

Altova MapForce Server 2021

Benutzer- und Referenzhandbuch

Altova MapForce Server 2021

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: 2020

© 2020 Altova GmbH

Inhaltsverzeichnis

1	Einführung	6
2	Installation	7
2.1	Systemanforderungen.....	7
2.2	Prozessorkerne und Lizenzen.....	8
2.3	Wichtige Pfade.....	9
2.4	Linux	10
2.4.1	Installation von MapForce Server.....	10
2.4.2	Anzeige der derzeit installierten Produkte.....	11
2.4.3	Deinstallation von MapForce Server.....	11
2.5	macOS.....	12
2.5.1	Installation von MapForce Server.....	12
2.5.2	Deinstallation von MapForce Server.....	13
2.6	Windows.....	14
2.6.1	Installation von MapForce Server.....	14
2.6.2	Deinstallation von MapForce Server.....	15
3	Ausführen von Mappings	16
3.1	Vorbereiten von Mappings für die Server-Ausführung.....	19
3.2	Globale Ressourcen.....	25
3.3	Anmeldeinformationen.....	27
3.4	Dynamische Authentifizierung.....	29
4	MapForce Server-Befehlszeile	30
4.1	assignlicense (nur Windows).....	33
4.2	exportresourcestrings.....	34
4.3	help	36

4.4	licenseserver.....	37
4.5	run	39
4.6	setdeflang.....	43
4.7	verifylicense (nur Windows).....	44
4.8	version.....	45

5 MapForce Server API 46

5.1	.NET-Schnittstelle.....	47
5.1.1	C#-Beispiel.....	48
5.1.2	Visual Basic .NET-Beispiel.....	51
5.2	COM-Schnittstelle.....	54
5.2.1	C++-Beispiel.....	54
5.2.2	VBScript-Beispiel.....	57
5.2.3	VBA-Beispiel.....	60
5.3	Java-Schnittstelle.....	63
5.3.1	Java-Beispiel.....	65
5.4	Beispiel: Ausführen eines Mappings mit Parametern.....	68
5.5	API-Referenz (COM, .NET).....	74
5.5.1	Schnittstellen.....	74
5.6	API-Referenz (Java).....	92
5.6.1	Klassen.....	92

6 Verwaltung digitaler Zertifikate 105

6.1	Definieren vertrauenswürdiger Zertifikate unter Linux.....	108
6.2	Definieren vertrauenswürdiger Zertifikate auf macOS-Systemen.....	110
6.3	Definieren vertrauenswürdiger Zertifikate unter Windows.....	111
6.4	Aufrufen der Zertifikatspeicher unter Windows.....	112
6.5	Exportieren von Zertifikaten aus Windows.....	115
6.6	Client-Zertifikate unter Linux.....	121
6.7	Client-Zertifikate auf macOS.....	123
6.8	Client-Zertifikate unter Windows.....	125

7 Katalogdateien 127

Index

130

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 Hochleistungsrechnern
- 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](#) ²⁵
- 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: 12.10.2020

2 Installation

2.1 Systemanforderungen

▼ Windows

Windows 7 SP1 mit Plattformupdate, Windows 8, Windows 10

▼ Windows Server

Windows Server 2008 R2 SP1 mit Plattformupdate oder höher

MapForce Server steht unter Windows sowohl als 32-Bit- als auch als 64-Bit-Paket zur Verfügung.

▼ Linux

- CentOS 7 oder neuer
- RedHat 7 oder neuer
- Debian 8 oder neuer
- Ubuntu 16.04 LTS oder neuer

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.

Erforderlich für	CentOS, RedHat	Debian	Ubuntu
LicenseServer	krb5-libs	libgssapi-krb5-2	libgssapi-krb5-2
MapForce Server	libidn, krb5-libs	libidn11, libgssapi-krb5-2	libidn11, libgssapi-krb5-2

▶ macOS

macOS 10.13 oder neuer

2.2 Prozessorkerne und Lizenzen

Die Lizenzierung von Altova Server-Produkten basiert auf der Anzahl der auf dem Produktrechner verfügbaren **physischen Prozessorkerne**. So hat z.B. ein Dual-Core-Prozessor zwei Prozessorkerne, ein Quad-Core-Prozessor hat vier Kerne, ein Hexa-Core-Prozessor hat 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, müssen Sie 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 Rechnerserver 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. 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.

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 Wichtige Pfade

Installationsverzeichnis

Die Applikation wird im folgenden Verzeichnis installiert:

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

Ausführbare Datei

Die ausführbare MapForce Server-Datei befindet sich nach der Installation unter dem folgenden Pfad:

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

Konfigurationsdaten zur Applikation

Die Konfigurationsdaten der Applikation sind im folgenden Verzeichnis gespeichert:

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

2.4 Linux

2.4.1 Installation von MapForce Server

Systemanforderungen

▼ Linux

- CentOS 7 oder neuer
- RedHat 7 oder neuer
- Debian 8 oder neuer
- Ubuntu 16.04 LTS oder neuer

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.

Erforderlich für	CentOS, RedHat	Debian	Ubuntu
LicenseServer	krb5-libs	libgssapi-krb5-2	libgssapi-krb5-2
MapForce Server	libidn, krb5-libs	libidn11, libgssapi-krb5-2	libidn11, libgssapi-krb5-2

Voraussetzungen

1. Führen Sie die Installation entweder als **root**-Benutzer durch oder als Benutzer mit **sudo**-Rechten.
2. Die vorherige Version von MapForce Server muss deinstalliert werden, bevor Sie eine neue Version installieren.
3. Wenn Sie FlowForce Server mit anderen Altova Server-Produkten installieren, sollten Sie zuerst FlowForce Server installieren.

Installation von MapForce Server

1. Laden Sie das Installationspaket von <https://www.altova.com/de/download#server> in ein lokales Verzeichnis herunter.
2. Wechseln Sie in das Verzeichnis, in das Sie das Installationspaket heruntergeladen haben, z.B.:

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

3. Installieren Sie das MapForce Server-Paket.

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

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

Lizenzierung von MapForce Server

1. Stellen Sie sicher, dass die neueste Version von Altova LicenseServer (<https://www.altova.com/de/licenseserver>) installiert ist und entweder auf dem lokalen Rechner oder im Netzwerk ausgeführt wird.
2. Registrieren Sie MapForce Server auf Altova LicenseServer. Dazu benötigen Sie entweder den Host-Namen oder die IP-Adresse des Rechners, auf dem LicenseServer ausgeführt wird. Wenn LicenseServer z.B auf 127.0.0.1 läuft, können Sie Ihr Produkt folgendermaßen registrieren:

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

Anmerkung: Beim obigen Befehl wird davon ausgegangen, dass Sie die ausführbare Datei vom Programminstallationsverzeichnis aus aufrufen, siehe auch [Wichtige Pfade](#) ⁹.

3. Melden Sie sich auf der Verwaltungsoberfläche von LicenseServer an und weisen Sie die Lizenz dem Rechner zu, auf dem MapForce Server ausgeführt wird.

Nähere Informationen dazu finden Sie in der Dokumentation zu Altova LicenseServer (<https://www.altova.com/de/documentation>).

2.4.2 Anzeige der derzeit installierten Produkte

Um zu überprüfen, ob ein Altova Server-Produkt aktuell installiert ist, führen Sie den folgenden Befehl aus:

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

2.4.3 Deinstallation von MapForce Server

Um MapForce Server zu deinstallieren, führen Sie den folgenden Befehl aus:

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

Um andere Pakete zu deinstallieren, verwenden Sie denselben Befehl wie oben und stellen Sie das zu entfernende Paket als Argument bereit. z.B:

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

2.5 macOS

2.5.1 Installation von MapForce Server

Systemanforderungen

▶ macOS

macOS 10.13 oder neuer

Voraussetzungen

- Führen Sie die Installation als Benutzer mit Admin-Rechten (Root)-Berechtigung durch.

FlowForce Server-Integration

Sie können MapForce Server entweder als Standalone-Produkt oder im Rahmen einer FlowForce Server-Installation installieren. Um MapForce Server unter Verwaltung von FlowForce Server zu installieren, aktivieren Sie die entsprechende Option im FlowForce-Installationsassistenten. Das FlowForce Server-Installationsprogramm enthält das MapForce Server-Installationsprogramm und das FlowForce Server Advanced Edition-Installationsprogramm enthält das MapForce Server Advanced Edition-Installationsprogramm. Sowohl das FlowForce Server- als auch das MapForce Server-Installationsprogramm kann vom Altova Download Center (<https://www.altova.com/de/download>) heruntergeladen werden.

Wenn Sie MapForce Server zusammen mit FlowForce Server installieren, sollten Sie zuerst FlowForce Server installieren. Beachten Sie andernfalls die "Aufgaben nach der Installation" weiter unten.

Installation von MapForce Server

1. Laden Sie die .dmg (Disk Image)-Datei von der Altova Website (<https://www.altova.com/de/download.html>) in ein lokales Verzeichnis herunter und klicken Sie darauf, um sie zu öffnen. Dadurch wird der Installer als neues Laufwerk auf Ihrem Computer angezeigt.
2. Doppelklicken Sie auf die Paketdatei (.pkg) und befolgen Sie die Anweisungen des Installationsassistenten, in dem Sie auch die Lizenzvereinbarung akzeptieren müssen, bevor Sie mit der Installation fortfahren können. Nach Fertigstellung des Assistenten steht das MapForce Server-Symbol in den Applikationen zur Verfügung. Das Paket wird im folgenden Ordner installiert: **/usr/local/Altova/MapForceServer2021/** .
3. Wenn Sie Altova License Server noch nicht installiert haben, oder ihn auf die neueste Version aktualisieren möchten, doppelklicken Sie auf die entsprechende Paketdatei (.pkg) und befolgen Sie die Anweisungen auf dem Bildschirm.

Lizenzierung von MapForce Server

1. Stellen Sie sicher, dass die neueste Version von Altova LicenseServer (<https://www.altova.com/de/licenseserver>) installiert ist und entweder auf dem lokalen Rechner oder im Netzwerk ausgeführt wird.
2. Registrieren Sie MapForce Server auf Altova LicenseServer. Dazu benötigen Sie entweder den Host-Namen oder die IP-Adresse des Rechners, auf dem LicenseServer ausgeführt wird. Wenn

LicenseServer z.B auf 127.0.0.1 läuft, können Sie Ihr Produkt folgendermaßen registrieren:

```
sudo mapforceserver licenseserver 127.0.0.1
```

3. Melden Sie sich auf der Verwaltungsoberfläche von LicenseServer an und weisen Sie die Lizenz dem Rechner zu, auf dem MapForce Server ausgeführt wird.

Nähere Informationen dazu finden Sie in der Dokumentation zu Altova LicenseServer (<https://www.altova.com/de/documentation>).

Aufgaben nach der Installation

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 /usr/local/Altova/MapForceServer2019/etc/*.tool /usr/local/Altova/FlowForceServer2019/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.

2.5.2 Deinstallation von MapForce Server

So deinstallieren Sie MapForce Server:

1. Klicken Sie unter **Applikationen** mit der rechten Maustaste auf das MapForce Server-Symbol wählen Sie den Befehl **Move to Trash**. Beachten Sie, dass Ihr MapForce Server-Installationsverzeichnis und die Konfigurationsdaten dadurch nicht entfernt werden.
2. Wenn Sie auch das MapForce Server-Installationsverzeichnis entfernen möchten, führen Sie den folgenden Befehl aus:

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

Anmerkung: Mit den obigen Schritten werden die Konfigurationsdaten nicht aus `/var/Altova/MapForceServer2021` entfernt. Einmal entfernte Konfigurationsdaten können später nicht mehr wiederhergestellt werden.

2.6 Windows

2.6.1 Installation von MapForce Server

Systemanforderungen

- Windows 7 SP1 mit Plattformupdate, Windows 8, Windows 10
- Windows Server 2008 R2 SP1 mit Plattformupdate oder höher

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.

Installation von MapForce Server

Sie können MapForce Server entweder als Standalone-Produkt oder im Rahmen einer FlowForce Server-Installation installieren. Um MapForce Server unter Verwaltung von FlowForce Server zu installieren, aktivieren Sie die entsprechende Option im FlowForce-Installationsassistenten. Das FlowForce Server-Installationsprogramm enthält das MapForce Server-Installationsprogramm und das FlowForce Server Advanced Edition-Installationsprogramm enthält das MapForce Server Advanced Edition-Installationsprogramm. Sowohl das FlowForce Server- als auch das MapForce Server-Installationsprogramm kann vom Altova Download Center (<https://www.altova.com/de/download>) heruntergeladen werden.

So installieren Sie MapForce Server als Standalone-Produkt:

1. Laden Sie die Installationsdatei vom Altova Download Center (<https://www.altova.com/de/download>) herunter und führen Sie sie aus.
2. Wählen Sie optional die Installationssprache aus dem Feld links unten im Assistenten aus und klicken Sie auf **Weiter**. Sie können die Sprache später über die Befehlszeile ändern.
3. Wählen Sie eine der folgenden Methoden:
 - a. 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 von Altova LicenseServer.
 - b. Wenn Sie Altova LicenseServer noch nicht installiert haben und Altova LicenseServer auf einem anderen Rechner installieren möchten, deaktivieren Sie die Option **Altova LicenseServer auf diesem Rechner installieren** und wählen Sie anschließend **Später registrieren**. In diesem Fall müssen Sie Altova LicenseServer separat installieren und MapForce Server später registrieren.
 - c. Wenn Altova LicenseServer auf Ihrem Rechner bereits installiert wurde, aber eine niedrigere Versionsnummer als die im Installationsassistenten angegebene hat, belassen Sie die Standardeinstellungen unverändert. In diesem Fall aktualisiert der Installationsassistent Ihre LicenseServer Version automatisch auf die im Dialogfeld angegebene. Beachten Sie, dass die vorhandene Registrierung und die vorhandenen Lizenzierungsdaten nach dem Upgrade erhalten bleiben.
 - d. Wenn Altova LicenseServer bereits auf Ihrem Rechner oder in Ihrem Netzwerk installiert wurde und

dieselbe Versionsnummer wie diejenige im Assistenten hat, gehen Sie folgendermaßen vor:

- i. Deaktivieren Sie das Kontrollkästchen **Altova LicenseServer auf diesem Rechner installieren**.
 - ii. Wählen Sie unter **Dieses Produkt registrieren auf** die Altova LicenseServer-Instanz, auf der Sie MapForce Server registrieren möchten oder wählen Sie **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.
4. Klicken Sie auf **Weiter**.
 5. Befolgen Sie die Anweisungen des Assistenten, um die Installation abzuschließen.

Lizenzierung von MapForce Server

1. Stellen Sie sicher, dass die neueste Version von Altova LicenseServer (<https://www.altova.com/de/licenseserver>) installiert ist und entweder auf dem lokalen Rechner oder im Netzwerk ausgeführt wird.
2. Registrieren Sie MapForce Server auf Altova LicenseServer, falls Sie das noch nicht bei der Installation getan haben. Dazu benötigen Sie entweder den Host-Namen oder die IP-Adresse des Rechners, auf dem LicenseServer ausgeführt wird. Wenn LicenseServer z.B auf 127.0.0.1 läuft, können Sie Ihr Produkt folgendermaßen registrieren:

```
MapForceServer.exe licenseserver 127.0.0.1
```

3. Melden Sie sich auf der Verwaltungsoberfläche von LicenseServer an und weisen Sie die Lizenz dem Rechner zu, auf dem MapForce Server ausgeführt wird.

Nähere Informationen dazu finden Sie in der Dokumentation zu Altova LicenseServer (<https://www.altova.com/de/documentation>).

