Altova MapForce Server 2017

User and Reference Manual

Altova MapForce Server 2017 User & Reference Manual

Published: 2017

Table of Contents

Chapter 1

Altova MapForce Server 2017

1 Altova MapForce Server 2017

MapForce Server is an enterprise software solution that runs data mapping transformations on Windows, Linux, and OS X operating systems. The data mappings themselves (or Mapping Design Files, *.mfd) are visually designed with Altova MapForce (http://www.altova.com/mapforce.html), where you define the inputs, outputs, and any intermediary processing steps that must be applied to your data. The role of MapForce Server is to run MapForce Server Execution (.mfx) files compiled with MapForce, and to produce the output files or data, or even update databases or call Web services, according to the design of the underlying mapping.

MapForce Server can run standalone as well as under the management of Altova FlowForce Server (http://www.altova.com/flowforce.html). When installed on the same machine as MapForce Server, FlowForce Server automates execution of mappings through scheduled or trigger-based jobs, which can also be exposed as Web services. In addition to this, FlowForce Server includes a built-in library of functions that enable you to take additional automated actions before or after mapping execution, such as sending email, copying files and directories, uploading files to FTP, running shell commands, and others.

MapForce Server Features

- Server-level performance when executing data mappings
- Cross-platform: MapForce Server runs on Windows, Linux, or OS X operating systems
- Command line interface
- An API that you can call from C++, C#, Java, VB.NET, VBScript, or VBA code
- Native integration with FlowForce Server

Limitations

- XML digital signatures are not supported
- Altova Global Resources (http://www.altova.com/global-resources.html) are not supported by the API; this limitation does not apply to the command line interface.
- ODBC and ADO database connections are supported only on Windows systems. On other operating systems, JDBC should be used.

System Requirements

▾ *Windows*

Windows Vista, Windows 7/8/10

▾ *Windows Server*

Windows Server 2008 R2 or newer

On Windows, MapForce Server is available both as a 32-bit and 64-bit package.

▾ *Linux*

- CentOS 6 or newer
- RedHat 6 or newer

- Debian 7 or newer
- Ubuntu 12.04 or newer

Note that the Libidn library, available under GNU LGPL, must be installed.

▼ *Mac OS X*

Mac OS X 10.10, 10.11, 10.12 or newer

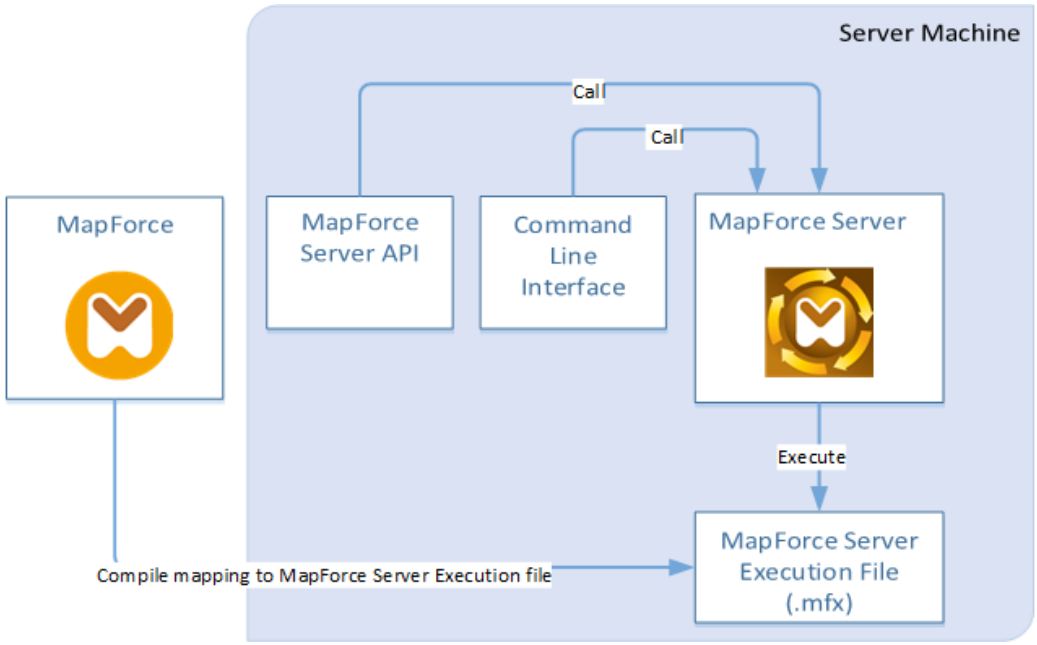
Last updated: 31 March 2017

1.1 How It Works

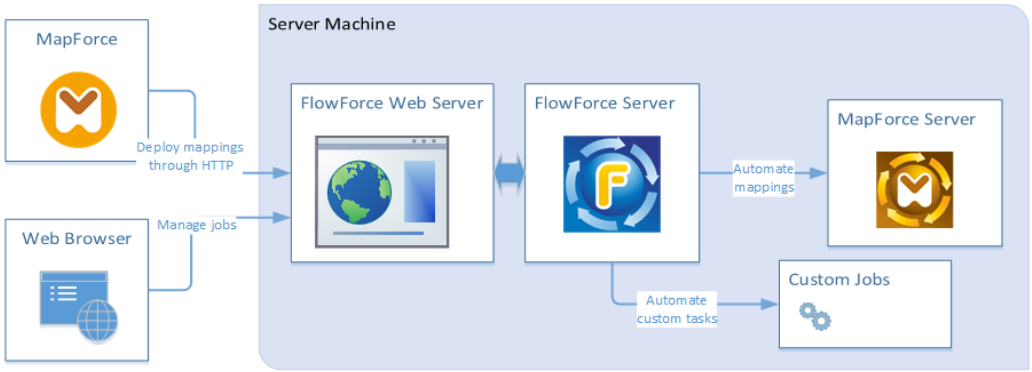
The role of MapForce Server is to execute data mappings created with Altova MapForce (http://www.altova.com/mapforce.html).

First, you design the data mappings (or Mapping Design Files, *.mfd) visually in MapForce, where you define the inputs, outputs, and any intermediary processing steps that must be applied to your data (including sorting, filtering, custom functions, and others). Once your mapping is ready, you can execute it with MapForce Server in one of the following ways:

- On the Windows machine where MapForce runs, compile the mapping to a MapForce Server Execution File (.mfx). The .mfx files can roughly be regarded as data mappings packaged for execution in a server environment. You can copy such files to any of the supported operating systems where MapForce Server runs (including across different platforms, see System Requirements). On the server machine, you can execute the .mfx file using the command line interface of MapForce Server, or using the MapForce Server API.



- On the Windows machine where MapForce runs, deploy the mapping to a server machine where both MapForce Server and FlowForce Server are installed. The server machine can be a different operating system (see System Requirements). Mappings deployed in this way become FlowForce Server functions and you can create scheduled or trigger-based jobs from them. When mappings run as FlowForce Server jobs, they can also be exposed as Web services, chained as sub-steps of other jobs, or made part of workflows which include sending emails, verifying exit codes, running shell commands, and others.



For more information about this scenario, see the FlowForce Server documentation (http://manual.altova.com/FlowForceServer).

How to execute mappings compiled as MapForce Server Execution files

1. Open in MapForce Enterprise or Professional the mapping to be compiled.
2. On the **File** menu, click **Compile to MapForce Server Execution file**, and select a destination directory.
3. Copy the .mfx file to the destination directory or server.
4. Call the "run" command of the command line interface (see Command Line Usage), or the equivalent method of the MapForce Server API (see MapForce Server API).

How to execute mappings deployed to FlowForce Server

1. Open in MapForce Enterprise or Professional the mapping to be deployed.
2. Make sure that the transformation language (execution engine) of the mapping is set to Built-in. To change the execution engine to Built-in, select the menu command **Output | Built-In Execution Engine**, or click the **Select Built-In Execution Engine** () toolbar button .
3. On the **File** menu, click **Deploy to FlowForce Server**.
4. Enter the server connection details (host, port), the FlowForce credentials, and the destination FlowForce container. To proceed to creating the FlowForce job immediately in the browser, select the option **Open web browser to create new job**. You can also create the FlowForce job at a later time (see next step).
5. Open a browser, log on to the FlowForce Server Web administration interface, and navigate to the container where you deployed the mapping (see previous step). This step is not required if you selected the option **Open web browser to create new job** in the previous step.
6. Define the FlowForce Server job, including its triggers, parameters, or additional execution steps (for an example, see http://manual.altova.com/FlowForceServer/fs_example_mapforce_mapping.htm). Whenever the job is configured to run, the underlying mapping transformation will be executed, and the mapping output will be produced.

