manager-Umgebung initialisiert. Sie erstellen damit ein Cache-Verzeichnis, in dem Informationen über alle Taxonomien lokal gespeichert werden. Die Initialisierung erfolgt automatisch bei der ersten Installation einer XBRL-fähigen Altova-Applikation. Normalerweise muss dieser Befehl nicht ausgeführt werden. Nach Ausführung des **reset**-Befehls ist dies allerdings erforderlich.

Syntax

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

Optionen

Für den Befehl **initialize** stehen die folgenden Optionen zur Verfügung:

`--silent, --s` Nur Fehlermeldungen anzeigen. Der Standardwert ist **false**.

<code>--verbose, --v</code>	Anzeige detaillierter Informationen während der Ausführung. Der Standardwert ist false .
<code>--help, --h</code>	Anzeige der Hilfe zum Befehl.

Beispiel

Mit dem folgenden Befehl wird der Taxonomie-Manager initialisiert:

```
taxonomymanager initialize
```

7.5.4 install

Mit diesem Befehl installieren Sie eine oder mehrere Taxonomien.

Syntax

```
<exec> install [options] Taxonomy+
```

Um mehrere Taxonomien zu installieren, fügen Sie das Argument **Taxonomy** mehrmals hinzu.

Als **Taxonomy**-Argument kann eines der folgenden verwendet werden:

- Ein Taxonomie-Identifizier (im Format **<name>-<version>**, z.B. **eba-2.10**). Um die Taxonomie-Identifizier der gewünschten Taxonomien zu eruieren, führen Sie den Befehl [list](#)¹⁷⁶ aus. Sie können auch einen abgekürzten Identifizier verwenden, sofern dieser eindeutig ist, z.B. **eba**. Falls Sie einen abgekürzten Identifizier verwenden, wird die neueste Version dieser Taxonomie installiert.
- Der Pfad zu einer von der Altova-Website heruntergeladenen **.altova_taxonomies**-Datei. Informationen zu diesen Dateien finden Sie in [der Einführung zum Taxonomie-Manager.: Funktionsweise](#)¹⁶¹.

Optionen

Für den Befehl **install** stehen die folgenden Optionen zur Verfügung:

<code>--silent, --s</code>	Nur Fehlermeldungen anzeigen. Der Standardwert ist false .
<code>--verbose, --v</code>	Anzeige detaillierter Informationen während der Ausführung. Der Standardwert ist false .
<code>--help, --h</code>	Anzeige der Hilfe zum Befehl.

Beispiel

Mit dem folgenden Befehl werden die neuesten **eba** (European Banking Authority) und **us-gaap** (US Generally Accepted Accounting Principles) Taxonomien installiert:

```
taxonomymanager install eba us-gaap
```

7.5.5 list

Mit diesem Befehl werden vom Taxonomie-Manager verwaltete Taxonomien aufgelistet. In der Liste wird eine der folgenden Informationen angezeigt:

- Alle verfügbaren Taxonomien
- Taxonomien, die im Namen den im Argument **Taxonomy** angegebenen String enthalten
- Nur installierte Taxonomien
- Nur Taxonomien, für die ein Upgrade installiert werden kann

Syntax

```
<exec> list | ls [options] Taxonomy?
```

Wenn kein **Taxonomy**-Argument angegeben wird, werden alle verfügbaren Taxonomien aufgelistet. Andernfalls werden die durch die angegebenen Optionen definierten Taxonomien aufgelistet (*siehe Beispiel unten*). Beachten Sie, dass Sie das Argument **Taxonomy** mehrfach angeben können.

Optionen

Für den Befehl **list** stehen die folgenden Optionen zur Verfügung:

--installed, --i	Auflisten nur der installierten Taxonomien. Der Standardwert ist false .
--upgradeable, --u	Auflisten nur derjenigen Taxonomien, für die Upgrades (Patches) zur Verfügung stehen. Der Standardwert ist false .
--help, --h	Anzeige der Hilfe zum Befehl.

Beispiele

- Um alle verfügbaren Taxonomien aufzulisten, führen Sie den folgenden Befehl aus: **taxonomymanager list**
- Um installierte Taxonomien aufzulisten, führen Sie **taxonomymanager list -i** aus.
- Um Taxonomien, die in ihrem Namen entweder "eba" oder "us-gaap" enthalten, aufzulisten, führen Sie **taxonomymanager list eba us-gaap** aus.