2.6.2 Deinstallation von MapForce Server

Deinstallieren von MapForce Server

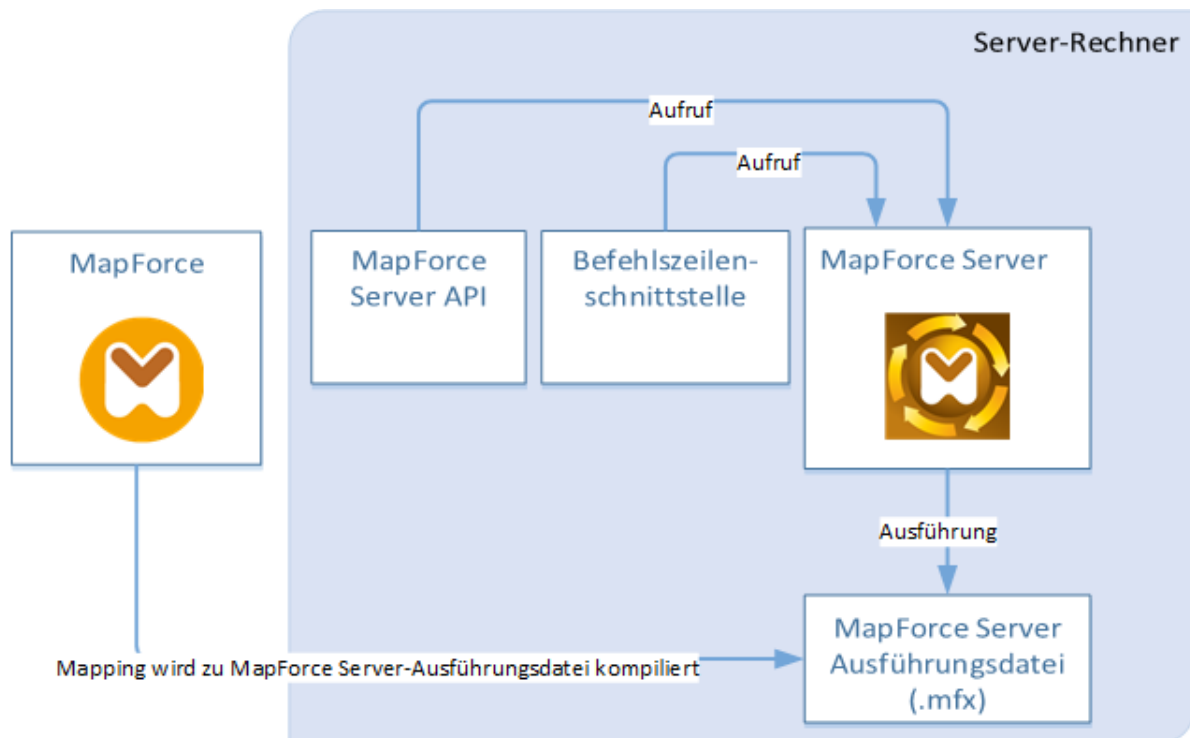
1. Klicken Sie mit der rechten Maustaste auf die Windows-Schaltfläche "Start" und wählen Sie **Einstellungen**.
2. Öffnen Sie die Systemsteuerung (Geben Sie die ersten Buchstaben davon ein und klicken Sie auf den Vorschlag).
3. Klicken Sie unter **Programme** auf **Programm deinstallieren**.
4. Wählen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

3 Ausführen von Mappings

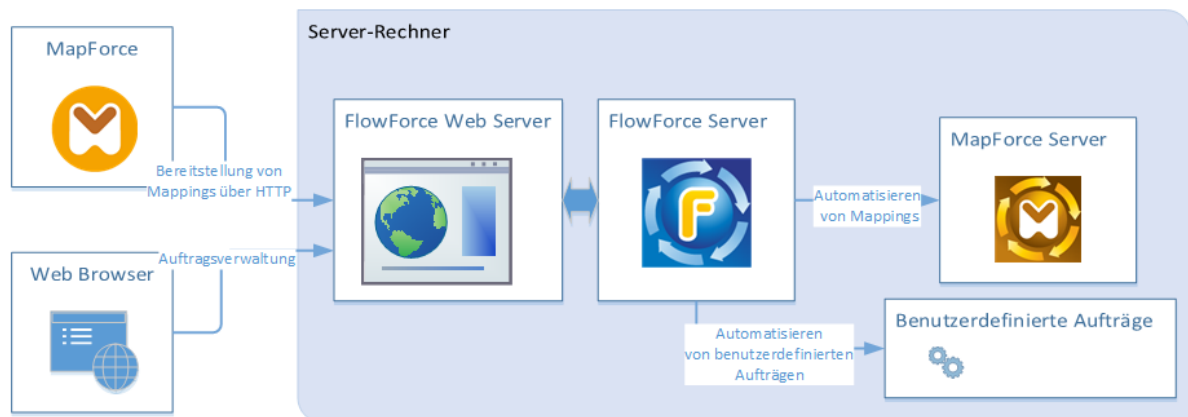
Die Aufgabe von MapForce Server ist es, mit Altova MapForce (<https://www.altova.com/de/mapforce.html>) erstellte Datenmappings auszuführen.

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](#)⁷) 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](#)⁷). 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](#)¹⁹.
5. Rufen Sie den Befehl "run" der [Befehlszeilenschnittstelle](#)³⁰ oder die entsprechende Methode der MapForce Server API (siehe [MapForce Server API](#)⁴⁶) 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 https://www.altova.com/manual/de/FlowForceServer/fs_example_mapforce_mapping.htm). 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](#)¹⁹.

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:

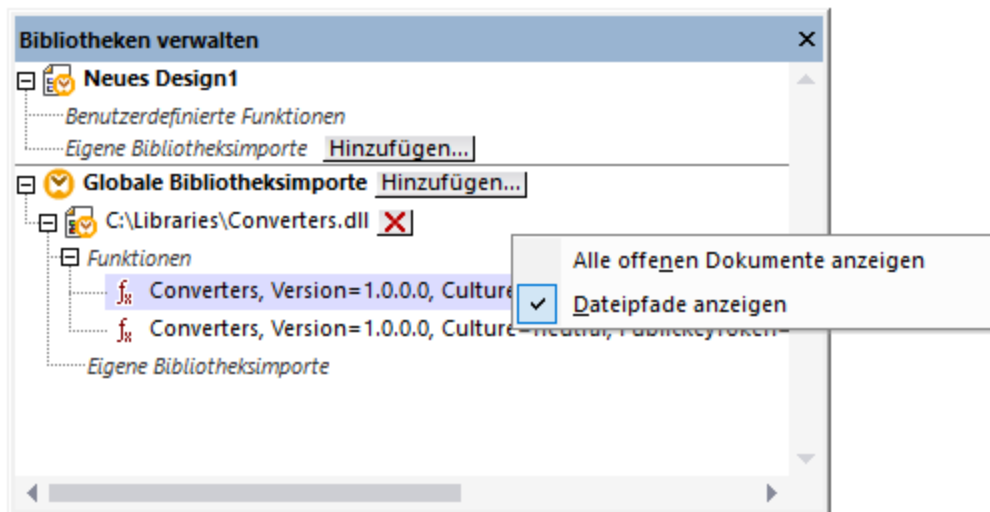
- Auf dem Quellrechner benötigen Sie zur Erstellung des Mappings und für die Bereitstellung des Mappings auf einem Zielrechner die MapForce Enterprise oder Professional Edition.
- Sowohl MapForce Server als auch FlowForce Server müssen auf dem Zielrechner lizenziert sein. Die Aufgabe von MapForce Server 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](#)¹⁰⁵).
- 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 in der FlowForce Server-Dokumentation im Kapitel "Bereitstellen eines Auftrags aus Web-Dienst" (https://www.altova.com/manual/de/FlowForceServer/FlowForceServerAdvanced/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](#) ²⁵.

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 übertragbare Referenzen auf Dateien, Ordner oder Datenbanken. Wenn Pfade 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, empfiehlt es sich, diese als globale Ressource zu definieren. Auf diese Art müssen Sie sich den Dateipfad nicht merken, da Sie die entsprechende globale Ressource stattdessen über das Dialogfeld "Datei öffnen" auswählen können. Dies hat auch den Vorteil, dass Sie den Pfad nur an einer Stelle ändern müssen, falls sich der Dateipfad ändern sollte.

Ein typisches Beispiel für die Verwendung globaler Ressourcen ist die einmalige Definition einer Datenbankverbindung, die dann in allen Altova-Applikationen, die globale Ressourcen unterstützen, wiederverwendet wird. So können Sie etwa auf dem Rechner, auf dem das MapForce Mapping erstellt wurde, eine Datenbankverbindung erstellen und dieselbe Verbindung dann auf dem Rechner, auf dem MapForce Server das Mapping ausführt, wiederverwenden (Eventuell muss dafür in einigen Fällen auf beiden Rechnern dieselbe Datenbank-Client-Software installiert sein).

Sie können optional mehrere Varianten derselben globalen Ressource (sogenannte "Konfigurationen") erstellen. Auf diese Art können Sie Datei- oder Ordnerpfade (oder sogar Datenbanken) ganz nach Bedarf einfach wechseln. So könnten Sie etwa eine "Datenbank"-Ressource mit zwei Konfigurationen erstellen: "Entwicklung" und "Produktion". In MapForce Server könnten Sie Daten dann z.B. ganz einfach entweder aus der Entwicklungs- oder der Produktionsdatenbank abrufen, indem Sie die gewünschte Konfiguration zur Mapping-Laufzeit als Befehlszeilenparameter angeben.

Globale Ressourcen können über die folgenden Altova Desktop-Applikationen erstellt werden: Altova Authentic, DatabaseSpy, MobileTogether Designer, MapForce, StyleVision und XMLSpy. Serverseitig können globale Ressourcen von den folgenden Altova-Server-Applikationen verwendet werden: 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 XML-Datei für die globalen Ressourcen. Auf dem Rechner, auf dem MapForce installiert ist, hat die Datei den Namen **GlobalResources.xml** und 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 namens "Default". Sie können aber bei Bedarf zusätzliche Konfigurationen definieren. Falls Sie dies tun, so müssen Sie anstelle von "Default" den Namen der gewünschten Konfiguration definieren.

Der Pfad der globalen Ressourcendatei 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.

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

```
C:\Program Files (x86)\Altova\MapForceServer2021\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 XML-Datei 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](#)⁴⁶.

3.3 Anmeldeinformationen

Mit Hilfe von Anmeldeinformationsobjekten können Authentifizierungsdaten (wie Benutzernamen und Passwörter) zwischen verschiedenen Mapping-Ausführungsumgebungen auf sichere Art portabel gemacht werden. Anmeldeinformationen eignen sich für die Verwendung in Mappings, für die eine einfache HTTP-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 Befehlszeilenooptionen 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
```

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();
```

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](#) ⁴⁶.

3.4 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:

- [assignlicense](#)³³: (nur Windows) Lädt eine Lizenz in LicenseServer hoch und weist sie MapForce Server zu.
- [exportresourcestrings](#)³⁴: Exportiert alle Applikationsressourcenstrings in eine XML-Datei.
- [help](#)³⁶: Zeigt Informationen zu dem im Argument angegebenen Befehl - oder wenn kein Befehl angegeben ist, zu allen Befehlen - an.
- [licenseserver](#)³⁷: Registriert MapForce Server auf einem LicenseServer im lokalen Netzwerk.
- [run](#)³⁹: Führt eine mit MapForce kompilierte Mapping-Ausführungsdatei (.mfx) aus.
- [setdeflang](#)⁴³: Definiert die Standardsprache von MapForce Server.
- [verifylicense](#)⁴⁴: (Nur Windows) Überprüft, ob der aktuelle MapForce Server lizenziert ist; optional dazu können Sie überprüfen, ob ein bestimmter Lizenzschlüssel zugewiesen ist.
- [version](#)⁴⁵: 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/MapForceServer2021/bin/mapforceserver
<i>macOS</i>	/usr/local/Altova/MapForceServer2021/bin/mapforceserver
<i>Windows</i>	C:\Programme\Altova\MapForceServer2021\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>	<code>/opt/Altova/MapForceServer2021/bin/mapforceserver help</code>
<i>macOS</i>	<code>/usr/local/Altova/MapForceServer2021/bin/mapforceserver help</code>
<i>Windows</i>	<code>C:\Programme (x86)\Altova\MapForceServer2021\bin\MapForceServer.exe help</code>

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/MapForceServer2021/bin</code>
<i>macOS</i>	<code>cd /usr/local/Altova/MapForceServer2021/bin</code>
<i>Windows</i>	<code>cd "C:\Programme (x86)\Altova\MapForceServer2021\bin\MapForceServer.exe"</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 assignlicense (nur Windows)

Syntax und Beschreibung

Der Befehl `assignlicense` steht nur unter Windows (nicht aber auf Linux- oder Mac-Systemen) zur Verfügung. Mit diesem Befehl 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/AltovaLicenseServer/>).

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.

4.2 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\MapForceServer2021\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\MapForceServer2021\bin`. Ihre DLL-Datei wird einen Namen in der Form `MapForceServer2021_lc.dll` haben. Der `lc` Teil des Namens enthält den Sprachcode. So steht z.B. in `MapForceServer2021_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 Sprachcode, 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`.

4.3 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.

4.4 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.
- Um eine Evaluierungslizenz für MapForce Server anzufordern, geben Sie die entsprechenden Werte für die Optionen `--company`, `--email` und `--name` an. Um eine Evaluierungslizenz für ein zusätzliches Altova-Produkt (d.h. zusätzlich zu MapForce Server) anzufordern, fügen Sie die Option `--request-eval-license` mit einem Wert, der dem Name dieses Produkts entspricht, hinzu. *Nähere Informationen dazu finden Sie in der Liste weiter unten.*

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/AltovaLicenseServer/>).

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.

4.5 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/documentation#mapforce>). Je nach Mapping-Design müssen auch andere Voraussetzungen erfüllt werden, siehe [Vorbereiten von Mappings für die Server-Ausführung](#)¹⁹.

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 `MapForceServer run [options] MfxFile`

Linux `mapforceserver run [options] MfxFile`

Mac `mapforceserver run [options] MfxFile`

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 ¹²⁷ . Form: <code>--catalog=FILE</code>
<code>--cert</code>	<code>-certificatespath</code>	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: <code>--certificatespath=DIRECTORY</code> Siehe auch Verwaltung digitaler Zertifikate ¹⁰⁵ .
<code>--cred</code>	<code>--credential=KEY:VALUE</code>	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: <code>name1=value1&name2=value2</code> Hierbei ist name einer der folgenden: <ul style="list-style-type: none"> • username • password Wenn Sie mehrere durch das Kaufmännische Und-Zeichen getrennte Name-Wert-Paare verwenden, setzen Sie das KEY:VALUE-Paar in Anführungszeichen, siehe Anmeldeinformationen ²⁷ .
<code>--gc</code>	<code>--globalresourceconfig</code>	Diese Option ist anwendbar, wenn im Mapping globale Ressourcen ²⁵ verwendet werden. Sie definiert den Namen der Konfiguration für globale Ressourcen (Alias). Diese Option muss zusammen mit der Option <code>--globalresourcefile</code> verwendet werden. Form: <code>--gc=VALUE</code>
<code>--gr</code>	<code>--globalresourcefile</code>	Diese Option ist anwendbar, wenn im Mapping globale Ressourcen ²⁵ verwendet werden. Sie definiert den Pfad zur Definitionsdatei für globale Ressourcen. Diese Option muss zusammen mit der Option <code>--globalresourceconfig</code> verwendet werden. Form: <code>--gr=FILE</code> .
<code>--l</code>	<code>--lang</code>	Die Sprache, in der Meldungen angezeigt werden. Form: <code>--lang=VALUE</code> (en,de,ja,es)
<code>--p</code>	<code>--param</code>	Diese Option ist anwendbar, wenn für das Mapping Input-Parameter verwendet werden. Damit wird einem im Mapping definierten Parameter ein Wert zugewiesen. Form: <code>--param=ParamName:ParamValue</code> . Vor jedem Parameter muss die Option <code>--param</code> verwendet werden. Verwenden

		<p>Sie Anführungszeichen, wenn <i>ParamName</i> oder <i>ParamValue</i> ein Leerzeichen enthält. Zum Beispiel: -- p=company: "Nanonull Inc".</p> <p>Nähere Informationen zu Mappings, die Input-Parameter erhalten, finden Sie in der MapForce-Dokumentation (https://www.altova.com/de/documentation#mapforce).</p>
	--taxonomy-package	<p>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.</p> <p>Form: --taxonomy-package=FILE</p>
	--taxonomy-packages-config-file	<p>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:</p> <p>\Benutzer\<benutzername>\Documents\Altova.</benutzername></p> <p>Form: --taxonomy-packages-config-file=FILE</p>

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\.**
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\MapForceServer2021\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](#)¹⁹.

4.6 setdeflang

Syntax und Beschreibung

Der Befehl `setdeflang` (Kurzform ist `sdl`) 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` (`sdl`):