Note: If MapForce Server runs on a machine other than the one where the mapping was designed, make sure to adjust paths to input files or database connection details in such

a way that they are meaningful in the new target execution environment. For example, if a mapping calls a database and requires a database driver, the driver must also be installed in the target environment in order for the mapping to be executed successfully. To view or adjust the database connection details, right-click the database component in MapForce and select **Properties**. After making any changes to the mapping design in MapForce, remember to recompile it to a MapForce Server execution file (.mfx) or, depending on the case, redeploy it to FlowForce Server. For more information about how database mappings and file paths are handled in various execution environments, refer to the MapForce documentation (https://www.altova.com/documentation.html).

Chapter 2

Setup on Windows

2 Setup on Windows

This section describes the installation and licensing of MapForce Server on Windows systems.

Installation on Windows

- System requirements
- Installing MapForce Server
- Altova LicenseServer
- LicenseServer versions
- Trial license
- Application folder location

Licensing on Windows

- Start ServiceController
- Start LicenseServer
- Register MapForce Server
- Assign a license

2.1 Installation on Windows

MapForce Server is available for installation on Windows systems. Its installation and setup procedure is described below.

- System requirements
  - *Windows*

    Windows Vista, Windows 7/8/10

  - *Windows Server*

    Windows Server 2008 R2 or newer

- Installing MapForce Server

  To install MapForce Server, download the installation package from the Altova Download Center (http://www.altova.com/download.html), run it and follow the on-screen instructions.

  After installation, the MapForce Server executable will be located by default at:

  ```
  <ProgramFilesFolder>\Altova\MapForceServer2017\bin\MapForceServer.exe
  ```

  All the necessary registrations to use MapForce Server via a COM interface, as a Java interface, and in the .NET environment will be done by the installer.

- Altova LicenseServer
  - In order for MapForce Server to work, it must be licensed via an Altova LicenseServer on your network.
  - When you install MapForce Server on Windows systems, an option is available that allows you to download and install Altova LicenseServer together with MapForce Server.
  - If an Altova LicenseServer is already installed on your network, you do not need to install another one—unless a newer version of Altova LicenseServer is required. (*See next point, LicenseServer versions*.)
  - During the installation process of MapForce Server, check or uncheck the option for installing Altova LicenseServer as appropriate.

  See the section, Licensing on Windows, for more information about how to register and license MapForce Server with Altova LicenseServer.

- LicenseServer versions
  - Altova server products must be licensed either with the version of LicenseServer that is appropriate to the installed MapForce Server version, or with a later version of LicenseServer.
  - The LicenseServer version that is appropriate for a particular version of MapForce Server is displayed during the installation of MapForce Server. You can install this version of LicenseServer along with MapForce Server, or you can install

LicenseServer separately.

- Before installing a newer version of LicenseServer, any older one must be de-installed. The LicenseServer installer will do this automatically if it detects an older version.
- LicenseServer versions are backwards compatible. They will work with older versions of MapForce Server.
- If you install a new version of MapForce Server and if your installed LicenseServer version is older than the appropriate LicenseServer, install the latest version available on the Altova website.
- At the time of LicenseServer de-installation, all registration and licensing information held in the older version of LicenseServer will be saved to a database on your server machine. This data will be imported automatically into the newer version when the newer version is installed.
- The version number of the currently installed LicenseServer is given at the bottom of the LicenseServer configuration page (all tabs).

Current version: 2.3

▼ Trial license

During the installation process, you will be given the option of requesting a 30-day trial license for MapForce Server. After submitting the request, a trial license will be sent to the email address you registered.

▼ Application folder location

The application will be installed in the following folder:

Windows Vista, Windows 7, 8, 10	`C:\Program Files\Altova\`
32 bit Version on 64-bit OS	`C:\Program Files (x86)\Altova\`

2.2 Licensing on Windows

MapForce Server must be licensed with an Altova LicenseServer in order to run it. Licensing is a two-step process:

1. **Register MapForce Server** with LicenseServer. Registration is done from MapForce Server.
2. **Assign a license** to MapForce Server. License-assigning is done from LicenseServer.

The steps you need to carry out are given below.

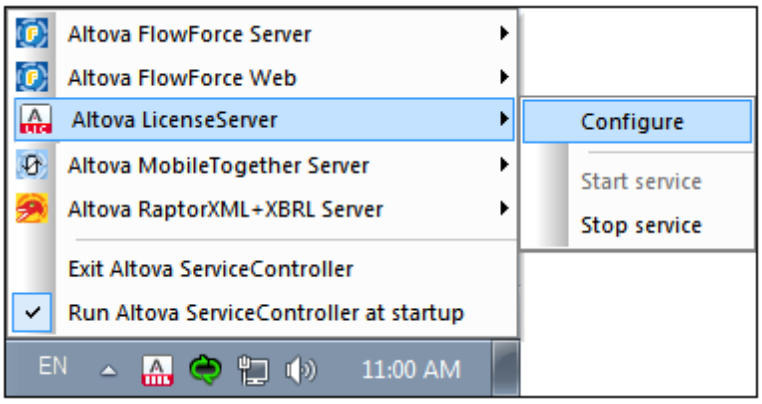
▼ Start ServiceController

Altova ServiceController is started in order to start Altova LicenseServer.

Altova ServiceController (ServiceController for short) is an application for conveniently starting, stopping and configuring Altova services **on Windows systems**.

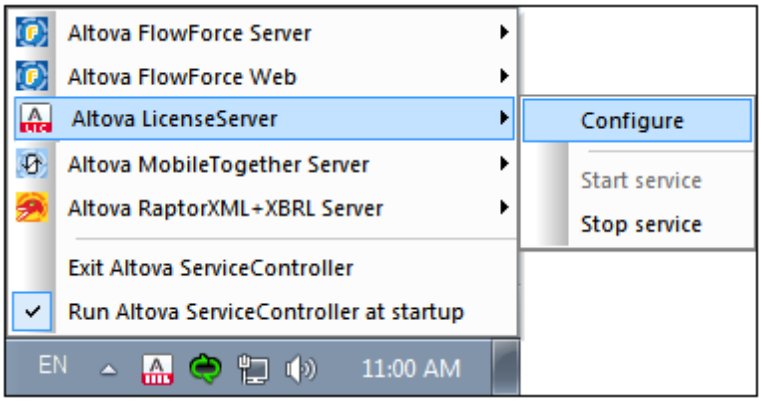
ServiceController is installed with Altova LicenseServer and with *Altova server products that are installed as services* (FlowForce Server, RaptorXML(+XBRL) Server, and Mobile Together Server). It can be started by clicking **Start | Altova LicenseServer | Altova ServiceController**. (This command is also available in the **Start** menu folders of *Altova server products that are installed as services* (FlowForce Server, RaptorXML(+XBRL) Server, and Mobile Together Server).) After ServiceController has been started, it can be accessed via the system tray (*screenshot below*).

To specify that ServiceController starts automatically on logging in to the system, click the **ServiceController** icon in the system tray to display the **ServiceController** menu (*screenshot below*), and then toggle on the command **Run Altova ServiceController at Startup**. (This command is toggled on by default.) To exit ServiceController, click the **ServiceController** icon in the system tray and, in the menu that appears (*see screenshot below*), click **Exit Altova ServiceController**.



- Start LicenseServer

  To start LicenseServer, click the **ServiceController** icon in the system tray, hover over **Altova LicenseServer** in the menu that pops up (*see screenshot below*), and then select **Start Service** from the LicenseServer submenu. If LicenseServer is already running, the *Start Service* option will be disabled.



- Register MapForce Server

  To register MapForce Server from the command line interface, use the `licenseserver` command:

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

  For example, if `localhost` is the name of the server on which LicenseServer is installed:

  ```
  MapForceServer licenseserver localhost
  ```

  After successful registration, go to the Server Management tab of LicenseServer's configuration page to assign a license to MapForce Server.

- Assign a license

  After successfully registering MapForce Server, it will be listed in the Server Management tab of the configuration page of LicenseServer. Go there and assign a license to MapForce Server.