7.5.6 reset

Mit diesem Befehl werden alle installierten Taxonomien und das Cache-Verzeichnis entfernt. Ihre Taxonomieumgebung wird vollständig zurückgesetzt. Nachdem Sie diesen Befehl ausgeführt haben, muss der Befehl [initialize](#)¹⁷⁴ ausgeführt werden, um das Cache-Verzeichnis neu zu erstellen. Führen Sie alternativ dazu den Befehl **reset** mit der Option **-i** aus. Da mit **reset -i** die Originalinstallation des Produkts wiederhergestellt wird, wird empfohlen, nach dem Zurücksetzen und Initialisieren auch den Befehl [update](#)¹⁷⁸ auszuführen. Führen Sie alternativ dazu den Befehl **reset** mit den Optionen **-i** und **-u** aus.

Syntax

<exec> reset [Optionen]

Optionen

Für den Befehl **reset** stehen die folgenden Optionen zur Verfügung:

<code>--init, --i</code>	Initialisierung des Taxonomie-Managers nach dem Zurücksetzen. Der Standardwert ist false .
<code>--update, --u</code>	Aktualisiert die Liste der verfügbaren Taxonomien im Cache. Der Standardwert ist false .
<code>--silent, --s</code>	Nur Fehlermeldungen anzeigen. Der Standardwert ist false .
<code>--verbose, --v</code>	Anzeige detaillierter Informationen während der Ausführung. Der Standardwert ist false .
<code>--help, --h</code>	Anzeige der Hilfe zum Befehl.

Beispiele

- Um den Taxonomie-Manager zurückzusetzen, führen Sie den folgenden Befehl aus: **taxonomymanager reset**
- Um den Taxonomie-Manager zurückzusetzen und ihn zu initialisieren, führen Sie **taxonomymanager reset -i** aus.
- Um den Taxonomie-Manager zurückzusetzen, ihn zu initialisieren und seine Taxonomieliste zu aktualisieren, führen Sie **taxonomymanager reset -i -u** aus.

7.5.7 uninstall

Mit diesem Befehl deinstallieren Sie eine oder mehrere Taxonomien. Standardmäßig werden auch alle Taxonomien, die von der aktuellen Taxonomie referenziert werden, deinstalliert. Um nur die aktuelle Taxonomie zu deinstallieren und die referenzierten Taxonomien beizubehalten, setzen Sie die Option `--k`.

Syntax

<exec> uninstall [options] Taxonomy+

Um mehrere Taxonomien zu deinstallieren, fügen Sie das Argument **Taxonomy** mehrmals hinzu.

Als **Taxonomy**-Argument kann eines der folgenden verwendet werden:

- Ein Taxonomie-Identifizier (im Format **<name>-<version>**, z.B: **eba-2.10**). Um die Taxonomie-Identifizier der installierten Taxonomien zu eruieren, führen Sie den Befehl **list -i** ¹⁷⁶ aus. Sie können auch einen abgekürzten Taxonomienamen verwenden, sofern dieser eindeutig ist, z.B. **eba**. Falls Sie einen abgekürzten Namen verwenden, werden alle Taxonomien, die die Abkürzung in ihrem Namen enthalten, deinstalliert.

- Der Pfad zu einer von der Altova-Website heruntergeladenen **.altova_taxonomies**-Datei. Informationen zu diesen Dateien finden Sie [der Einführung zum Taxonomie-Manager.: Funktionsweise](#)¹⁶¹.

Optionen

Für den Befehl **uninstall** stehen die folgenden Optionen zur Verfügung:

<code>--keep-references, --k</code>	Definieren Sie diese Option, um referenzierte Taxonomien beizubehalten. Der Standardwert ist false .
<code>--silent, --s</code>	Nur Fehlermeldungen anzeigen. Der Standardwert ist false .
<code>--verbose, --v</code>	Anzeige detaillierter Informationen während der Ausführung. Der Standardwert ist false .
<code>--help, --h</code>	Anzeige der Hilfe zum Befehl.

Beispiel

Mit dem folgenden Befehl werden die Taxonomien **eba-2.10** und **us-gaap-2020.0** und deren Abhängigkeiten deinstalliert:

```
taxonomymanager uninstall eba-2.10 us-gaap-2020.0
```

Mit dem folgenden Befehl wird die **eba-2.10**-Taxonomie, nicht aber die davon referenzierten Taxonomien deinstalliert:

```
taxonomymanager uninstall --k eba-2.10
```

7.5.8 update

Mit diesem Befehl wird die Liste der über den Online-Speicher verfügbaren Taxonomien abgefragt und das lokale Cache-Verzeichnis wird aktualisiert. Normalerweise muss dieser Befehl nur ausgeführt werden, wenn Sie [reset](#)¹⁷⁶ und [initialize](#)¹⁷⁴ ausgeführt haben.