```
mapforceserver sdl 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

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

4.7 verifylicense (nur Windows)

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. Dieser Befehl wird nur auf Windows-Systemen, nicht aber unter Linux oder Max unterstützt.

```
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/AltovaLicenseServer/>).

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, --license-key = Value
```

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

4.8 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](#) ⁴⁷
- [COM-Schnittstelle](#) ⁵⁴
- [Java-Schnittstelle](#) ⁶³

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

- [API-Referenz \(COM, .NET\)](#) ⁷⁴
- [API-Referenz \(Java\)](#) ⁹²

5.1 .NET-Schnittstelle

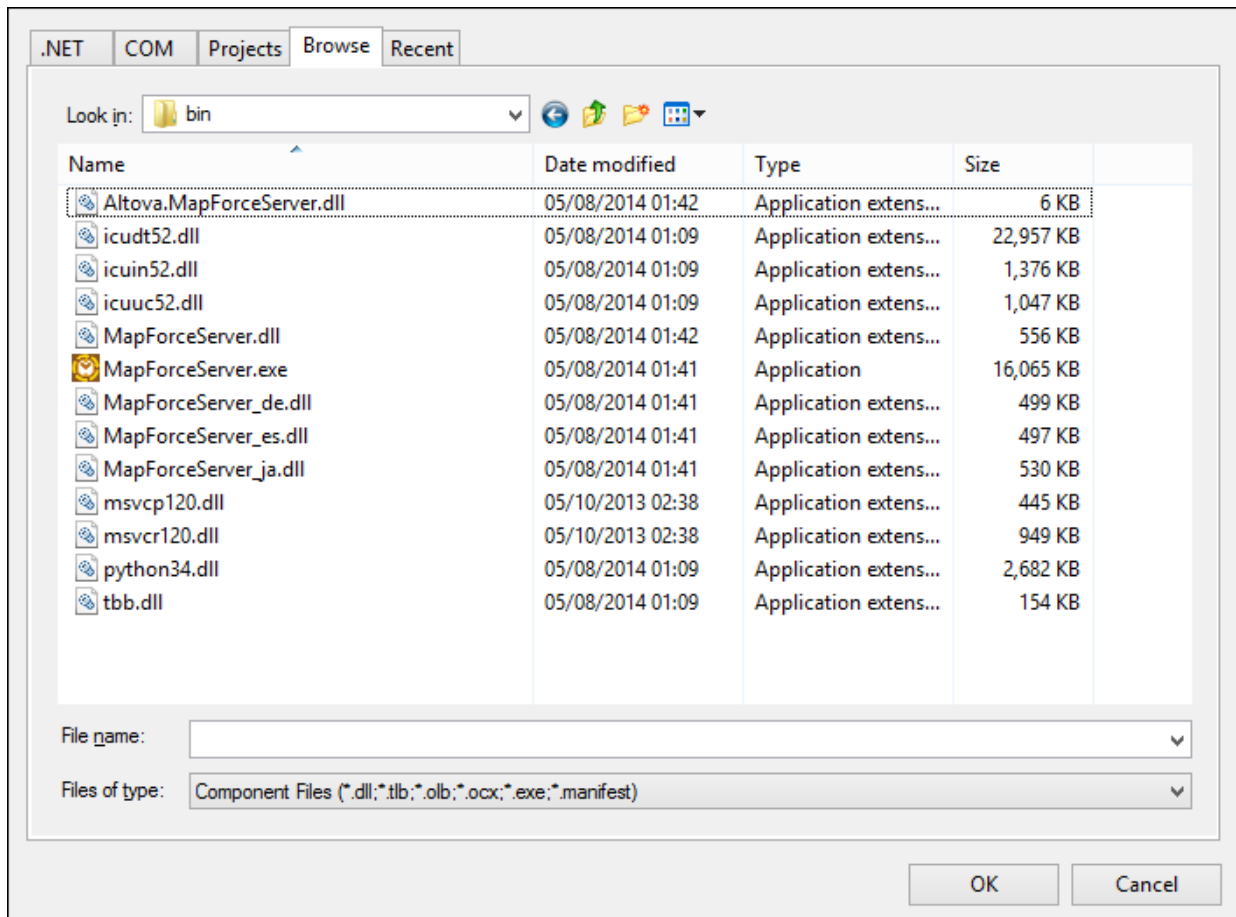
Die .NET-Schnittstelle ist als Wrapper rund um die COM-Schnittstelle gebaut. Sie dient als primäre von Altova signierte Interop-Assembly und verwendet den Namespace `Altova.MapForceServer`. MapForce Server wird bei der Installation automatisch als COM-Serverobjekt registriert und muss daher nicht manuell 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: 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\MapForceServer2021\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](#)⁴⁷). 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\.**

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
                // 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\MapForceServer2021\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](#)⁴⁷). 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\MapForce2021\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
            'objMFS.ServerPath = "C:\Program Files (x86)
            \Altova\MapForceServer2021\bin\MapForceServer.exe"
            'objMFS.ServerPath = "C:\Program
            Files\Altova\MapForceServer2021\bin\MapForceServer.exe"

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

            '-----
            -----

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

```

```
this program as administrator.
    System.Console.WriteLine(objMFS.LastExecutionMessage)
End If

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

'-----
-----
'an example with parameterized input
' the default of 'lower=5' gets changed to the value '10'
' mfx reads file Temperatures.xml and writes its output to
Temperatures_out.xml.
System.Console.WriteLine(vbCrLf & "Executing ClassifyTemperatures.mfx with
parameter 'lower' set to '10' ...")
objMFS.AddParameter("lower", "10")
If (objMFS.Run("ClassifyTemperatures.mfx")) Then
    System.Console.WriteLine("File Temperatures_out.xml has been written
successfully.")
Else
    'execution failed. maybe no write permissions in working directory? Run
this program as administrator.
    System.Console.WriteLine(objMFS.LastExecutionMessage)
End If

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

End Sub

End Module
```

5.2 COM-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\MapForceServer2021\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](#)⁵⁴).

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\MapForce2021\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
    // pMFS.ServerPath = "C:\\Program Files (x86)\\Altova\\MapForceServer2021\\bin\\
\\MapForceServer.exe";
    // pMFS.ServerPath = "C:\\Program Files\\Altova\\MapForceServer2021\\bin\\
\\MapForceServer.exe";

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

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

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

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

```



```

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

CoUninitialize();

return 0;
}

```

5.2.2 VBScript-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\MapForceServer2021\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](#)⁵⁴).

```

Option Explicit

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

'Create the MapForce Server object
Dim objMFS
' Since we load a COM-DLL we need care about the process architecture
On Error Resume Next ' ignore any COM errors avoiding uncontrolled script termination
Dim WshShell
Dim WshProcEnv
Set WshShell = CreateObject("WScript.Shell")
Set WshProcEnv = WshShell.Environment("Process")
Dim process_architecture
process_architecture= WshProcEnv("PROCESSOR_ARCHITECTURE")
If process_architecture = "x86" Then
    Set objMFS = WScript.GetObject( "", "MapForce.Server" )
    If Err.Number <> 0 then

```

```

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

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

' Default path to the MapForce Server executable is the installation path (same dir with
the MapForceServer.dll)
' In case you moved the binaries on the disk, you need to explicitly set the path to the
.exe file
' objMFS.ServerPath = "C:\Program Files (x86)
\Altova\MapForceServer2021\bin\MapForceServer.exe"

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

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

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

REM -----
REM run an example with input and output paths stored inside the MFX file
' the path to the mfx file can be absolute or relative to the working directory