  ### Note on cores and licenses

  The licensing of Altova server products is based on the number of processor cores available on the product machine. For example, a dual-core processor has two cores, a quad-core processor four cores, a hexa-core processor six cores, and so on. The number of cores licensed for a product must be greater than or equal to the number of cores available on that server machine, whether the server is a physical or virtual machine. For example, if a server has eight cores (an octa-core processor), you must purchase at least one 8-core license. You can also combine licenses to achieve the core count. So, two 4-core licenses can also be used for an octa-core server instead of one 8-core license.

  If you are using a computer server with a large number of CPU cores but only have a low volume to process, you may also create a virtual machine that is allocated a smaller number

of cores, and purchase a license for that number. Such a deployment, of course, will have less processing speed than if all available cores on the server were utilized.

Note: Each Altova server product license can be used for only one client machine—the machine on which the Altova server product is installed—at a time, even if the license has unused licensing capacity. For example, if a 10-core license is used for a client machine that has 6 CPU cores, then the remaining 4 cores of licensing capacity cannot be used simultaneously for another client machine.

MobileTogether Server licenses

MobileTogether Server licenses are based on the number of CPU cores on the MobileTogether Server machine. Core licenses allow an unlimited number of MobileTogether Client devices to connect to the server. However, if you check the *Limit to single thread execution* check box, then only one mobile device will be able to connect to the MobileTogether Server at any time. This is useful for evaluation and small-scale testing. Note that, if, in this case, a second device connects to MobileTogether Sever, then it will take over the license. The first device will not be able to connect any more and will receive an error message to this effect.

Chapter 3

Setup on Linux

3 Setup on Linux

This section describes the installation and licensing of MapForce Server on Linux systems (Debian, Ubuntu, CentOS, RedHat).

Installation on Linux

- System requirements
- Uninstall old versions of Altova server products
- Download the Linux package
- Install MapForce Server
- Altova LicenseServer
- LicenseServer versions
- Trial license

Licensing on Linux

- Start LicenseServer
- Register MapForce Server
- Assign a license

Notes about Environment

3.1 Installation on Linux

MapForce Server is available for installation on Linux systems. Its installation and setup procedure is described below.

System requirements

Linux

- CentOS 6 or newer
- RedHat 6 or newer
- Debian 7 or newer
- Ubuntu 12.04 or newer

Note that the Libidn library, available under GNU LGPL, must be installed.

FlowForce Server integration

If you are installing MapForce Server together with FlowForce Server, it is recommended that you install FlowForce Server first. Otherwise, after having installed both MapForce Server and FlowForce Server, run the following command:

```
cp /opt/Altova/MapForceServer2017/etc/*.tool /opt/Altova/
FlowForceServer2017/tools
```

This command copies the **.tool** file from **/etc** directory of MapForce Server to the FlowForce Server **/tools** directory. The **.tool** file is required by FlowForce Server; it contains the path to the MapForce Server executable. You do not need to run this command if you install FlowForce Server before installing MapForce Server.

Uninstall old versions of Altova server products

If you need to uninstall a previous version, do this as follows. On the Linux command line interface (CLI), you can check which Altova server products are installed with the following command:

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

If MapForce Server is not installed, go ahead with the installation as documented below in *Installing MapForce Server*.

If MapForce Server is installed and you wish to install a newer version of MapForce Server, uninstall the old version with the command:

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

If you need to uninstall an old version of Altova LicenseServer, do this with the following command:

```
[Debian, Ubuntu]:   sudo dpkg --remove licenseserver
```

```
[CentOS, RedHat]:   sudo rpm -e licenseserver
```

Download the Linux package

MapForce Server installation packages for the following Linux systems are available at the Altova website.

Distribution	Package extension
Debian 7 and higher	.deb
Ubuntu12.04 and higher	.deb
CentOS 6 and higher	.rpm
RedHat 6 and higher	.rpm

After downloading the Linux package, copy it to any directory on the Linux system. Since you will need an Altova LicenseServer in order to run MapForce Server, you may want to download LicenseServer from the Altova website at the same time as you download MapForce Server, rather than download it at a later time.

Install MapForce Server

In a terminal window, switch to the directory where you have copied the Linux package. For example, if you copied it to a user directory called MyAltova (that is located, say, in the /home/User directory), then switch to this directory as follows:

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

Install MapForce Server with the following command:

```
[Debian]:   sudo dpkg --install mapforceserver-2017-debian.deb
[Ubuntu]:   sudo dpkg --install mapforceserver-2017-ubuntu.deb
[CentOS]:   sudo rpm -ivh mapforceserver-2017-1.x86_64.rpm
[RedHat]:   sudo rpm -ivh mapforceserver-2017-1.x86_64.rpm
```

The MapForce Server package will be installed in the folder:

```
/opt/Altova/MapForceServer2017
```

Altova LicenseServer

In order for any Altova Server product—including MapForce Server—to run, that server product must be licensed via an Altova LicenseServer on your network.

On Linux systems, Altova LicenseServer will need to be installed separately. Download LicenseServer from the Altova website and copy the package to any directory on the Linux system. Install it just like you installed MapForce Server (*see previous step*).

```
[Debian]:   sudo dpkg --install licenseserver-2.3-debian.deb
[Ubuntu]:   sudo dpkg --install licenseserver-2.3-ubuntu.deb
[CentOS]:   sudo rpm -ivh licenseserver-2.3-1.x86_64.rpm
[RedHat]:   sudo rpm -ivh licenseserver-2.3-1.x86_64.rpm
```

The LicenseServer package will be installed in:
/opt/Altova/LicenseServer

For information about how to register MapForce Server with Altova LicenseServer and license it, see the section, Licensing on Linux.

- LicenseServer versions

  - Altova server products must be licensed either with the version of LicenseServer that is appropriate to the installed MapForce Server version, or with a later version of LicenseServer.
  - The LicenseServer version that is appropriate for a particular version of MapForce Server is displayed during the installation of MapForce Server. You can install this version of LicenseServer along with MapForce Server, or you can install LicenseServer separately.
  - Before installing a newer version of LicenseServer, any older one must be de-installed. The LicenseServer installer will do this automatically if it detects an older version.
  - LicenseServer versions are backwards compatible. They will work with older versions of MapForce Server.
  - If you install a new version of MapForce Server and if your installed LicenseServer version is older than the appropriate LicenseServer, install the latest version available on the Altova website.
  - At the time of LicenseServer de-installation, all registration and licensing information held in the older version of LicenseServer will be saved to a database on your server machine. This data will be imported automatically into the newer version when the newer version is installed.
  - The version number of the currently installed LicenseServer is given at the bottom of the LicenseServer configuration page (all tabs).

    Current version: 2.3

- Trial license

  During the installation process, you will be given the option of requesting a 30-day trial license for MapForce Server. After submitting the request, a trial license will be sent to the email address you registered.

3.2 Licensing on Linux

MapForce Server must be licensed with an Altova LicenseServer in order to run it. Licensing is a two-step process:

1. **Register MapForce Server** with LicenseServer. Registration is done from MapForce Server.
2. **Assign a license** to MapForce Server. License-assigning is done from LicenseServer.

The steps you need to carry out are given below.

▼ Start LicenseServer

To correctly register and license MapForce Server with LicenseServer, LicenseServer must be running as a daemon on the network. Start LicenseServer as a daemon with the following command:

[< Debian 8]	**sudo /etc/init.d/licenseserver start**
[≥ Debian 8]	**sudo systemctl start licenseserver**
[< CentOS 7]	**sudo initctl start licenseserver**
[≥ CentOS 7]	**sudo systemctl start licenseserver**
[< Ubuntu 15]	**sudo initctl start licenseserver**
[≥ Ubuntu 15]	**sudo systemctl start licenseserver**
[RedHat]	**sudo initctl start licenseserver**

If at any time you need to stop LicenseServer, replace **start** with **stop** in the above commands. For example:

```
sudo /etc/init.d/licenseserver stop
```

▼ Register MapForce Server

To register MapForce Server from the command line interface, use the `licenseserver` command:

```
sudo /opt/Altova/MapForceServer2017/bin/mapforceserver licenseserver
[options] ServerName-Or-IP-Address
```

For example, if `localhost` is the name of the server on which LicenseServer is installed:

```
sudo /opt/Altova/MapForceServer2017/bin/mapforceserver licenseserver
localhost
```

In the command above, `localhost` is the name of the server on which LicenseServer is installed. Notice also that the location of the MapForce Server executable is:

```
/opt/Altova/MapForceServer2017/bin/
```

After successful registration, go to the Server Management tab of LicenseServer's

configuration page to assign a license to MapForce Server.