Syntax

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

Optionen

Für den Befehl **update** stehen die folgenden Optionen zur Verfügung:

<code>--silent, --s</code>	Nur Fehlermeldungen anzeigen. Der Standardwert ist false .
<code>--verbose, --v</code>	Anzeige detaillierter Informationen während der Ausführung. Der Standardwert ist false .
<code>--help, --h</code>	Anzeige der Hilfe zum Befehl.

Beispiel

Mit dem folgenden Befehl wird der lokale Cache mit der Liste der neuesten Taxonomien aktualisiert:

taxonomymanager update

7.5.9 upgrade

Mit diesem Befehl werden alle installierten Taxonomien, für die ein Upgrade installiert werden kann, auf die neueste verfügbare *Patch*-Version aktualisiert. Um herauszufinden, welche Taxonomien aktualisiert werden können, starten Sie den Befehl [list -u](#)¹⁷⁶.

Anmerkung: Der Befehl **upgrade** würde eine veraltete Taxonomie entfernen, falls keine neuere Version zur Verfügung steht.

Syntax

<exec> upgrade [Optionen]

Optionen

Für den Befehl **upgrade** stehen die folgenden Optionen zur Verfügung:

--silent, --s	Nur Fehlermeldungen anzeigen. Der Standardwert ist false .
--verbose, --v	Anzeige detaillierter Informationen während der Ausführung. Der Standardwert ist false .
--help, --h	Anzeige der Hilfe zum Befehl.

8 Katalogdateien

Wenn Sie den Altova-Katalog-Mechanismus verwenden, finden Sie die entsprechenden Katalogdateien im Ordner **etc** des MapForce Server-Applikationsordners. Ausführliche Informationen dazu finden Sie [im Abschnitt "Kataloge" des Altova MapForce-Handbuchs](#).

CustomCatalog.xml kann anhand der Vorlagendatei **CustomCatalog_template.xml** erstellt werden. Benennen Sie die Vorlagendatei in **CustomCatalog.xml** um, da dies die Datei ist, die im Katalogmechanismus verwendet wird (und nicht die Vorlagendatei).

Beachten Sie dazu Folgendes:

- Bei einer Neuinstallation derselben Hauptversion (selbe oder unterschiedliche Nebenversion) wird die Vorlagendatei durch eine neue Vorlagendatei ersetzt, wobei **CustomCatalog.xml** jedoch unverändert bleibt.
- Wenn Sie jedoch eine neue Hauptversion über eine vorherige Hauptversion installieren, wird der Ordner der vorherigen Hauptversion zusammen mit seiner **CustomCatalog.xml**-Datei gelöscht. Wenn Sie also **CustomCatalog.xml** weiterhin verwenden möchten, müssen Sie **CustomCatalog.xml** aus dem Ordner der vorherigen Hauptversion in einem anderen Ordner sichern. Nach Installation der neuen Hauptversion können Sie die zuvor gespeicherte Datei **CustomCatalog.xml** in den Ordner **etc** der neuen Hauptversion kopieren und dort je nach Bedarf bearbeiten.

Index

A

accepteula,

als CLI-Befehl, 66

ADO, 6, 34**ADO.NET, 6, 34****Altova ServiceController, 13****assignlicense,**

als CLI-Befehl, 67

B

Befehlszeilenschnittstelle,

Verwendung, 63

C

C#,

Beispiel, 81

Mappings ausführen mit, 81

C++,

Beispiel, 88

Mappings ausführen mit, 88

CLI-Befehle,

accepteula, 66

assignlicense, 67

exportresourcestrings, 68

help, 70

licenseserver, 71

run, 72

setdeflang, 76

verifylicense, 77, 78

D

Deinstallation, 9**Deinstallieren, 9****Digitale Zertifikate,**

auf Linux übertragen, 155

auf Mac übertragen, 157

aus Windows exportieren, 149

in MapForce Mappings, 142

unter Windows verwalten, 148

vertrauenswürdige auf Mac-Systemen einrichten, 146

vertrauenswürdige unter Linux einrichten, 145

vertrauenswürdige unter Windows einrichten, 147

E

Einrichten,

auf macOS, 23

unter Linux, 17

unter Windows, 9

Einrichten von MapForce Server, 8**exportresourcestrings,**

als CLI-Befehl, 68

F

Fenster,

vertrauenswürdige Server-Zertifikate einrichten, 147

FlowForce Server, 6, 31, 34

G

Globale Ressourcen,

Definition, 40

in Mappings verwenden, 40

H

help,

als CLI-Befehl, 70

HTTPS,

Webservices aufrufen über, 142

I

Installation auf macOS, 23
Installation auf Windows Server Core, 10
Installation unter Linux, 17
Installation unter Windows, 9
Installation von LicenseServer auf macOS, 24
Installation von LicenseServer unter Linux, 19
Installieren von LicenseServer unter Windows, 12
Installieren von MapForce Server, 8