```

```

' depends on existence of file AltovaTools.xml in working direcorey
' 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:

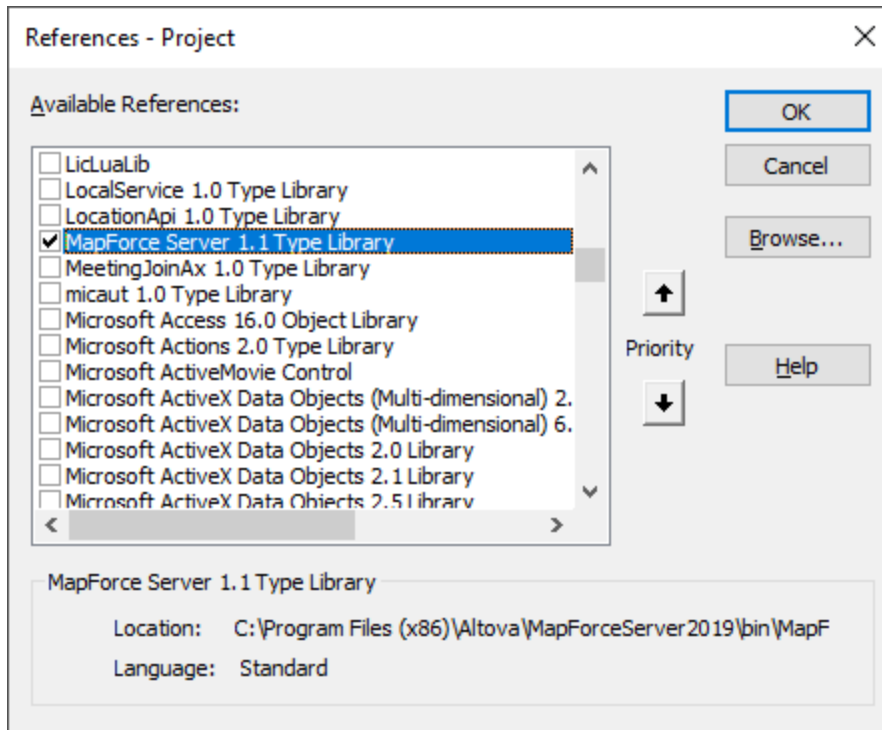
1. Microsoft Office und MapForce Server müssen auf demselben Rechner installiert sein.
2. Die Architektur von MapForce Server (32-Bit oder 64-Bit) muss mit der von Microsoft Office übereinstimmen. Wenn Sie z.B. VBA auf Microsoft Office 32-Bit ausführen, stellen Sie sicher, dass Sie die MapForce Server 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

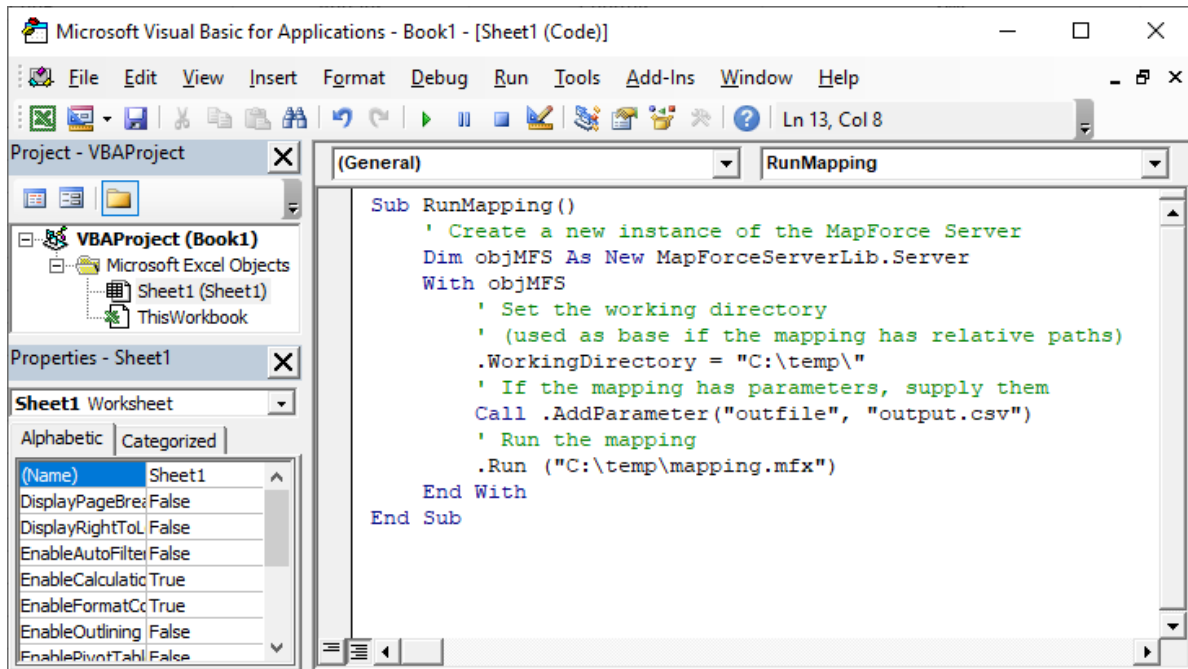
Nachdem Sie in Ihrem VBA-Projekt eine Referenz zur MapForce Server-Bibliothek hinzugefügt haben, können Sie den VBA-Code im Code-Editor-Fenster eingeben. Mit dem folgenden Beispielcode wird z.B. MapForce Server aufgerufen und eine ausführbare Mapping-Datei (**mapping.mfx**), die einen Input-Parameter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



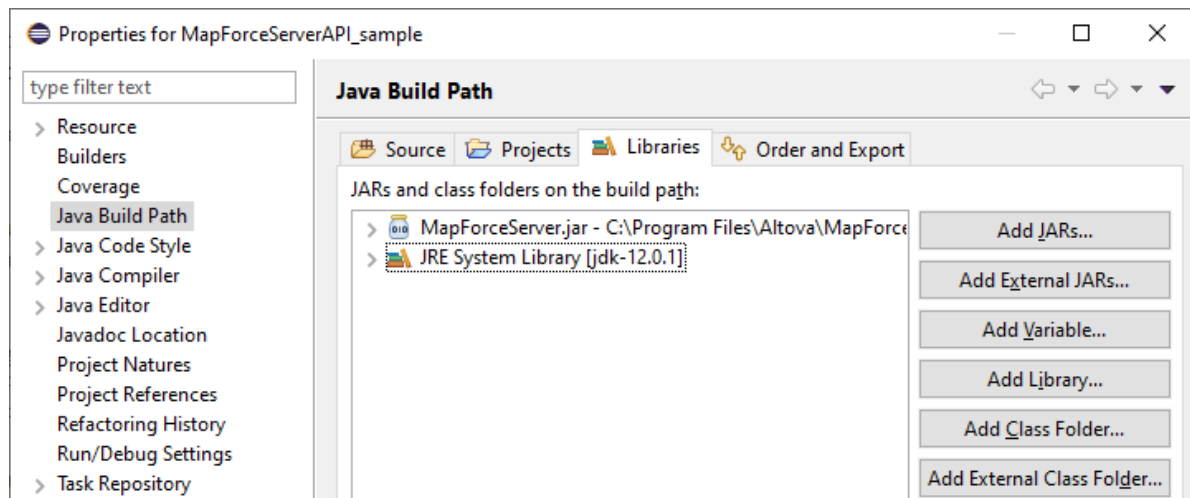
5.3 Java-Schnittstelle

Um die MapForce Server API von Java-Code aus aufrufen zu können, muss die folgende Referenz zur **MapForceServer.jar**-Bibliothek zur Java-Classpath-Datei hinzugefügt werden. Die **MapForceServer.jar**-Datei befindet sich im Ordner **bin** des MapForce Server Installationsordners. Sie können die Datei entweder über ihren Originalpfad referenzieren oder sie in einen anderen Ordner kopieren, falls Sie Ihr Projekt anders einrichten möchten.

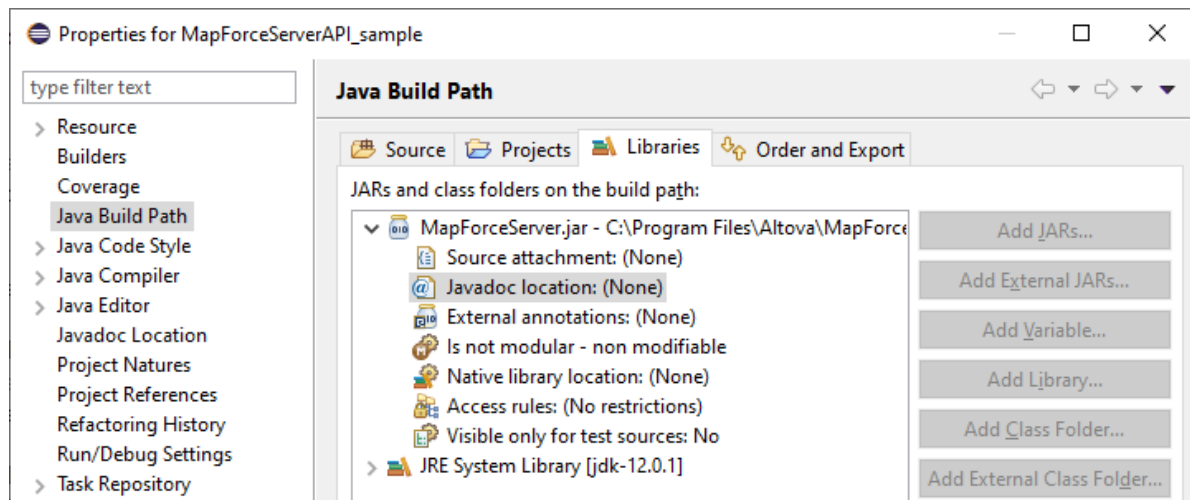
Wenn Sie Eclipse als Java-Entwicklungsumgebung verwenden, können Sie die erforderliche Bibliotheksreferenz zum CLASSPATH hinzufügen, indem Sie die Eigenschaft des Java-Projekts wie unten gezeigt bearbeiten. Sie können optional auch Dokumentation im JavaDoc-Format zur .jar-Bibliothek hinzufügen. Die JavaDoc befindet sich im Ordner **bin** des MapForce-Installationsordners. In der Anleitung unten wird beschrieben, wie Sie die JavaDoc-Dokumentation von Eclipse aus sichtbar machen.

So fügen Sie eine Referenzen zur MapForce Server-Bibliothek hinzu:

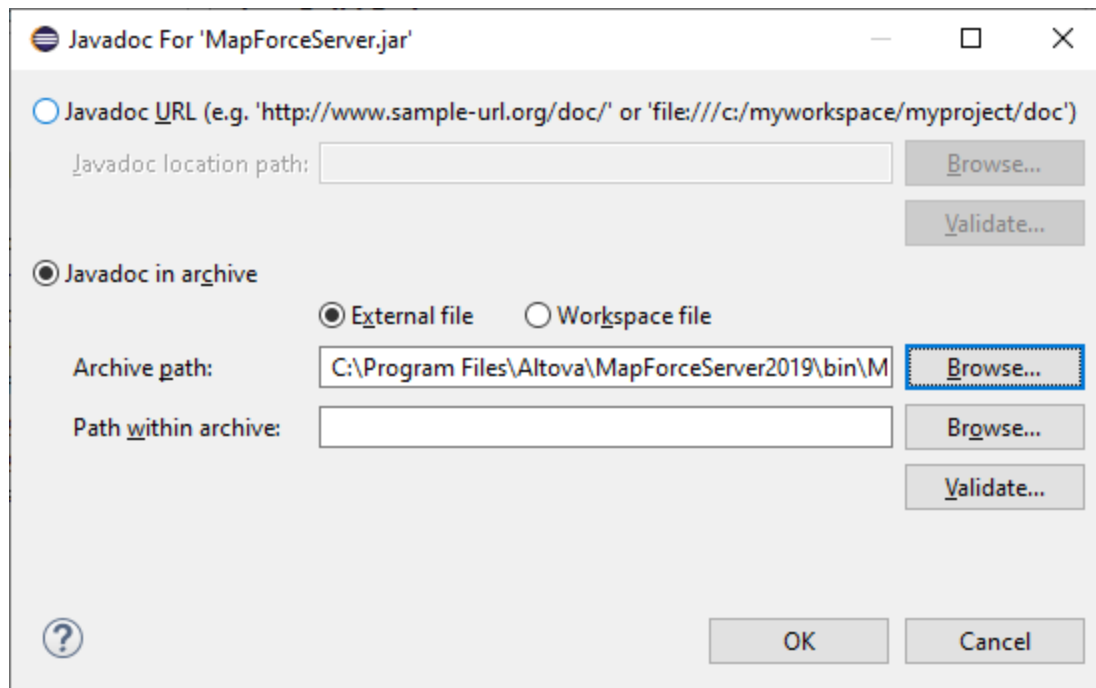
1. Klicken Sie bei geöffnetem Projekt in Eclipse im Menü **Project** auf **Properties**.
2. Klicken Sie auf **Java Build Path**.
3. Klicken Sie auf dem Register **Libraries** auf **Add External JARs** und navigieren Sie zur Datei `MapForceServer.jar` im MapForce Server Installationsordner.



4. Erweitern Sie optional den Eintrag **MapForceServer.jar** und doppelklicken Sie auf den Eintrag **Javadoc location: (None)**.



5. Vergewissern Sie sich, dass die Optionen **Javadoc in archive** und **External file** ausgewählt sind und navigieren Sie anschließend zur Datei `MapForceServer_JavaDoc.zip` im MapForce Server Installationsordner.



6. Klicken Sie auf OK.

Unten sehen Sie ein Beispiel dafür, wie die Eclipse `.classpath`-Datei aussehen könnte, wenn Sie die Dateien auf einer 64-Bit MapForce Server-Version auf einem 64-Bit-Windows-Betriebssystem vom Originalinstallationsordner aus 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/MapForceServer2021/bin/MapForceServer.jar">
    <attributes>
      <attribute name="javadoc_location" value="jar:file:/C:/Program%
20Files/Altova/MapForceServer2021/bin/MapForceServer_JavaDoc.zip!/" />
    </attributes>
  </classpathentry>
  <classpathentry kind="con" path="org.eclipse.jdt.launching.JRE_CONTAINER">
    <attributes>
      <attribute name="module" value="true" />
    </attributes>
  </classpathentry>
  <classpathentry kind="output" path="" />
</classpath>

```

5.3.1 Java-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\MapForceServer2021\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](#)⁶³).

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](#)¹⁹.

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/MapForceServer2021/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 (in diesem Beispiel **TokenizeString.mfx**). Bei Erfolg generiert das unten stehende Programm eine CSV-Datei im Arbeitsverzeichnis. Bei einem Fehler versucht das Programm die letzte von MapForce Server generierte Ausführungsmeldung zu drucken.

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

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

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

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

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

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

5.4 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#](#)⁴⁸, [C++](#)⁵⁴, [VB.NET](#)⁵¹, [VBScript](#)⁵⁷ oder [Java](#)⁶⁵.

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#](#)⁴⁸, [C++](#)⁵⁴, [VB.NET](#)⁵¹, [VBScript](#)⁵⁷ oder [Java](#)⁶⁵.

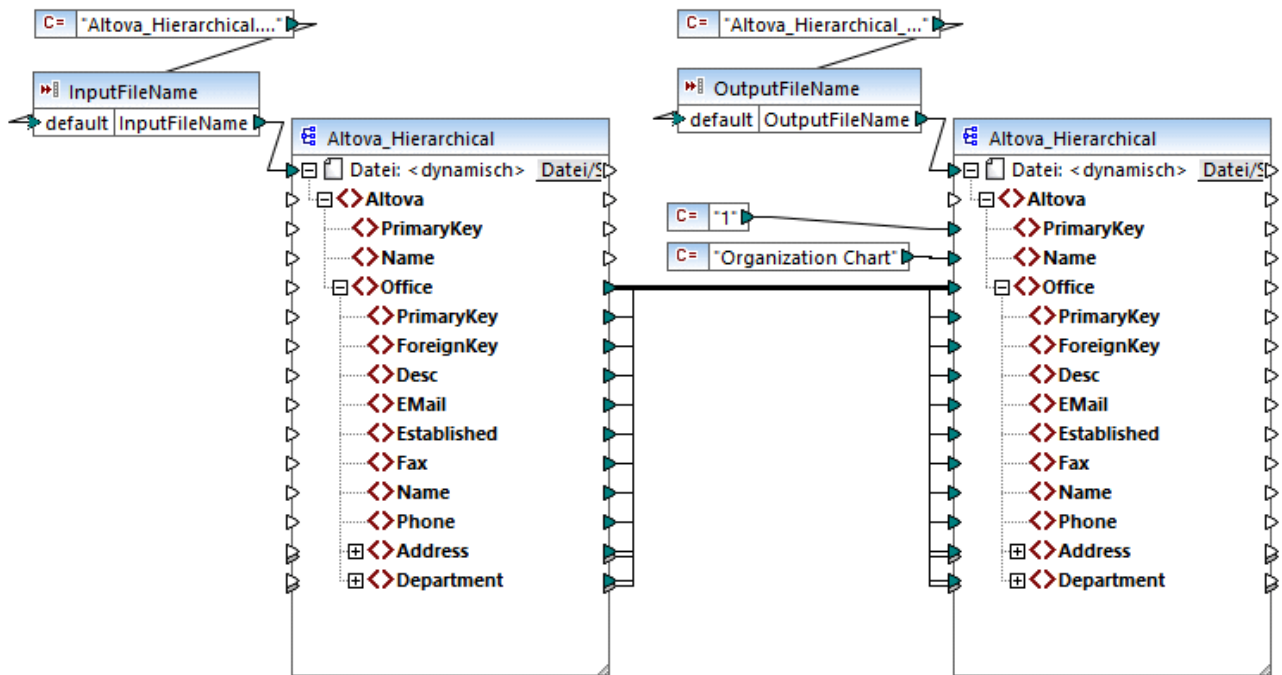
Vorbereiten von Mappings für die Server-Ausführung

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\\Dokumente\Altova\MapForce2021\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-Zieldatei kopiert werden. Nur die Felder **PrimaryKey** und **Name** der XML-Zieldatei werden mit Konstanten aus dem Mapping befüllt.

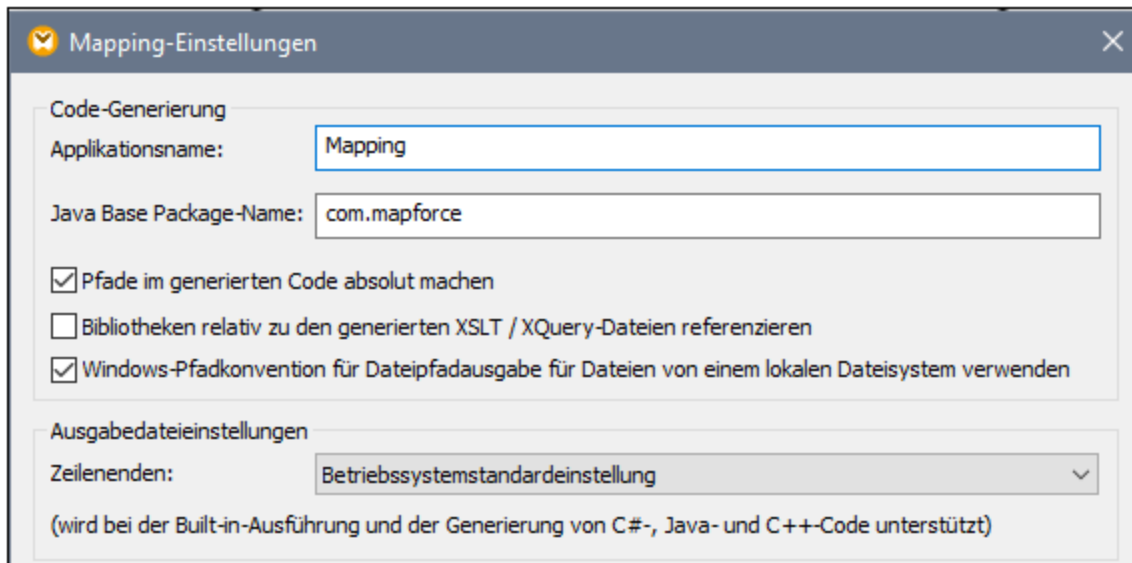


FileNamesAsParameters.mfd

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_Hierarhical.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](#)⁸³ (oder gegebenenfalls der Java-Methode [setWorkingDirectory](#)¹⁰⁴) 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](#)¹⁹.

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\\Dokumente\Altova\MapForce2021\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](#)¹⁹ 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](#)⁴⁴.

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](#)⁴⁷ beschrieben, und vergewissern Sie sich, dass MapForce Server eine gültige Lizenz hat, siehe auch CLI-Befehl [verifylicense](#)⁴⁴.

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](#)⁶³ beschrieben.
- MapForce Server eine gültige Lizenz hat, siehe auch CLI-Befehl [verifylicense](#)⁴⁴.

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 ⁷⁷	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 ⁷⁸	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 ⁷⁸	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 ⁷⁹	Schreibgeschützt. Gibt true zurück, wenn der MapForce Server-Prozessor eine ausführbare 64-Bit-Datei ist.
LastExecutionMessage ⁷⁹	Schreibgeschützt. Ruft die beim letzten Run -Befehl erhaltene Meldung ab.
MajorVersion ⁸⁰	Schreibgeschützt. Ruft die Hauptversion des Produkts als Anzahl von Jahren seit 1998 ab (z.B. "20" für Altova MapForce Server 2018).
MinorVersion ⁸⁰	Schreibgeschützt. Ruft die Nebenversion des Produkts ab (z.B. "2" für Altova MapForce Server 2018 r2).