▼ Assign a license

After successfully registering MapForce Server, it will be listed in the Server Management tab of the configuration page of LicenseServer. Go there and assign a license to MapForce Server.

Note on cores and licenses

The licensing of Altova server products is based on the number of processor cores available on the product machine. For example, a dual-core processor has two cores, a quad-core processor four cores, a hexa-core processor six cores, and so on. The number of cores licensed for a product must be greater than or equal to the number of cores available on that server machine, whether the server is a physical or virtual machine. For example, if a server has eight cores (an octa-core processor), you must purchase at least one 8-core license. You can also combine licenses to achieve the core count. So, two 4-core licenses can also be used for an octa-core server instead of one 8-core license.

If you are using a computer server with a large number of CPU cores but only have a low volume to process, you may also create a virtual machine that is allocated a smaller number of cores, and purchase a license for that number. Such a deployment, of course, will have less processing speed than if all available cores on the server were utilized.

Note: Each Altova server product license can be used for only one client machine—the machine on which the Altova server product is installed—at a time, even if the license has unused licensing capacity. For example, if a 10-core license is used for a client machine that has 6 CPU cores, then the remaining 4 cores of licensing capacity cannot be used simultaneously for another client machine.

MobileTogether Server licenses

MobileTogether Server licenses are based on the number of CPU cores on the MobileTogether Server machine. Core licenses allow an unlimited number of MobileTogether Client devices to connect to the server. However, if you check the *Limit to single thread execution* check box, then only one mobile device will be able to connect to the MobileTogether Server at any time. This is useful for evaluation and small-scale testing. Note that, if, in this case, a second device connects to MobileTogether Sever, then it will take over the license. The first device will not be able to connect any more and will receive an error message to this effect.

3.3 Notes about Environment

Folders

Given below is a list of important folders in your MapForce Server setup.

- *Installation root*

  /opt/Altova/MapForceServer2017/

- *License Files*

  /var/opt/Altova/MapForceServer

- *Environment settings*

  /etc/profile.d/jdbc.sh
  The environment settings file must be defined according to your specific environment. The example path above serves only as a general guide.
  Note: The environment settings file sets the variables for **all users** on the system.

JDBC-Connections

Note the following points:

- The Java Runtime Environment or SDK must be installed.
- The JDBC drivers for the target database must be installed.
- The following environment variables must be set correctly for your environment:
  - `CLASSPATH`: to find the jar-files
  - `PATH`: to find the JRE, but might not be necessary depending on the installation
  - `JAVA_HOME`: if necessary, depending on the installation.

Note
On Linux servers, the only database connections supported are JDBC.

Chapter 4

Setup on Mac OS X

4 Setup on Mac OS X

This section describes the installation and licensing of MapForce Server on Mac OS X systems.

Installation on Mac OS X

- System requirements
- Uninstall old versions of Altova server products
- Download the Mac OS X package
- Install MapForce Server
- Altova LicenseServer
- LicenseServer versions
- Trial license

Licensing on Mac OS X

- Start LicenseServer
- Register MapForce Server
- Assign a license

Notes about Environment

4.1 Installation on Mac OS X

MapForce Server is available for installation on Mac OS X systems. Its installation and setup procedure is described below.

- System requirements
  - *Mac OS X*

    Mac OS X 10.10, 10.11, 10.12 or newer

- FlowForce Server integration

  If you install MapForce Server together with FlowForce Server, it is recommended that you install FlowForce Server first. If you install MapForce Server before FlowForce Server, then, after having installed both MapForce Server and FlowForce Server, run the following command:

  ```
  cp /usr/local/Altova/MapForceServer2017/etc/*.tool /usr/local/Altova/
  FlowForceServer2017/tools
  ```

  This command copies the **.tool** file from **/etc** directory of MapForce Server to the FlowForce Server **/tools** directory. The **.tool** file is required by FlowForce Server; it contains the path to the MapForce Server executable. You do not need to run this command if you install FlowForce Server before installing MapForce Server.

- Uninstall old versions of Altova server products

  In the Applications folder in Finder, right-click the MapForce Server icon and select **Move to Trash**. The application will be moved to Trash. You will, however, still need to remove the application from the `usr` folder. Do this with the command:

  ```
  sudo rm -rf /usr/local/Altova/MapForceServer2017/
  ```

  If you need to uninstall an old version of Altova LicenseServer, you must first stop it running as a service. Do this with the following command:

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

  To check whether the service has been stopped, open the Activity Monitor in Finder and make sure that LicenseServer is not in the list. Then proceed to uninstall in the same way as described above for MapForce Server.

- Download the disk image file

  Download the disk image (.dmg) file from the Altova website (http://www.altova.com/download.html).

- Install MapForce Server

  Click to open the downloaded disk image (.dmg). This causes the MapForce Server installer

to appear as a new virtual drive on your computer. On the new virtual drive, double-click the installer package (.pkg). Go through the successive steps of the installer wizard. These are self-explanatory and include one step in which you have to agree to the license agreement before being able to proceed. To eject the drive after installation, right-click it and select **Eject**.

The MapForce Server package will be installed in the folder:
`/usr/local/Altova/MapForceServer2017` (application binaries)
`/var/Altova/MapForceServer` (data files: database and logs)

Altova LicenseServer

In order for any Altova Server product—including MapForce Server—to run, that server product must be licensed via an Altova LicenseServer on your network.

The Altova LicenseServer installation package is available on the virtual drive you have mounted in the previous step. To install Altova LicenseServer, double-click the installer package included on the virtual drive and follow the on-screen instructions. You will need to accept the license agreement for installation to proceed.

Altova LicenseServer can also be downloaded and installed separately from the Altova website (http://www.altova.com/download.html).

The LicenseServer package will be installed in the folder:
`/usr/local/Altova/LicenseServer`

For information about how to register MapForce Server with Altova LicenseServer and license it, see the section, Licensing on Mac OS X.

LicenseServer versions

- Altova server products must be licensed either with the version of LicenseServer that is appropriate to the installed MapForce Server version, or with a later version of LicenseServer.
- The LicenseServer version that is appropriate for a particular version of MapForce Server is displayed during the installation of MapForce Server. You can install this version of LicenseServer along with MapForce Server, or you can install LicenseServer separately.
- Before installing a newer version of LicenseServer, any older one must be de-installed. The LicenseServer installer will do this automatically if it detects an older version.
- LicenseServer versions are backwards compatible. They will work with older versions of MapForce Server.
- If you install a new version of MapForce Server and if your installed LicenseServer version is older than the appropriate LicenseServer, install the latest version available on the Altova website.
- At the time of LicenseServer de-installation, all registration and licensing information held in the older version of LicenseServer will be saved to a database on your server machine. This data will be imported automatically into the newer version when the newer version is installed.

- The version number of the currently installed LicenseServer is given at the bottom of the LicenseServer configuration page (all tabs).

  Current version: 2.3

▼ Trial license

During the installation process, you will be given the option of requesting a 30-day trial license for MapForce Server. After submitting the request, a trial license will be sent to the email address you registered.

4.2 Licensing on Mac OS X

MapForce Server must be licensed with an Altova LicenseServer in order to run it. Licensing is a two-step process:

1. **Register MapForce Server** with LicenseServer. Registration is done from MapForce Server.
2. **Assign a license** to MapForce Server. License-assigning is done from LicenseServer.

The steps you need to carry out are given below.

▼ Start LicenseServer

To correctly register and license MapForce Server with LicenseServer, LicenseServer must be running as a daemon. Start LicenseServer as a daemon with the following command:

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

If at any time you need to stop LicenseServer, replace **load** with **unload** in the above command:

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

▼ Register MapForce Server

To register MapForce Server from the command line interface, use the `licenseserver` command:

```
sudo /usr/local/Altova/MapForceServer2017/bin/MapForceServer
licenseserver [options] ServerName-Or-IP-Address
```

For example, if `localhost` is the name of the server on which LicenseServer is installed:

```
sudo /usr/local/Altova/MapForceServer2017/bin/MapForceServer
licenseserver localhost
```

In the command above, `localhost` is the name of the server on which LicenseServer is installed. Notice also that the location of the MapForce Server executable is:

```
/usr/local/Altova/MapForceServer2017/bin/
```

After successful registration, go to the Server Management tab of LicenseServer's configuration page to assign a license to MapForce Server.