J

Java, 34
 Beispiel, 101
 MapForce Server aufrufen über, 101
 MapForce Server zu CLASSPATH hinzufügen, 97
JDBC, 6
Join-Optimierung,
 Funktionsweise, 42

L

licenseserver,
 als CLI-Befehl, 71
LicenseServer-Versionen, 12, 19, 24
Linux,
 Client-Zertifikate übertragen auf, 155
 Installation, 17
 Mappings mit Webservice-Aufrufen über HTTPS ausführen, 155
 vertrauenswürdige Server-Zertifikate einrichten, 145
Lizenzieren von MapForce Server, 8
 Lizenz auf macOS zuweisen, 26
 Lizenz unter Linux zuweisen, 21
 unter Windows eine Lizenz zuweisen, 15

M

Mac,

 Client-Zertifikate übertragen auf, 157
 Mappings mit Webservice-Aufrufen über HTTPS ausführen, 157
 vertrauenswürdige Server-Zertifikate einrichten, 146

macOS,

 Installation, 23

MapForce Server,

 auf einen neuen Rechner migrieren, 29

MapForce Server API,

 C#-Beispiel, 81
 C++-Beispiel, 88
 Einführung, 79
 für .NET, 80
 für COM, 88
 für Java, 97
 Java-Beispiel, 101
 VBA-Beispiel, 94

Mappings,

 auf FlowForce Server bereitstellen, 31
 mit C# ausführen, 81
 mit C++ ausführen, 88
 mit Java ausführen, 101
 mit VBA ausführen, 94
 über die Befehlszeile ausführen, 72
 vorbereiten für die Ausführung, 34
 zu Ausführungsdateien kompilieren, 31

Migrieren von MapForce Server auf einen neuen Rechner, 29

O

ODBC, 6, 34

R

Registrieren von MapForce Server auf LicenseServer auf macOS, 25

Registrieren von MapForce Server auf LicenseServer unter Linux, 20

Registrieren von MapForce Server auf LicenseServer unter Windows, 14

run,

 als CLI-Befehl, 72

S

Security considerations, 30

setdeflang,

als CLI-Befehl, 76

Starten von LicenseServer auf macOS, 25

Starten von LicenseServer unter Linux, 20

Starten von LicenseServer unter Windows, 13

Starten von MapForce Server auf macOS, 25

Starten von MapForce Server unter Linux, 20

Starten von MapForce Server unter Windows, 13

T

Taxonomie-Manager,

CLI-Befehl Help, 173

CLI-Befehl Info, 174

CLI-Befehl Initialize, 174

CLI-Befehl Install, 175

CLI-Befehl List, 176

CLI-Befehl Reset, 176

CLI-Befehl Uninstall, 177

CLI-Befehl Update, 178

CLI-Befehl Upgrade, 179

CLI-Übersicht, 173

Installieren einer Taxonomie, 170

Patch auf Taxonomie anwenden, 170

starten, 165

Status von Taxonomien, 168

Taxonomie deinstallieren, 172

Taxonomien nach Status auflisten in, 168

Übersicht, 161

Upgrade für eine Taxonomie installieren, 170

zurücksetzen, 172

U

Upgraden von MapForce Server unter Windows, 28

V

VBA,

Beispiel, 94

Mappings ausführen mit, 94

verifylicense,

als CLI-Befehl, 77

version,

als CLI-Befehl, 78

W

Webservices,

über HTTPS aufrufen, 142

Windows,

Installation, 9

Mappings mit Webservice-Aufrufen über HTTPS ausführen, 159

Upgraden von MapForce Server unter, 28

X

XML-Kataloge,

Funktionsweise, 180

konfigurieren, 180

Z

Zuweisen einer Lizenz zu MapForce Server auf macOS, 26

Zuweisen einer Lizenz zu MapForce Server unter Linux, 21

Zuweisen einer Lizenz zu MapForce Server unter Windows, 15