Name	Beschreibung
ProductName ⁸¹	Schreibgeschützt. Ruft den Namen des Produkts ab (z.B. "Altova MapForce Server").
ProductNameAndVersion ⁸¹	Schreibgeschützt. Ruft den vollständigen Namen und die Version des Produkts ab (z.B. "Altova MapForce Server 2018 r2 sp1 (x64)").
ServerPath ⁸²	Ruft den Pfad der ausführbaren MapForce Server-Datei ab bzw. definiert ihn.
ServicePackVersion ⁸²	Schreibgeschützt. Ruft die Service Pack-Version des Produkts ab (z.B. "1" für Altova MapForce Server 2018 r2 sp1).
WorkingDirectory ⁸³	Ruft das aktuelle Verzeichnis für ausgeführte Aufträge ab bzw. definiert es (relative Pfade werden anhand des Arbeitsverzeichnisses aufgelöst).

Methoden

Name	Beschreibung
AddCredentialProperty ⁸³	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.
AddParameter ⁸⁴	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 ⁸⁵	Erstellt neue Anmeldeinformationen mit dem als Argument bereitgestellten Namen. Wenn Sie diese Methode aufrufen, müssen Sie mittels <code>AddCredentialProperty()</code> auch Eigenschaften zu dieser hinzufügen und die Anmeldeinformationen schließlich durch Aufruf von <code>EndCredential()</code> schließen.
ClearCredentialList ⁸⁶	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 ⁸⁶	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 ⁸⁷	Löscht die Liste der zuvor mit der Methode <code>AddParameter</code> definierten Parameter.
EndCredential ⁸⁷	Schließt ein zuvor mit der Methode <code>BeginCredential</code> definiertes Anmeldeinformationen-Objekt.
GetOutputParameter ⁸⁸	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 ⁸⁹	Führt eine MapForce Server-Ausführungsdatei (.mfx-Datei) aus. Gibt bei Erfolg <code>true</code> zurück, andernfalls <code>false</code> .
SetOption ⁸⁹	<p>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 ²⁵). Derzeit werden die folgenden Optionen unterstützt:</p> <ul style="list-style-type: none"> • 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.

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 ⁹¹	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](#) ⁵⁴
- [C#-Beispiel](#) ⁴⁸
- [VBScript-Beispiel](#) ⁵⁷
- [VB.NET-Beispiel](#) ⁵¹.

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* pRetVal);
```

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* pRetVal);
```