▼ Assign a license

After successfully registering MapForce Server, it will be listed in the Server Management tab of the configuration page of LicenseServer. Go there and assign a license to MapForce Server.

Note on cores and licenses

The licensing of Altova server products is based on the number of processor cores available

on the product machine. For example, a dual-core processor has two cores, a quad-core processor four cores, a hexa-core processor six cores, and so on. The number of cores licensed for a product must be greater than or equal to the number of cores available on that server machine, whether the server is a physical or virtual machine. For example, if a server has eight cores (an octa-core processor), you must purchase at least one 8-core license. You can also combine licenses to achieve the core count. So, two 4-core licenses can also be used for an octa-core server instead of one 8-core license.

If you are using a computer server with a large number of CPU cores but only have a low volume to process, you may also create a virtual machine that is allocated a smaller number of cores, and purchase a license for that number. Such a deployment, of course, will have less processing speed than if all available cores on the server were utilized.

Note: Each Altova server product license can be used for only one client machine—the machine on which the Altova server product is installed—at a time, even if the license has unused licensing capacity. For example, if a 10-core license is used for a client machine that has 6 CPU cores, then the remaining 4 cores of licensing capacity cannot be used simultaneously for another client machine.

MobileTogether Server licenses

MobileTogether Server licenses are based on the number of CPU cores on the MobileTogether Server machine. Core licenses allow an unlimited number of MobileTogether Client devices to connect to the server. However, if you check the *Limit to single thread execution* check box, then only one mobile device will be able to connect to the MobileTogether Server at any time. This is useful for evaluation and small-scale testing. Note that, if, in this case, a second device connects to MobileTogether Sever, then it will take over the license. The first device will not be able to connect any more and will receive an error message to this effect.

4.3 Notes about Environment

Folders

Given below is a list of important folders in your MapForce Server setup.

- *Installation root*

  /usr/local/Altova/MapForceServer2017/

- *License Files*

  /var/Altova/MapForceServer

- *Environment settings*

  /Library/LaunchDaemons/com.altova.MobileTogetherServer.plist
  The environment settings file must be defined according to your specific environment. The example path above serves only as a general guide.
  Note: These environment variables are only set for the MapForce Server process and do not have an impact on other users.

JDBC-Connections

Note the following points:

- The Java Runtime Environment or SDK must be installed.
- The JDBC-Connects for the target database must be installed.
- The following environment variables must be set correctly for your environment:
  - **CLASSPATH:** to find the jar-files
  - **PATH:** to find the JRE, but might not be necessary depending on the installation
  - **JAVA_HOME:** if necessary, depending on the installation

> **Note**
> On Mac OS X servers, the only database connections supported are JDBC.

Chapter 5

Command Line Usage

5 Command Line Usage

This section:

Default location of MapForce Server executable

Given below are the default locations of the MapForce Server executable:

Windows `<ProgramFilesFolder>\Altova\MapForceServer2017\bin\MapForceServer.exe`

Linux `/opt/Altova/MapForceServer2017/bin/mapforceserver`

Mac `/usr/local/Altova/MapForceServer2017/bin/mapforceserver`

Usage and list of CLI commands

General command line syntax for `MapForceServer` is:

Windows **`MapForceServer --h | --help | --version | <command> [options] [arguments]`**

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

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

▼ Casing and slashes on the command line

`MapForceServer` *on Windows*
`mapforceserver` *on Unix (Linux, Mac)*

* Note that lowercase (`mapforceserver`) works on all platforms (Windows, Linux, and Mac), while upper-lower (`MapForceServer`) works only on Windows and Mac.
* Use forward slashes on Linux and Mac, backslashes on Windows.

where

`--h | --help`	Displays the help text.
`--version`	Displays the version of MapForce Server.

Valid CLI commands are listed below and are explained in the sub-sections of this section.

licenseserver	Register MapForce Server with LicenseServer on the local network.
assignlicense	Upload a license to LicenseServer and assign it to MapForce Server on this machine.
verifylicense	Check if the current MapForce Server is licensed; optionally, check if a given license key is assigned.
run	Executes a MapForce Server Execution file (`.mfx`)
exportresourcestrings	Exports all application resource strings to an XML file.
setdeflang | sdl	Sets the default language.
help	Displays help for a specific command. For example: `help run`

5.1 licenseserver

This section:

Command syntax

Windows	**MapForceServer licenseserver [options] *Server-Or-IP-Address***
Linux	**mapforceserver licenseserver [options] *Server-Or-IP-Address***
Mac	**mapforceserver licenseserver [options] *Server-Or-IP-Address***

Command description

On execution, the `licenseserver` command registers MapForce Server with the LicenseServer specified by the *`Server-Or-IP-Address`* argument. ***Server-Or-IP-Address*** identifies the machine on the network on which Altova LicenseServer is installed and running. It can be the machine's name or its IP address.

For the `licenseserver` command to be executed successfully, the two servers must be connected on the network and LicenseServer must be running. You must have administrator privileges to be able to register MapForce Server with LicenseServer.

Once MapForce Server has been successfully registered with LicenseServer, you will receive a message to this effect. The message will also display the URL of the LicenseServer. You can now go to LicenseServer to assign MapForce Server a license. *For details, see the LicenseServer documentation*.

Examples

Examples of the `licenseserver` command:

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

The commands above specifiy, respectively, the machine named `DOC.altova.com`, and the user's

machine (`localhost` and `127.0.0.1`) as the machine running Altova LicenseServer. In each case, the command registers MapForce Server with the LicenseServer on the machine specified.

- Casing and slashes on the command line

  `MapForceServer` *on Windows*
  `mapforceserver` *on Unix (Linux, Mac)*

  * Note that lowercase (`mapforceserver`) works on all platforms (Windows, Linux, and Mac), while upper-lower (`MapForceServer`) works only on Windows and Mac.
  * Use forward slashes on Linux and Mac, backslashes on Windows.

Command options

Options are listed in their short forms (in the first column) and long forms (second column), together with their descriptions (third column). On the command line, one or two dashes can be used for both short and long forms.

`--j`	`--json`	Prints the result of the registration attempt as a machine-parsable JSON object. Form: `--json=`*`true`*`|`*`false`*

5.2 assignlicense (Windows only)

This section:

Command syntax

Windows **`MapForceServer assignlicense [options] FILE`**

Note: This command is supported only on Windows systems. It is not supported on Linux or Mac systems.

Command description

On execution, the `assignlicense` command uploads the license file specified by the *`FILE`* argument to the registered LicenseServer, and assigns the license to MapForce Server on this machine. The *`FILE`* argument takes the filepath of the license file.

The `--test-only` option allows you to upload to LicenseServer and validate the license, without assigning the license to MapForce Server. *For details about licensing, see the LicenseServer documentation*.

Examples

Examples of the `assignlicense` command:

```
mapforceserver assignlicense C:\MapForce Server12345.altova_licenses
mapforceserver assignlicense --test-only=true C:\MapForce
Server12345.altova_licenses
```

The first command above uploads the specified license to LicenseServer and assigns it to MapForce Server.
The second command uploads the specified license to LicenseServer and validates it, without assigning it to MapForce Server.

▼ Casing and slashes on the command line

`MapForceServer` *on Windows*

`mapforceserver` *on Unix (Linux, Mac)*

* Note that lowercase (`mapforceserver`) works on all platforms (Windows, Linux, and Mac), while upper-lower (`MapForceServer`) works only on Windows and Mac.
* Use forward slashes on Linux and Mac, backslashes on Windows.

Command options

Options are listed in their short forms (in the first column) and long forms (second column), together with their descriptions (third column). On the command line, one or two dashes can be used for both short and long forms.

`--t`	`--test-only`	Values are `true|false`. If `true`, then the license file is only uploaded to LicenseServer and validated. It is not assigned to MapForce Server. Form: `--test-only=true|false`. Set to `true` if the option is specified without a value.

5.3 verifylicense (Windows only)

This section:

Command syntax

Windows **`MapForceServer verifylicense [options]`**

Note: This command is supported only on Windows systems. It is not supported on Linux or Mac systems.