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* pRetVal);
```

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

5.5.1.1.10 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

5.5.1.1.11 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

5.5.1.1.12 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	<code>System.String</code>	Definiert den Namen der Anmeldeinformationen-Eigenschaft.
bstrValue	<code>System.String</code>	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	<i>String</i>	Definiert den Parameternamen.
bstrValue	<i>String</i>	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	<i>String</i>	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	<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)

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	<i>String</i>	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](#)²⁵). 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 globale 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	<code>String</code>	Definiert den Namen der zu setzenden Option.
bstrValue	<code>String</code>	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 ⁹⁵	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 ⁹⁵	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 ⁹⁷	Weist einem im Mapping definierten Parameter einen Wert zu.
beginCredential ⁹⁷	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 ⁹⁷	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 ⁹⁸	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 ⁹⁸	Löscht die Liste der zuvor mit der Methode <code>addParameter</code> definierten Parameter.
endCredential ⁹⁸	Schließt ein zuvor mit der Methode <code>beginCredential</code> definiertes Anmeldeinformationen-Objekt.
getAPIMajorVersion ⁹⁹	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 ⁹⁹	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 ⁹⁹	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 ¹⁰⁰	Ruft die beim letzten <code>run</code> -Befehl erhaltene Meldung ab.
getMajorVersion ¹⁰⁰	Ruft die Hauptversion des Produkts als Anzahl von Jahren seit 1998 ab (z.B. "20" für Altova MapForce Server 2018).
getMinorVersion ¹⁰⁰	Ruft die Nebenversion des Produkts ab (z.B. "2" für Altova MapForce Server 2018 r2).
getOutputParameter ¹⁰⁰	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 ¹⁰¹	Ruft den Namen des Produkts ab (z.B. "Altova MapForce Server").
getProductNameAndVersion ¹⁰¹	Ruft den vollständigen Namen und die Version des Produkts ab (z.B. "Altova MapForce Server 2018 r2 sp1 (x64)").
getServicePackVersion ¹⁰¹	Ruft die Service Pack-Version des Produkts ab bzw. definiert sie (z.B. "1" für Altova MapForce Server 2018 r2 sp1 (x64)).
is64bit ¹⁰²	Gibt true zurück, wenn der MapForce Server-Prozessor eine ausführbare 64-Bit-Datei ist.

Name	Beschreibung
run ¹⁰²	Führt eine MapForce Server-Ausführungsdatei (.mfx-Datei) aus. Gibt bei Erfolg true zurück, andernfalls false .
setOption ¹⁰³	<p>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 ²⁵). Derzeit werden die folgenden Optionen unterstützt:</p> <ul style="list-style-type: none"> • 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. <p>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.</p>
setServerPath ¹⁰⁴	Definiert den Pfad der ausführbaren MapForce Server-Datei.
setWorkingDirectory ¹⁰⁴	Definiert das aktuelle Verzeichnis für ausgeführte Aufträge (relative Pfade werden anhand des Arbeitsverzeichnisses aufgelöst).

Beispiele

Ein Beispiel zur Erstellung einer neuen Instanz von MapForceServer in Java finden Sie im [Java-Beispiel](#) ⁶⁵.

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	<code>String</code>	Der Name der Anmeldeinformationen-Eigenschaft (z.B. "username" oder "password").
arg1	<code>String</code>	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.MapForceServer();

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

5.6.1.1.12 `getLastExecutionMessage`

Ruft die beim letzten `run`-Befehl erhaltene Meldung ab.

Signatur

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

Allgemeine Signatur

5.6.1.1.13 `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

5.6.1.1.14 `getMinorVersion`

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

Signatur

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

Allgemeine Signatur

5.6.1.1.15 `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** *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

5.6.1.1.1.19 is64bit

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

Signatur

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

Allgemeine Signatur

5.6.1.1.1.20 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.

5.6.1.1.1.21 *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](#)²⁵). 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.

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	<code>String</code>	Definiert den Namen der zu setzenden Option.
arg1	<code>String</code>	Definiert den Wert der zu setzenden Option.

5.6.1.1.1.22 `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.

5.6.1.1.1.23 `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.

5.6.1.2 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](#)¹¹¹). Auf macOS-Systemen können Sie die entsprechende Operation in Keychain Access durchführen (siehe [Vertrauenswürdige Server-Zertifikate auf macOS](#)¹¹⁰). Eine Anleitung zu Linux-Systemen finden Sie unter [Vertrauenswürdige Server-Zertifikate unter Linux](#)¹⁰⁸.

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

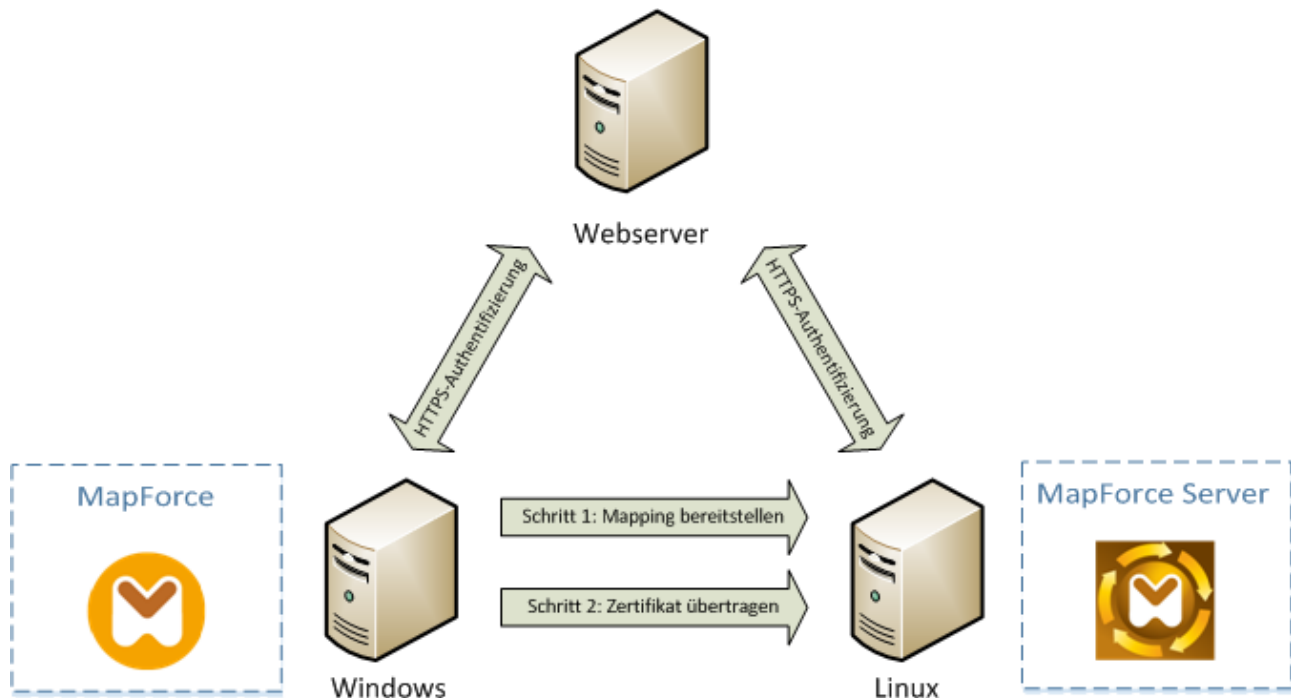
vorhaben, auf einen Webserver zuzugreifen, für den Client-Zertifikate erforderlich sind, müssen Sie möglicherweise den Administrator des Webserver bezüglich einer Client-Konfigurationsanweisung kontaktieren. So kann ein Webserver z.B. bei IIS (Internet Information Services) so konfiguriert werden, dass 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	<p>Unter Windows können Sie Zertifikate mit Hilfe des Zertifikat-Snap-in verwalten (siehe Aufrufen der Zertifikatspeicher unter Windows¹¹²).</p> <p>TLS-Unterstützung steht über den <i>Sicheren Kanal</i> (auch als SKanal bezeichnet) zur Verfügung.</p>
Linux	<p>Unter Linux können Sie Zertifikate mit Hilfe des OpenSSL (<code>openssl</code>) 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.</p> <p>TLS-Unterstützung steht über die OpenSSL-Bibliothek zur Verfügung (https://www.openssl.org/).</p>
macOS	<p>Auf macOS-Rechnern können Sie Zertifikate mit dem <i>Keychain Access Manager</i> unter Finder > Applications > Utilities verwalten.</p> <p>TLS-Unterstützung steht über die native <i>Secure Transport</i>-Bibliothek zur Verfügung</p>

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](#)¹²⁵). 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](#)¹¹⁵). Importieren Sie anschließend dasselbe Zertifikat in den Speicher **Aktueller Benutzer\Persönlich** auf dem Zielbetriebssystem (siehe [Client-Zertifikate unter Windows](#)¹²⁵).

Eine Anleitung zum Transferieren von Client-Zertifikaten auf die Linux- und macOS-Plattform finden Sie unter [Client-Zertifikate unter Linux](#)¹²¹ bzw. [Client-Zertifikate unter macOS](#)¹²³.

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 Webservers 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:

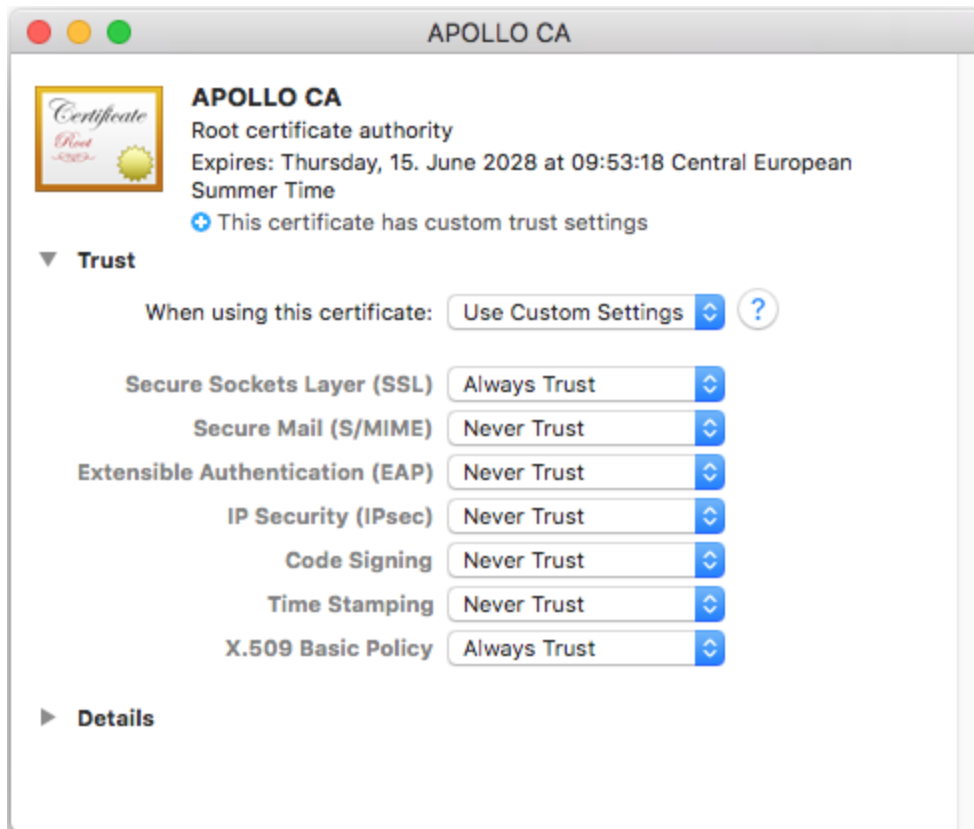
1. Klicken Sie unter **Options | Privacy & Security** auf **View Certificates**.
2. Klicken Sie auf dem Register **Authorities** auf **Import** und navigieren Sie zur zuvor erstellten Root-Zertifikatdatei.
3. Aktivieren Sie die Option **Trust this CA to identify websites**, wenn Sie gefragt werden.



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**.



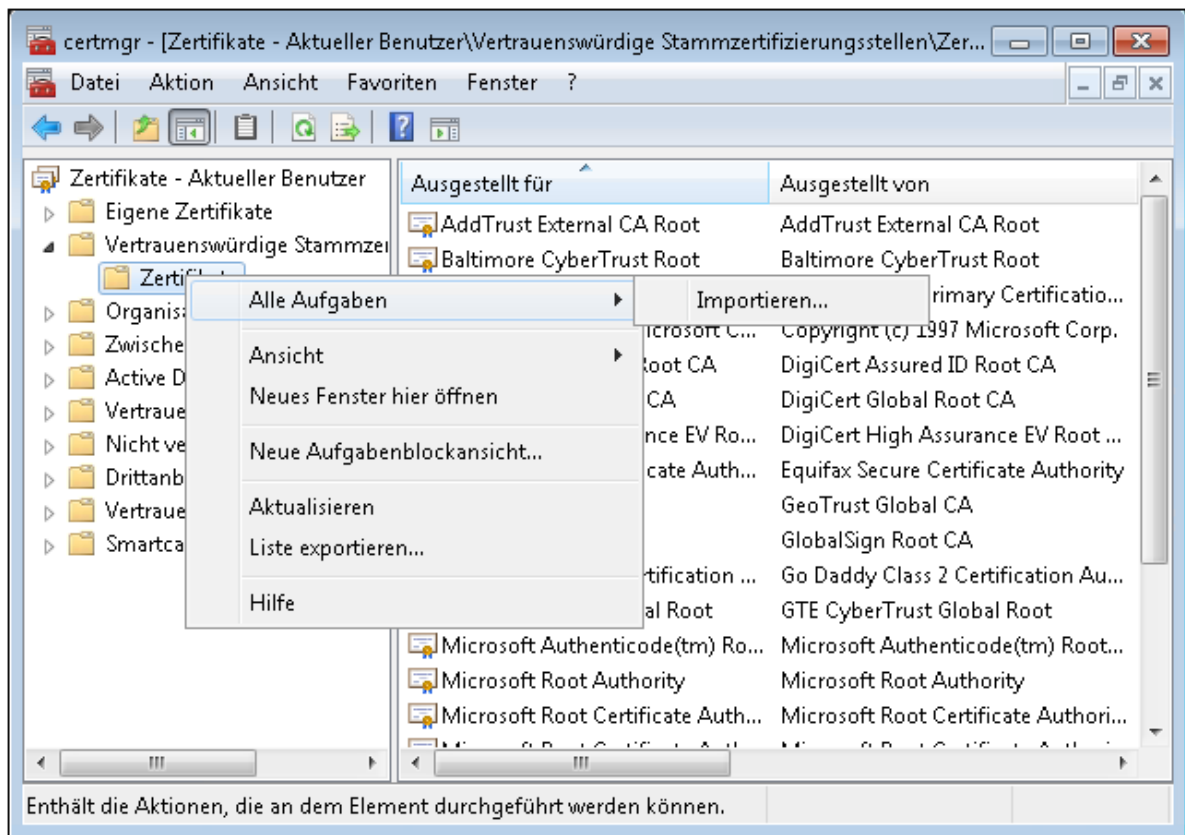
6.3 Definieren vertrauenswürdiger Zertifikate unter Windows

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](#)¹¹².

Führen Sie die folgenden Schritte nur durch, wenn Sie sicher sind, dass das Webserver-Zertifikat authentisch ist.

1. Klicken Sie unter "Vertrauenswürdige Stammzertifizierungsstellen" mit der rechten Maustaste auf **Zertifikate** wählen Sie **Alle Aufgaben | Importieren** und befolgen Sie die Anweisungen des Zertifikatimportassistenten.



Nähere Informationen dazu finden Sie unter [https://technet.microsoft.com/en-us/library/cc754489\(v=ws.11\).aspx](https://technet.microsoft.com/en-us/library/cc754489(v=ws.11).aspx).

6.4 Aufrufen der Zertifikatspeicher unter Windows

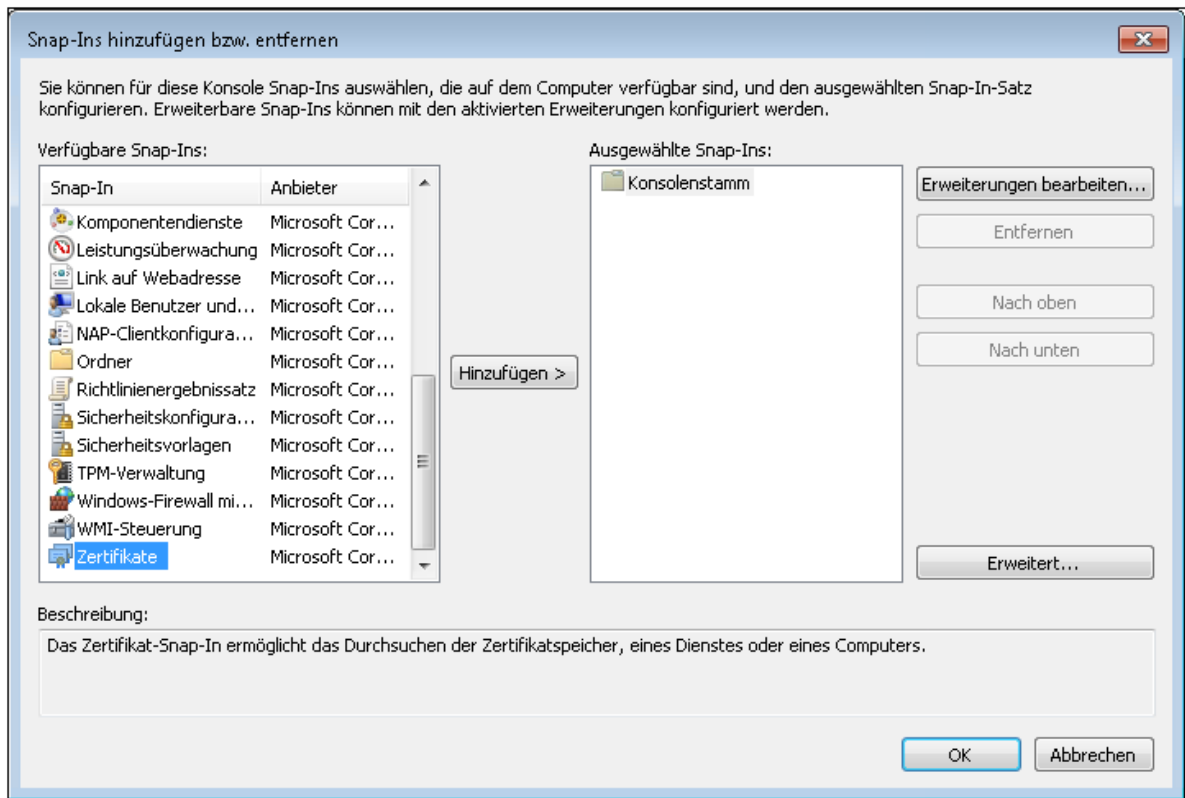
Auf Windows-Systemen können Sie Zertifikate entweder über das Microsoft Management Console (MMC) Snap-in entweder für Ihr Benutzerkonto oder für das Computerkonto verwalten.

So öffnen Sie das Snap-in "Zertifikate" (für den aktuellen Windows-Benutzer):

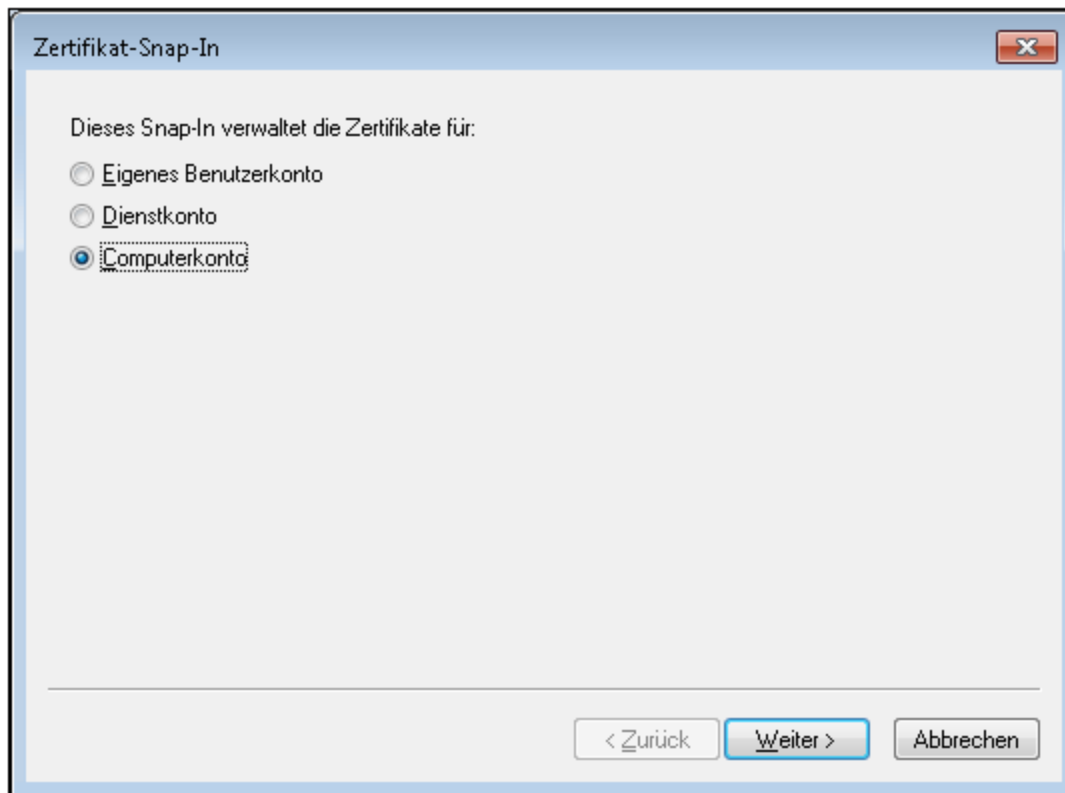
- Führen Sie `certmgr.msc` in der Befehlszeile aus.

So öffnen Sie das Snap-in "Zertifikate" (für das Computerkonto):

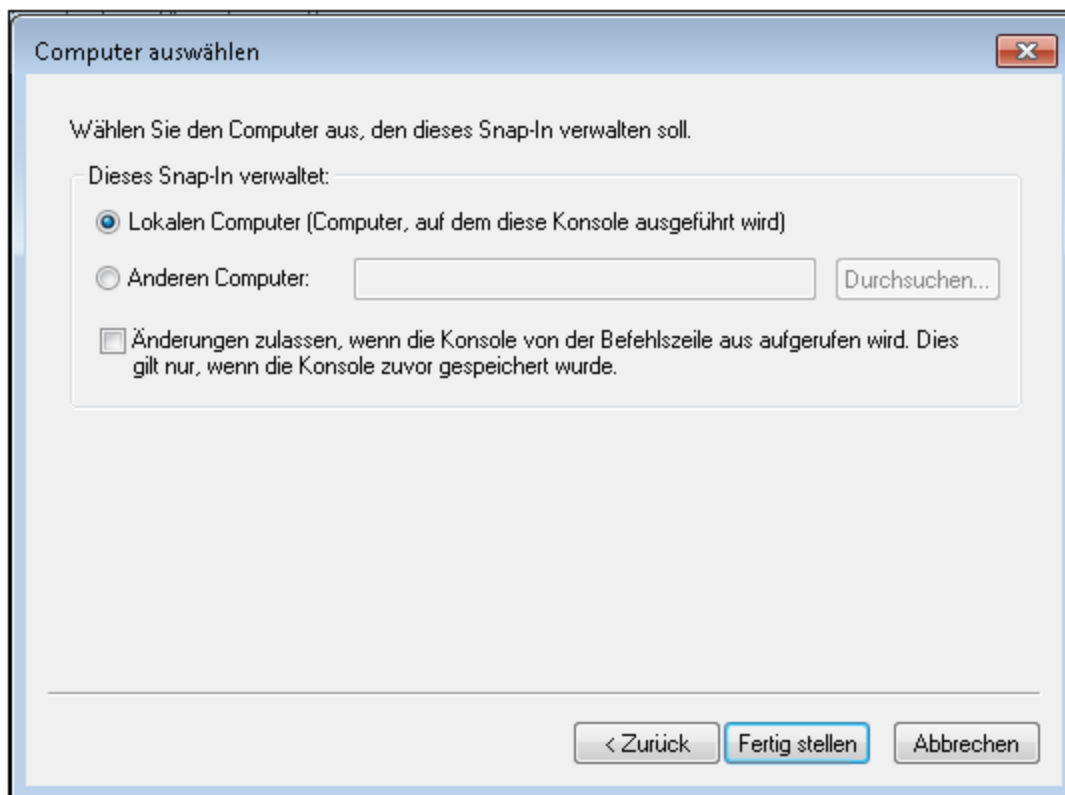
1. Führen Sie in der Befehlszeile den Befehl `mmc` aus.
2. Klicken Sie im Menü **Datei** auf **Snap-in hinzufügen/entfernen**.
3. Klicken Sie auf **Zertifikate** und anschließend auf **Hinzufügen**.



4. Klicken Sie auf **Computerkonto** und anschließend auf **Weiter**.



5. Klicken Sie auf **Lokalen Computer** und anschließend auf **Fertig stellen**.

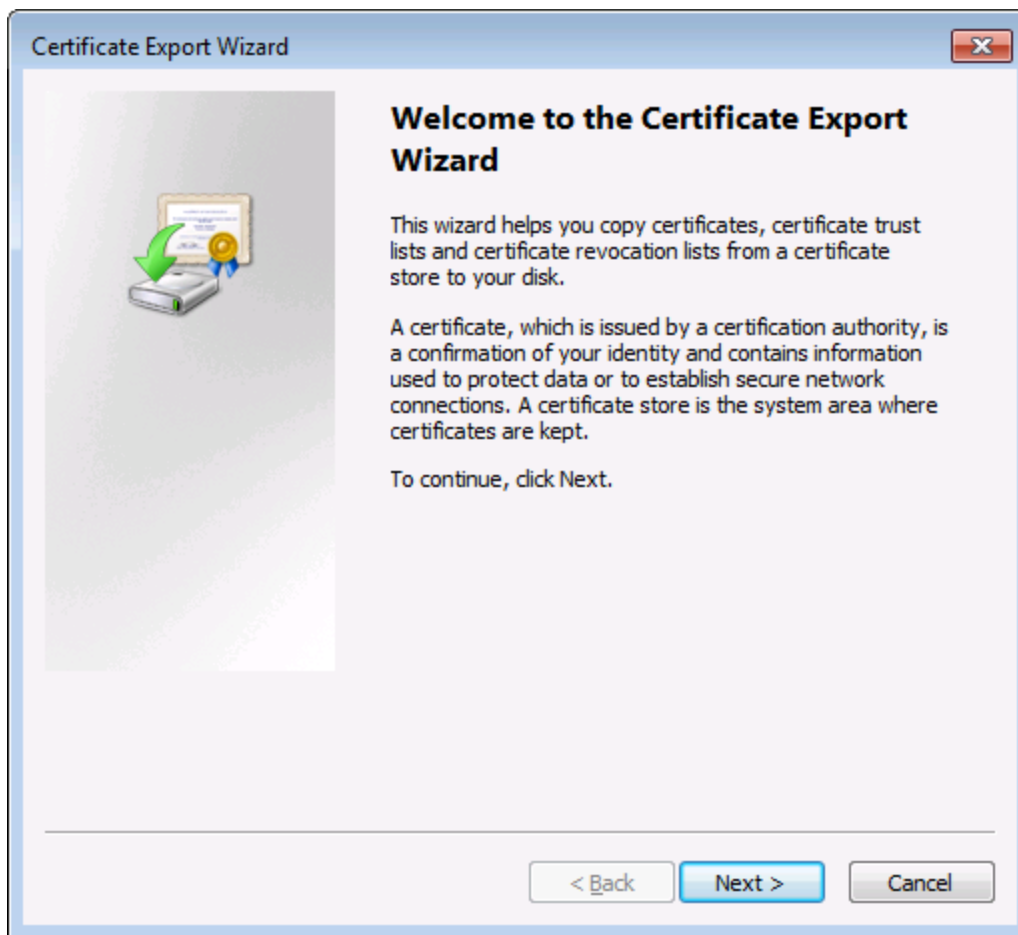


6.5 Exportieren von Zertifikaten aus Windows

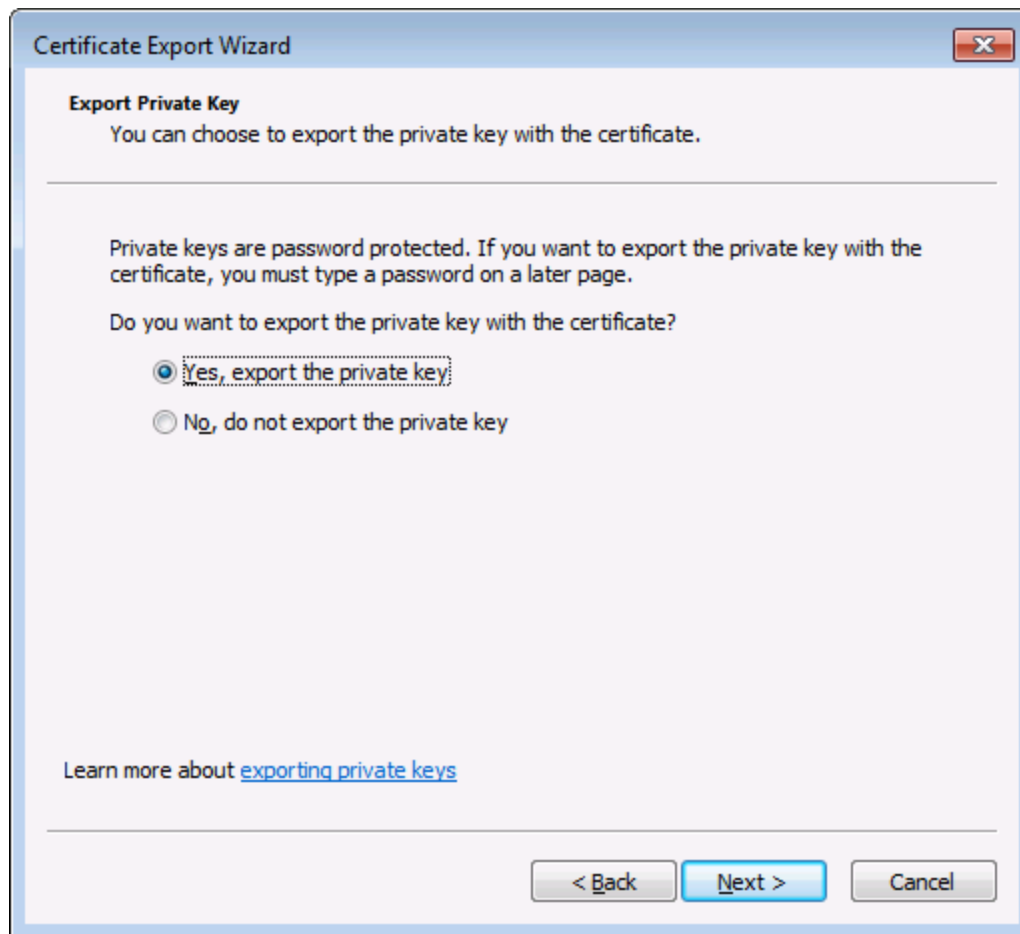
Bei Mappings, in denen Webservices über HTTPS aufgerufen werden und die auf einem macOS- oder Linux-Server, auf dem MapForce Server oder FlowForce Server ausgeführt wird, bereitgestellt werden, muss auf dem Nicht-Windows-Betriebssystem dasselbe Client-Zertifikat zur Verfügung stehen, das auch beim Design 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:

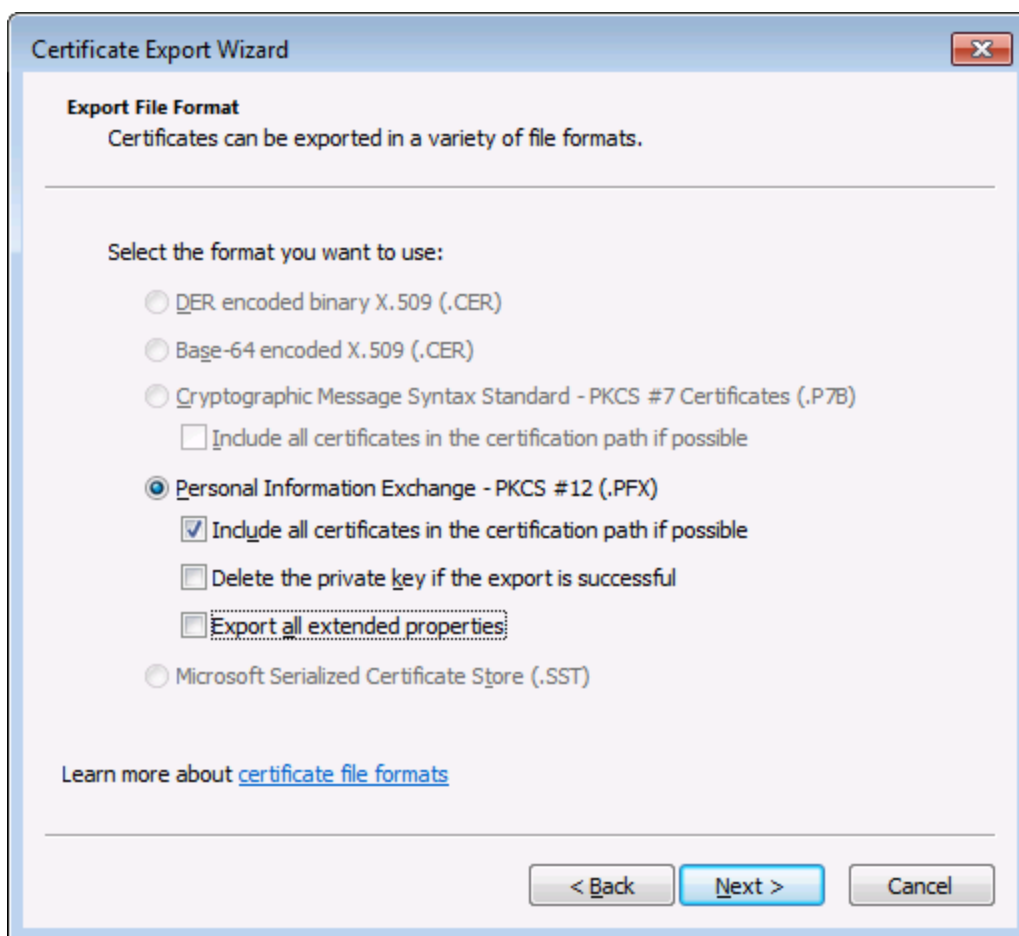
1. Öffnen Sie in Windows das Zertifikate-Snap-in (siehe [Aufrufen der Zertifikatspeicher unter Windows](#)¹¹²).
2. Klicken Sie mit der rechten Maustaste auf das gewünschte Zertifikat, platzieren Sie den Mauszeiger über **Alle Aufgaben** und klicken Sie auf **Exportieren**.
3. Klicken Sie auf **Weiter**.



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

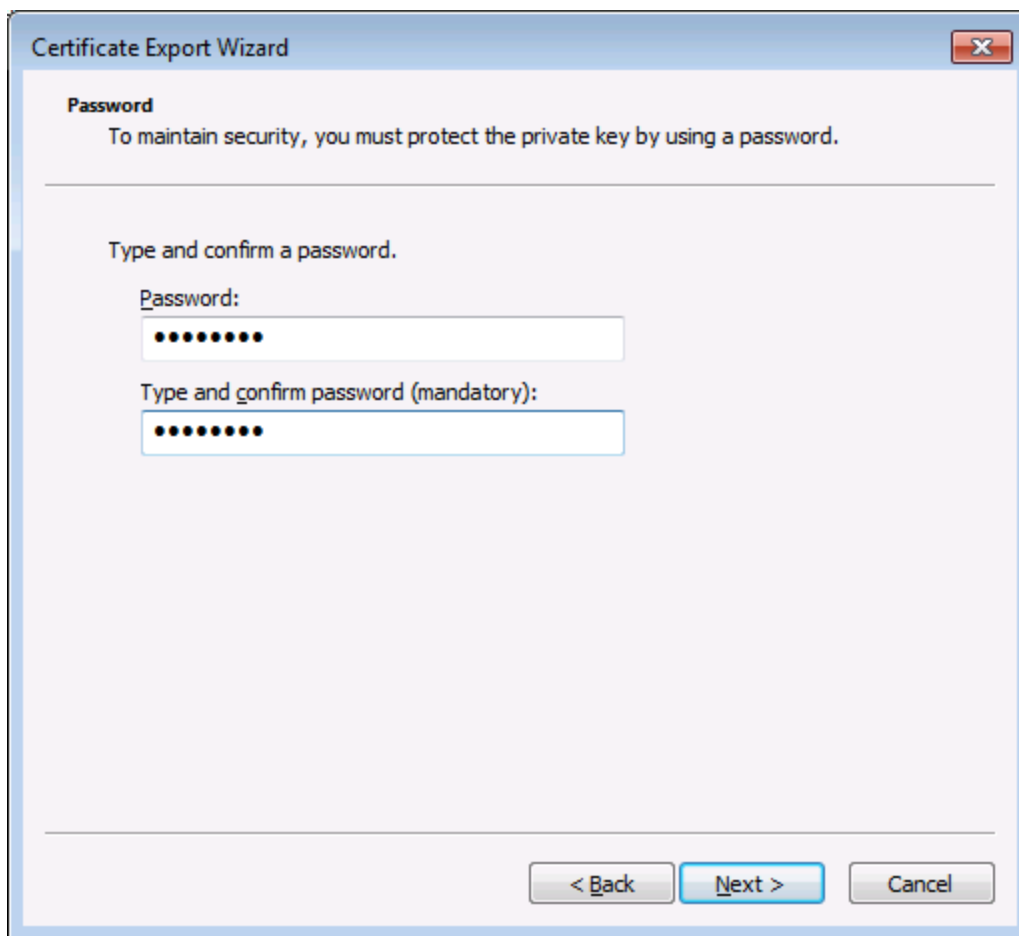


5. Wählen Sie das Dateiformat *Privater Informationsaustausch - PKCS #12 (.pfx)* aus und klicken Sie auf **Weiter**.

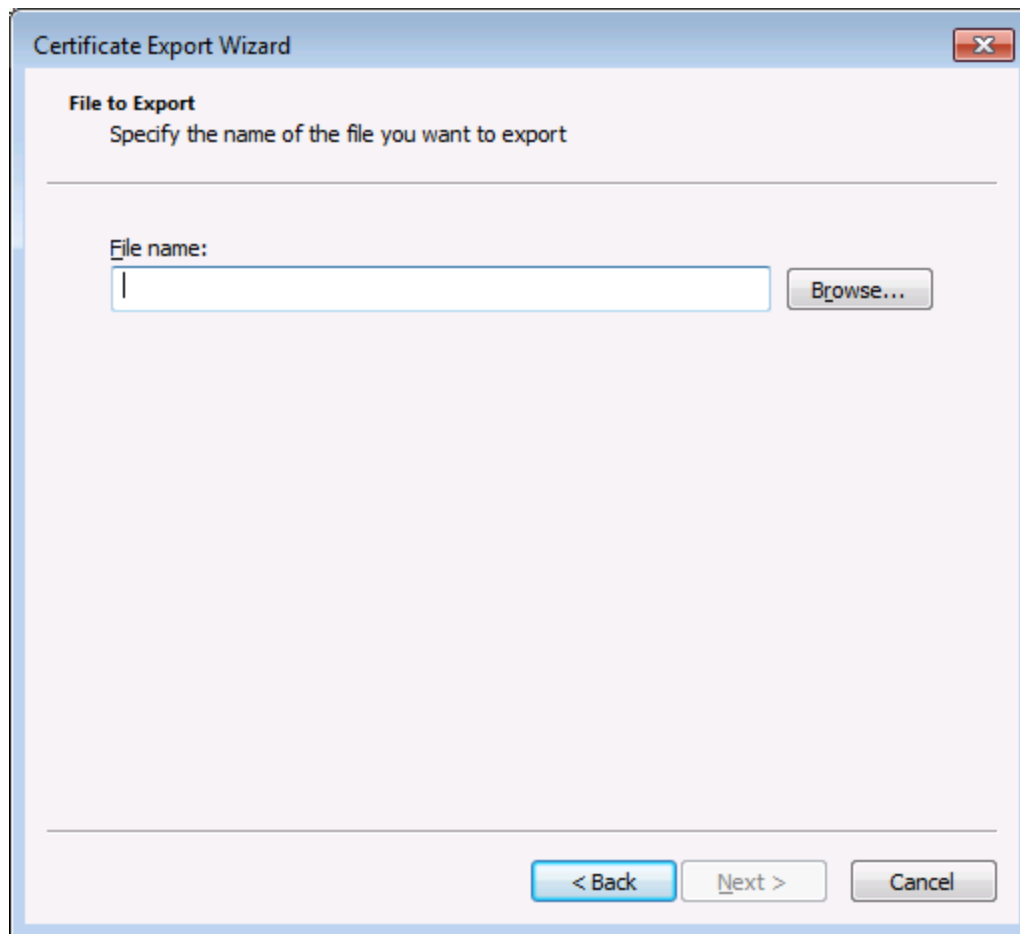


Anmerkung: Stellen Sie sicher, dass Sie nicht die Option **Privaten Schlüssel nach erfolgreichem Export löschen** aktiviert haben, da Sie das Zertifikat nach dem Export sonst nicht mehr verwenden können.

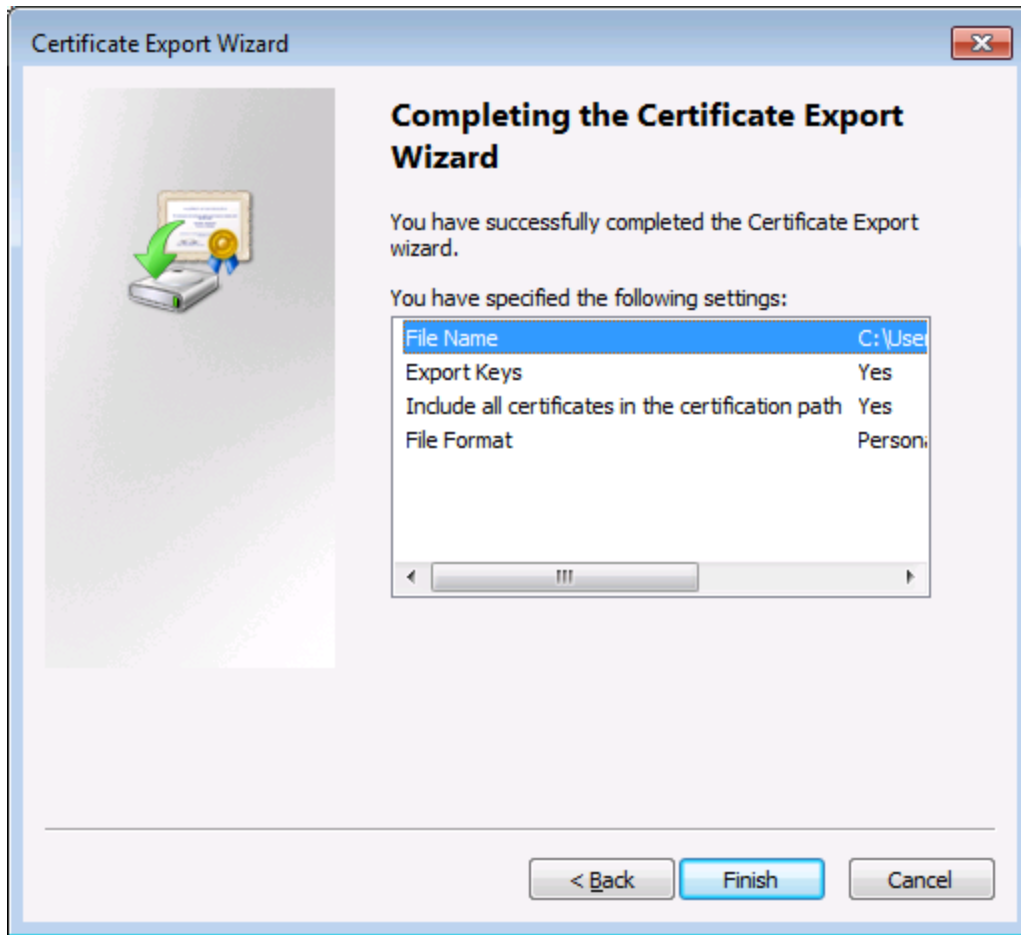
6. Geben Sie ein Passwort ein und klicken Sie auf **Weiter**. Sie benötigen dieses Passwort nachdem Sie das Zertifikat in das Zielbetriebssystem kopiert haben.



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](#)¹¹⁵).
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.

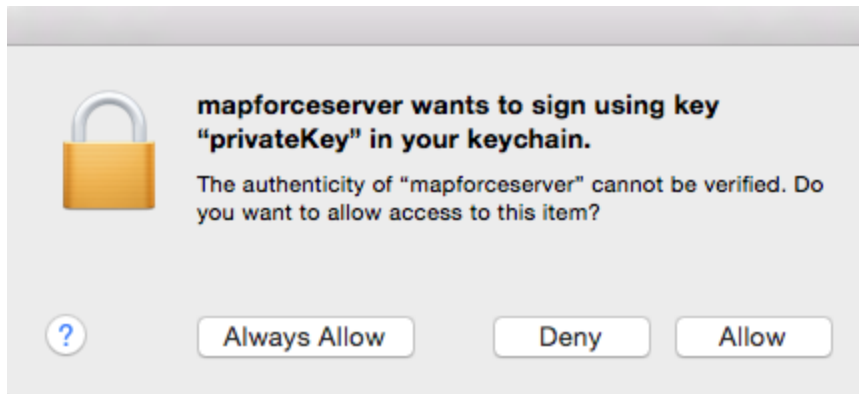
So übertragen Sie das Client-Zertifikat auf das macOS-System:

1. Exportieren Sie das Client-Zertifikat mit dem privaten Schlüssel im Dateiformat *Privater Informationsaustausch - PKCS #12 (.pfx)* (siehe [Exportieren von Zertifikaten aus Windows](#)¹¹⁵) 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](#)¹¹⁰).
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:

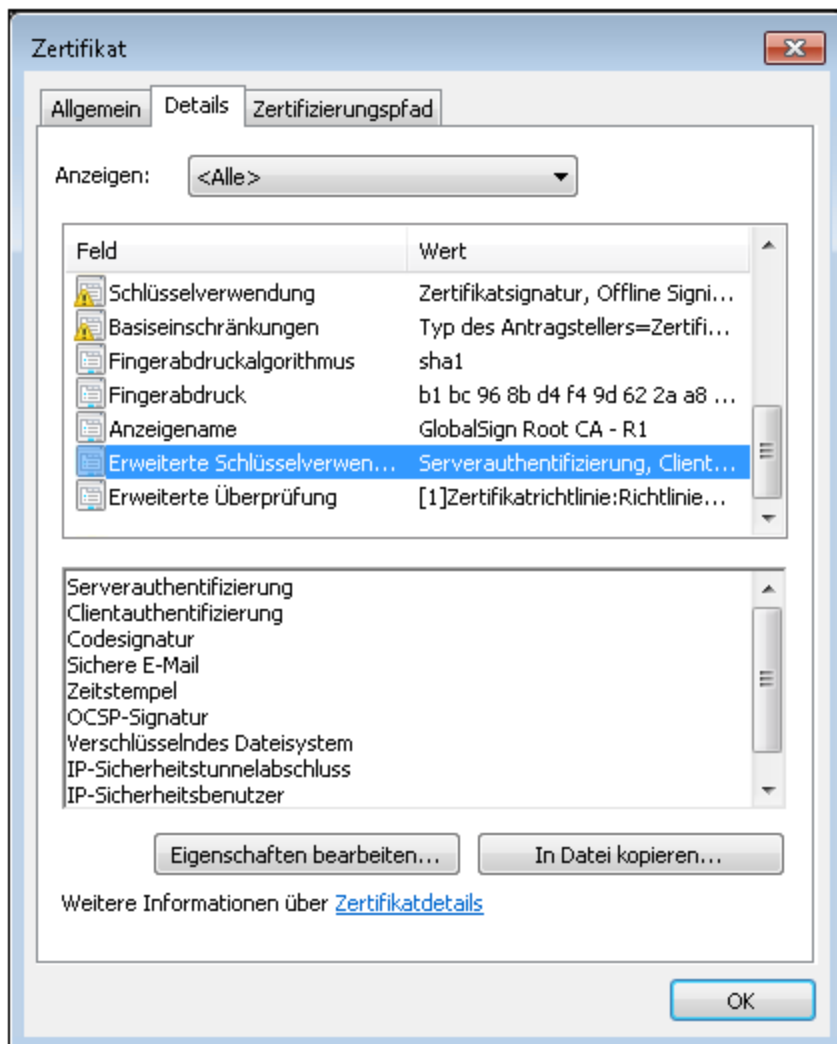
- Wenn Sie das Mapping über einen entfernten Rechner mittels SSH ausführen, entsperren Sie zuerst die Keychain mit dem Befehl `security unlock-keychain`.
- Wenn Sie das Mapping über die grafische Benutzeroberfläche von macOS ausführen, klicken Sie auf **Allow**, wenn Sie gefragt werden, ob MapForce Server 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 Katalogdateien

MapForce Server unterstützt einen Teil der OASIS XML-Kataloge (<https://www.oasis-open.org/committees/entity/spec-2001-08-06.html>). Mit Hilfe von Katalogen kann MapForce Server gebräuchliche DTDs und XML-Schemas (sowie Stylesheets und andere Dateien) aus lokalen Benutzerordnern abrufen, anstatt sie über eine öffentliche URI aufzulösen, wodurch die Verarbeitungsgeschwindigkeit erhöht wird, Sie auch offline arbeiten können (d.h. ohne mit einem Netzwerk verbunden zu sein) und Dokumente leichter übertragbar werden.

Funktionsweise von Katalogen

Kataloge dienen normalerweise dazu, eine öffentliche DTD- oder Schema-Referenz auf eine lokale URI (normalerweise einen lokalen Dateipfad) umzuleiten. Dazu wird in einer Katalogdatei im XML-Format eine Zuordnung zwischen der öffentlichen Schema-URI und einer lokalen URI definiert. Wenn also MapForce Server ein XML-Dokument parst, sucht es zuerst in der Katalogdatei nach der Schema-URI (oder gegebenenfalls dem Public oder System Identifier einer DTD). Wenn in der Katalogdatei eine Zuordnung gefunden wird, so wird das Schema anhand dieser Referenz aus einer lokalen Datei gelesen. Wenn in der Katalogdatei keine Zuordnung gefunden wird, wird die URI des XML-Dokuments in ihrem Ist-Zustand aufgelöst.

Angenommen, die folgende XML-Datei muss von MapForce Server verarbeitet werden:

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

Nehmen wir außerdem an, dass es irgendwo in einem lokalen Verzeichnis (das MapForce Server, wie weiter unten erläutert, kennt) eine **catalog.xml**-Datei gibt und diese die folgende Zeile enthält:

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

Beim Parsen der XML-Datei findet MapForce Server in der Katalogdatei eine Entsprechung für die Schema-Referenz `http://www.w3.org/2001/XMLSchema-instance.xsd`. Dadurch wird das Schema aus `files/XMLSchema-instance.xsd` (einem lokalen Pfad relativ zur Katalogdatei) geladen. Wenn in der Katalogdatei keine Zuordnung gefunden würde, würde das Schema aus `http://www.w3.org/2001/XMLSchema-instance` geladen werden.