Command description

Checks whether the current product is licensed. Additionally, the `--license-key` option enables you to check whether a specific license key is already assigned to the product. *For details about licensing, see the LicenseServer documentation*.

Examples

Examples of the `verifylicense` command:

```
mapforceserver verifylicense
mapforceserver verifylicense --license-key=ABCDEFG-HIJKLMN-OPQRSTU-VWXYZ12-
3456789
```

The commands above check, respectively, whether the product has been licensed, and whether the product has been licensed with the license key given as the value of the `--license-key` option.

▼ Casing and slashes on the command line

`MapForceServer` *on Windows*
`mapforceserver` *on Unix (Linux, Mac)*

* Note that lowercase (`mapforceserver`) works on all platforms (Windows, Linux, and Mac), while upper-lower (`MapForceServer`) works only on Windows and Mac.
* Use forward slashes on Linux and Mac, backslashes on Windows.

Command options

Options are listed in their short forms (in the first column) and long forms (second column), together with their descriptions (third column). On the command line, one or two dashes can be used for both short and long forms.

`--l`	`--license-key`	Checks whether the submitted license key has already been assigned to the product. Form: `--license-key=`*VALUE*

5.4 run

This section:

Command syntax

Windows	**MapForceServer run [options] MfxFile**
Linux	**mapforceserver run [options] MfxFile**
Mac	**mapforceserver run [options] MfxFile**

Command description

The `run` command executes a MapForce Server execution file (`.mfx` file). It requires an input `.mfx` file (**`MfxFile`**) as its argument.

Using stream redirection at the command line

If the mapping uses an output component to return a simple value such as string, this output is written in the `stdout` (standard output) stream. On the other hand, the success and error messages are available in the `stderr` (standard error) stream. If you do not want the standard output stream to be displayed on the screen together with the success or error messages, redirect either the standard output or the standard error stream (or both) to files. If neither the `stdout` nor the `stderr` streams are redirected, they are both displayed on the screen, combined.

For example, to redirect the standard output stream to a file, use:

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

To redirect the standard error stream to a file, use:

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

To redirect both streams simultaneously, use:

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

For further information about stream redirection, refer to the documentation of your operating

system's command shell.

Examples

In Altova MapForce, open a MapForce Design file (`.mfd` file) from the MapForce Examples folder and compile it to a MapForce Execution file (`.mfx` file) with the MapForce command **File | Compile to MapForce Server Execution File**. For the scope of this example, let's assume you selected the file `ChainedPersonList.mfd` and saved it to `C:\temp\ChainedPersonList.mfx`.

Now, in MapForce Server, you can use the `run` command to execute `ChainedPersonList.mfx`:

```
mapforceserver run C:\temp\ChainedPersonList.mfx
```

The two output files of this MapForce Design file (`PersonList.xml` and `Contacts.xml`) are generated in the MapForce Examples folder.

▼ Casing and slashes on the command line

`MapForceServer` *on Windows*
`mapforceserver` *on Unix (Linux, Mac)*

* Note that lowercase (`mapforceserver`) works on all platforms (Windows, Linux, and Mac), while upper-lower (`MapForceServer`) works only on Windows and Mac.
* Use forward slashes on Linux and Mac, backslashes on Windows.

Command options

Options are listed in their short forms (in the first column) and long forms (second column), together with their descriptions (third column). On the command line, one or two dashes can be used for both short and long forms.

For information about global resources and input parameters, see the Altova MapForce documentation.

	`--catalog`	Specifies the absolute path to a root catalog file that is not the installed root catalog file. The default value is the absolute path to the installed root catalog file. Form: `--catalog=FILE`
`--cert`	`--certificatespath`	This command is applicable to MapForce Server running on Linux. It specifies the path to the directory where any certificate files required by the mapping are stored. Form: `--certificatespath=DIRECTORY`

		See also Digital Certificate Management.
--gc	--globalresourceconfig	The name of the global resource configuration. Form: --gc=VALUE
--gr	--globalresourcefile	The path of the global resource definition file. Form: --gr=FILE.
--l	--lang	The language used for displaying messages. Form: --lang=VALUE (en,de,ja,es, fr)
--p	--param	Assigns a value to a parameter defined in the mapping. Form: --param=*ParamName:ParamValue*. The --param switch must be used before each parameter. Use quotes if *ParamName* or *ParamValue* contains a space. For example: --p=company:"Nanonull Inc".
	--taxonomy-package	Specifies the absolute path to an additional XBRL taxonomy package as described in the Taxonomy Packages 1.0 recommendation. The value of FILE gives the location of the taxonomy package. Add the option multiple times to specify more than one taxonomy package. Form: --taxonomy-package=FILE
	--taxonomy-packages-config-file	Specifies the path to a configuration file called **TaxonomyPackagesConfig.json**, used to load XBRL taxonomy packages. This configuration file is updated every time when you add, remove, activate, or deactivate XBRL taxonomy packages from the graphical user interface of Altova XMLSpy, MapForce, or StyleVision. If you have added custom XBRL taxonomy packages using one of the products above, the file is located at **C:\Users\<username>\Documents\Altova**. Form: --taxonomy-packages-config-file=FILE

Note: On Windows systems: Avoid using the end backslash and closing quote on the command line \", for example, as in: "C:\My directory\". These two characters are interpreted by the command line parser as a literal double-quotation mark. Use the double backslash \\ if spaces occur in the command line and you need the quotes (for example: "C:\My Directory\\"). Alternatively, try to avoid using spaces and, therefore, quotes at all.

5.5 exportresourcestrings

This section:

Command syntax

Windows	**MapForceServer exportresourcestrings *LanguageCode XMLOutputFile***
Linux	**mapforceserver exportresourcestrings *LanguageCode XMLOutputFile***
Mac	**mapforceserver exportresourcestrings *LanguageCode XMLOutputFile***

Arguments

The `exportresourcestrings` command takes the following arguments:

`LanguageCode`	Specifies the language of resource strings in the exported XML file. Supported languages are: `en`, `de`, `es`, `fr`, `ja`
`XMLOutputFile`	Specifies the location and name of the exported XML file.

Command description

The **`exportresourcestrings`** command outputs an XML file containing the resource strings of the MapForce Server application. It takes two arguments: (i) the language of the resource strings in the output XML file, and (ii) the path and name of the output XML file. Allowed export languages (with their language codes in parentheses) are: English (`en`), German, (`de`), Spanish (`es`), French (`fr`), and Japanese (`ja`).

Examples

An example of the `exportresourcestrings` command:

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

This command creates a file called `Strings.xml` at `c:\` that contains all the resource strings of the MapForce Server application in German.

- Casing and slashes on the command line

  `MapForceServer` *on Windows*
  `mapforceserver` *on Unix (Linux, Mac)*

  * Note that lowercase (`mapforceserver`) works on all platforms (Windows, Linux, and Mac), while upper-lower (`MapForceServer`) works only on Windows and Mac.
  * Use forward slashes on Linux and Mac, backslashes on Windows.

Creating localized versions of MapForce Server

You can create a localized version of MapForce Server for any language of your choice. Five localized versions (English, German, Spanish, French, and Japanese) are already available in the `C:\Program Files (x86)\Altova\MapForceServer2017\bin` folder, and therefore do not need to be created.

Create a localized version as follows:

1. Generate an XML file containing the resource strings by using the `exportresourcestrings` command (*see command syntax above*). The resource strings in this XML file will be one of the five supported languages: English (`en`), German (`de`), Spanish (`es`), French (`fr`), or Japanese (`ja`), according to the ***LanguageCode*** argument used with the command.
2. Translate the resource strings from one of the four supported languages into the target language. The resource strings are the contents of the `<string>` elements in the XML file. Do not translate variables in curly brackets, such as `{option}` or `{product}`.
3. Contact Altova Support to generate a localized MapForce Server DLL file from your translated XML file.
4. After you receive your localized DLL file from Altova Support, save the DLL in the `C:\Program Files (x86)\Altova\<%APPFOLDER\bin` folder. Your DLL file will have a name of the form `MapForceServer2017_lc.dll`. The `_lc` part of the name contains the language code. For example, in `MapForceServer2017_de.dll`, the `de` part is the language code for German (Deutsch).
5. Run the `setdeflang` command to set your localized DLL file as the MapForce Server application to use. For the argument of the `setdeflang` command, use the language code that is part of the DLL name.