Root-Katalog

Beim Start von MapForce Server wird eine Datei namens **RootCatalog.xml** aus dem Verzeichnis %

ProgramFilesFolder%\Altova\MapForceServer2021\etc geladen. **RootCatalog.xml** enthält eine Liste von Katalogdateien, wobei sich jede davon in einem `nextCatalog` Element befindet. Diese Katalogdateien werden durchsucht und die URIs werden von MapForce Server entsprechend den Zuordnungen in den Katalogdateien aufgelöst.

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

RootCatalog.xml

Beachten Sie, dass im Codefragment oben die folgenden Kataloge für den Lookup aufgelistet sind:

- **CustomCatalog.xml** ist die Datei, in der Sie Ihre eigenen Zuordnungen erstellen können. Diese Datei befindet sich im Verzeichnis **%ProgramFilesFolder%\Altova\MapForceServer2021\etc**. Sie können für jedes benutzerdefinierte Schema Zuordnungen zu **CustomCatalog.xml** hinzufügen, wenn dieses nicht schon von den von Altova konfigurierten Katalogdateien adressiert wird (siehe folgende Punkte).
- **CoreCatalog.xml** enthält bestimmte Altova-spezifische Mappings, zum Auffinden von Schemas. Diese Datei befindet sich ebenfalls im Verzeichnis **%ProgramFilesFolder%\Altova\MapForceServer2021\etc**.
- Mehrere **catalog.xml**-Dateien aus dem Verzeichnis **Schemas** relativ zur **RootCatalog.xml**-Datei. Jede **catalog.xml**-Datei befindet sich in einem Verzeichnis eines bestimmten Schemas (wie z.B. SVG, DITA, DocBook, WSDL, usw.) und jede weist URIs Public und/oder System Identifier zu, die auf lokal gespeicherte Kopien der entsprechenden Schemas verweisen.

Beachten Sie dazu Folgendes:

- Wenn Sie beabsichtigen, die Datei **CustomCatalog.xml** zu ändern, verwenden Sie nur die [unterstützten Elemente](#)¹²⁸. Außerdem dürfen vorhandene Zuordnungen nicht dupliziert werden, da dies Fehler verursachen könnte.

Unterstützte Elemente

Verwenden Sie beim Erstellen von Einträgen für **CustomCatalog.xml** nur die unten aufgelisteten Elemente. Andere Elemente der OASIS XML-Katalogspezifikation werden nicht unterstützt.

Element	Attribute	Beispiel
public	<ul style="list-style-type: none"> • <code>publicId</code> definiert den Public Identifier einer Ressource • <code>uri</code> definiert eine URI-Referenz (z.B. einen relativen Pfad zu einer lokalen Datei) 	<pre><public publicId="-//W3C//DTD XMLSCHEMA 200102//EN" uri="files/XMLSchema.dtd" /></pre>

Element	Attribute	Beispiel
system	<ul style="list-style-type: none"> • <code>systemId</code> definiert den System Identifier einer Ressource • <code>uri</code> definiert eine URI-Referenz (z.B. einen relativen Pfad zu einer lokalen Datei) 	<pre><system systemId="http://www.w3.org/2009/ XMLSchema/datatypes.dtd" uri="files/datatypes.dtd" /></pre>
uri	<ul style="list-style-type: none"> • <code>name</code> definiert eine URI-Referenz • <code>uri</code> definiert eine alternative URI-Referenz (z.B. einen relativen Pfad zu einer lokalen Datei) 	<pre><uri name="http://www.w3.org/2009/XMLS chema/XMLSchema.xsd" uri="files/XMLSchema.xsd" /></pre>
rewriteURI	<ul style="list-style-type: none"> • <code>uriStartString</code> definiert den Anfangsteil einer neu zu schreibenden URI • <code>uri</code> definiert den Ersetzungsstring (z.B. einen relativen Pfad zu einem lokalen Verzeichnis) 	<pre><rewriteURI uriStartString='http://www.altova .com/schemas/svg/' rewritePrefix='files/' /></pre>
rewriteSystem	<ul style="list-style-type: none"> • <code>systemIdStartString</code> definiert den Anfangsteil eines neu zu schreibenden System Identifiers • <code>rewritePrefix</code> definiert den Ersetzungsstring (z.B. einen relativen Pfad zu einem lokalen Verzeichnis) 	<pre><rewriteSystem systemIdStartString='http://www.a ltova.com/schemas/svg/' rewritePrefix='files/' /></pre>

Die Elemente **public**, **system** und **uri** können auch das Attribut `xml:base` erhalten, mit dem eine Basis-URI definiert wird, anhand welcher eine relative URI aufgelöst würde. Nähere Informationen finden Sie in der XML-Katalogspezifikation (<http://www.oasis-open.org/committees/entity/spec-2001-08-06.html>).

Index

A

ADO, 6, 19
ADO.NET, 6, 19
assignlicense,
als CLI-Befehl, 33

B

Befehlszeilenschnittstelle,
Verwendung, 30

C

C#,
Beispiel, 48
Mappings ausführen mit, 48
C++,
Beispiel, 54
Mappings ausführen mit, 54
CLI-Befehle,
assignlicense, 33
exportresourcestrings, 34
help, 36
licenseserver, 37
run, 39
setdeflang, 43
verifylicense, 44, 45

D

Digitale Zertifikate,
auf Linux übertragen, 121
auf Mac übertragen, 123
aus Windows exportieren, 115
in MapForce Mappings, 105
unter Windows verwalten, 112

vertrauenswürdige auf Mac-Systemen einrichten, 110
vertrauenswürdige unter Linux einrichten, 108
vertrauenswürdige unter Windows einrichten, 111

E

exportresourcestrings,
als CLI-Befehl, 34

F

FlowForce Server, 6, 16, 19

G

Globale Ressourcen,
Definition, 25
in Mappings verwenden, 25

H

help,
als CLI-Befehl, 36
HTTPS,
Webservices aufrufen über, 105

J

Java, 19
Beispiel, 65
MapForce Server aufrufen über, 65
MapForce Server zum CLASSPATH hinzufügen, 63
JDBC, 6

L

licenseserver,
als CLI-Befehl, 37

Linux,

- Anzeige der installierten Altova-Produkte, 11
- Client-Zertifikate übertragen auf, 121
- Deinstallation von MapForce Server von, 11
- Info zur Unterstützung, 7
- Installation von MapForce Server unter, 10
- Mappings mit Webservice-Aufrufen über HTTPS ausführen, 121
- vertrauenswürdige Server-Zertifikate einrichten, 108

M**Mac,**

- Client-Zertifikate übertragen auf, 123
- Mappings mit Webservice-Aufrufen über HTTPS ausführen, 123
- vertrauenswürdige Server-Zertifikate einrichten, 110

macOS,

- Deinstallation von MapForce Server von, 13
- Info zur Unterstützung, 7
- Installation von MapForce Server unter, 12

MapForce Server API,

- C#-Beispiel, 48
- C++-Beispiel, 54
- Einführung, 46
- für .NET, 47
- für COM, 54
- für Java, 63
- Java-Beispiel, 65
- VBA-Beispiel, 60

Mappings,

- auf FlowForce Server bereitstellen, 16
- mit C# ausführen, 48
- mit C++ ausführen, 54
- mit Java ausführen, 65
- mit VBA ausführen, 60
- über die Befehlszeile ausführen, 39
- vorbereiten für die Ausführung, 19
- zu Ausführungsdateien kompilieren, 16

O**ODBC, 6, 19****R****run,**

- als CLI-Befehl, 39

S**setdeflang,**

- als CLI-Befehl, 43

V**VBA,**

- Beispiel, 60
- Mappings ausführen mit, 60

verifylicense,

- als CLI-Befehl, 44

version,

- als CLI-Befehl, 45

W**Webservices,**

- über HTTPS aufrufen, 105

Windows,

- Deinstallation von MapForce Server von, 15
- Info zur Unterstützung, 7
- Installation von MapForce Server unter, 14
- Mappings mit Webservice-Aufrufen über HTTPS ausführen, 125
- vertrauenswürdige Server-Zertifikate einrichten, 111

X**XML Catalogs,**

- configuring, 127
- how it works, 127