Note: Altova MapForce Server is delivered with support for five languages: English, German, Spanish, French, and Japanese. So you do not need to create a localized version of these languages. To set any of these languages as the default language, use MapForce Server's `setdeflang` command.

5.6 setdeflang

This section:

Command syntax

Windows	**MapForceServer setdeflang | sdl *LangaugeCode***
Linux	**mapforceserver setdeflang | sdl *LangaugeCode***
Mac	**mapforceserver setdeflang | sdl *LangaugeCode***

Command description

The **setdeflang** command (short form is `sdl`) sets the default language of MapForce Server. It takes a mandatory `LanguageCode` argument.

Examples

An example of the `setdeflang` command:

```
mapforceserver setdeflang DE
```

The command above sets the default language for messages to German.

▼ Casing and slashes on the command line

MapForceServer *on Windows*
mapforceserver *on Unix (Linux, Mac)*

* Note that lowercase (`mapforceserver`) works on all platforms (Windows, Linux, and Mac), while upper-lower (`MapForceServer`) works only on Windows and Mac.
* Use forward slashes on Linux and Mac, backslashes on Windows.

Supported languages

The table below lists the languages currently supported together with their language codes.

EN	English
DE	German
ES	Spanish
FR	French
JA	Japanese

5.7 help

This section:

Command syntax

Windows	**MapForceServer help *Command***
Linux	**mapforceserver help *Command***
Mac	**mapforceserver help *Command***

Command description

The **help** command takes a single argument (*Command*): the name of the command for which help is required. It displays the correct syntax of the command and other information relevant to the correct execution of the command.

Examples

An example of the `help` command:

```
mapforceserver help exportresourcestrings
```

The command above contains one argument: the command `exportresourcestrings`, for which help is required. When the example command above is executed, information about the `exportresourcestrings` command will be displayed in the terminal.

▼ Casing and slashes on the command line

MapForceServer *on Windows*
mapforceserver *on Unix (Linux, Mac)*

* Note that lowercase (`mapforceserver`) works on all platforms (Windows, Linux, and Mac), while upper-lower (`MapForceServer`) works only on Windows and Mac.
* Use forward slashes on Linux and Mac, backslashes on Windows.

The `--help` option

Help information about a command is also available by using the `--help` option with the command for which help information is required. For example, using the `--help` option with the `exportresourcestrings` command, as follows:

```
mapforceserver exportresourcestrings --help
```

achieves the same result as does using the `help` command with an argument of `exportresourcestrings`:

```
mapforceserver help exportresourcestrings
```

In both cases, help information about the `exportresourcestrings` command is displayed.

Chapter 6

MapForce Server API

6 MapForce Server API

MapForce Server provides an application programming interface (API) that you can access programmatically from your .NET, Java, or COM-based code.

For an introduction to each platform, refer to the following topics:

- About the .NET Interface
- About the COM Interface
- About the Java Interface

For code examples, refer to the Code Examples section.

For a technical description of the API, refer to the following topics:

- API Reference (COM, .NET)
- API Reference (Java)

6.1 About the .NET Interface

The .NET interface is built as a wrapper around the COM interface. It is provided as a primary interop assembly signed by Altova and uses the namespace `Altova.MapForceServer`.

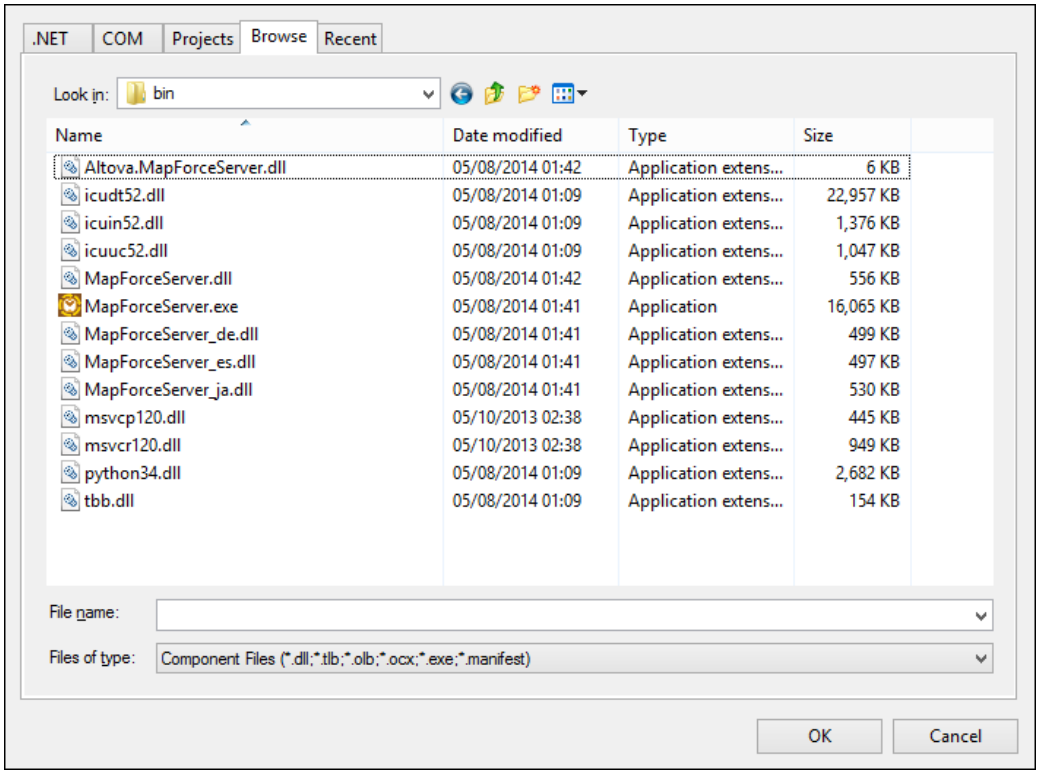
During installation, MapForce Server will be registered automatically as a COM server object, so there is no need for a manual registration. If you receive an access error, open the Component Services and give permissions to the same account that runs the application pool containing MapForce Server.

In order to use MapForce Server in your .NET project, add a reference to the `Altova.MapForceServer.dll` file (see the instructions below). The `Altova.MapForceServer.dll` is located in the `bin` folder of the MapForce Server installation folder. This .dll file is automatically added to the global assembly cache (GAC) during MapForce Server installation (the GAC is typically located in the `C:\WINDOWS\assembly` folder).

Once MapForce Server has been registered as a COM server object, and the `Altova.MapForceServer.dll` is available to the .NET interface, the MapForce API functionality becomes available in your .NET project.

To add a reference to the MapForce Server DLL in a Visual Studio .NET project:

1. With the .NET project open in Visual Studio, click **Project | Add Reference**. The Add Reference dialog box pops up.



2. On the Browse tab, browse for the folder: *<MapForceServer application folder>*`/bin`, select the `Altova.MapForceServer.dll`, and click **OK**.

You can view the structure of the `Altova.MapForceServer` assembly using the Visual Studio Object Browser (to display the Object Browser, click **Object Browser** on the **View** menu).

6.2 About the COM Interface

MapForce Server is automatically registered as a COM server object during installation. To check whether the registration was successful, open the Registry Editor (for example, by typing `regedit.exe` command at the command line). If registration was successful, the Registry will contain the classes `MapForce.Server`. These two classes will typically be found under `HKEY_LOCAL_MACHINE\SOFTWARE\Classes`.

Once the COM server object is registered, you can invoke it from within applications and scripting languages that have programming support for COM calls. If you wish to change the location of the MapForce Server installation package, it is best to uninstall MapForce Server and then reinstall it at the required location. In this way, the necessary de-registration and registration are carried out by the installer process.

6.3 About the Java Interface

To access the MapForce Server API from Java code, the following references must be added to the .classpath file of your Java project.

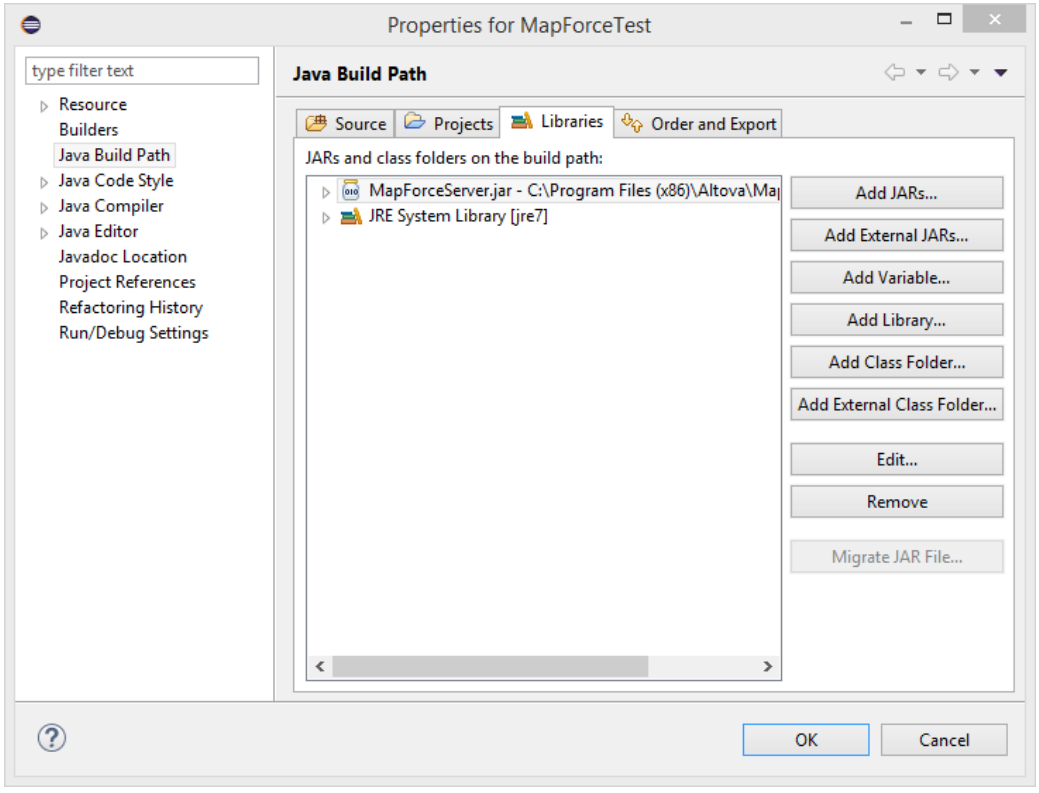
`MapForceServer.jar`	The library that communicates with MapForce Server.
`MapForceServer_JavaDoc.zip`	A Javadoc archive file containing documentation for the Java MapForce Server API.

Both files are available in the `bin` folder of the MapForce Server installation folder. You can either reference the file from their original location or copy them to another location if this fits your project setup.

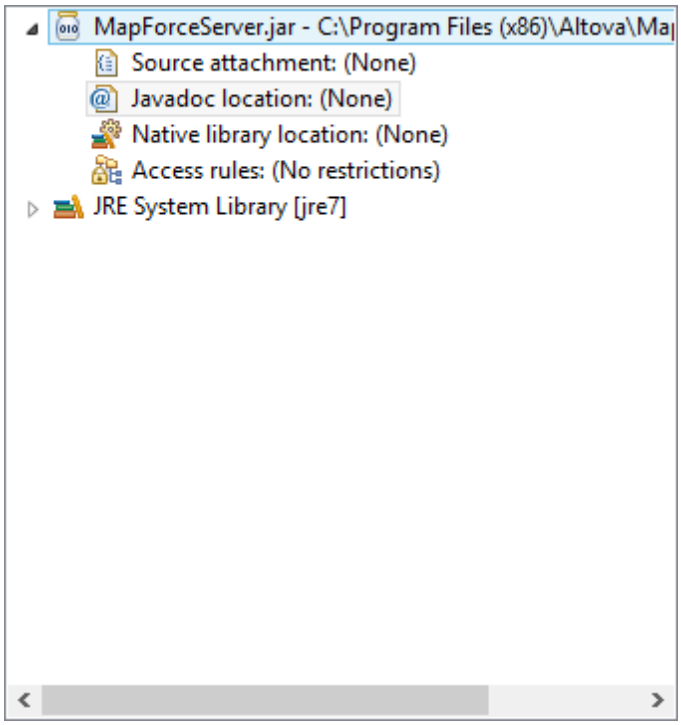
In Eclipse, you can add the classpath references by editing the properties of the Java project, as shown in the sample instructions below (notice the instructions apply to Eclipse 4.4).

To add the MapForce Server library references in Eclipse:

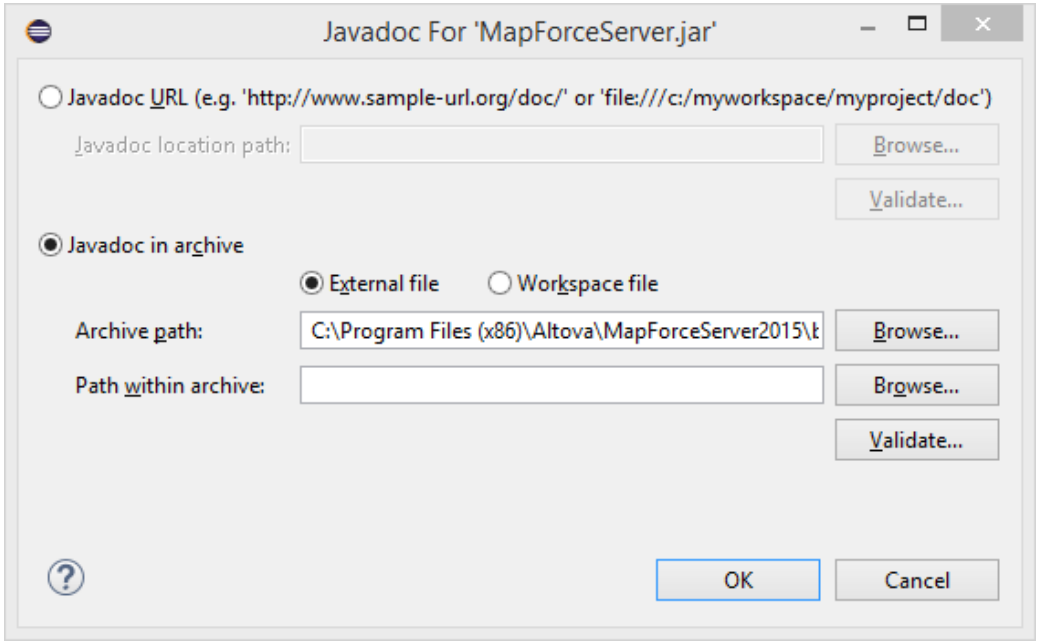
1. With the project open in Eclipse, on the **Project** menu, click **Properties**.
2. On the **Libraries** tab, click **Add External JARs**, and then browse for the `MapForceServer.jar` file located in the MapForce Server installation folder.



3. Under **JARs and class folders on the build path**, expand the **MapForceServer.jar** record, and then double click on the **Javadoc location: (None)** record.



4. Ensure that the **Javadoc in archive** and **External file** options are selected, and then browse for the `MapForceServer_JavaDoc.zip` file located in the MapForce Server installation folder.



5. Click OK. The reference to the MapForce Server library and Javadoc archive has now been added to the .classpath file of the project.

   Below is an example of how the .classpath file might look if you are referencing the files from the original installation folder, on a 64-bit operating system running 32-bit MapForce

Server (the relevant lines are highlighted in yellow):

```
<?xml version="1.0" encoding="UTF-8"?>
<classpath>
	<classpathentry kind="src" path="src"/>
	<classpathentry kind="con"
path="org.eclipse.jdt.launching.JRE_CONTAINER/
org.eclipse.jdt.internal.debug.ui.launcher.StandardVMType/jre7"/>
	<classpathentry kind="lib" path="C:/Program Files (x86)/Altova/
MapForceServer2017/bin/MapForceServer.jar">
		<attributes>
			<attribute name="javadoc_location"
value="jar:file:/C:/Program%20Files%20(x86)/Altova/MapForceServer2017/
bin/MapForceServer_JavaDoc.zip!/"/>
		</attributes>
	</classpathentry>
	<classpathentry kind="output" path="bin"/>
</classpath>
```

6.4 Code Examples

This section includes sample code which illustrates how to run a MapForce mapping programmatically. Samples are included for the following languages:

- C#
- C++
- Java
- VBScript
- Visual Basic
- Visual Basic for Applications (VBA)

6.4.1 C#

The following example illustrates how to run a mapping execution file (.mfx) from C# code.

Before testing this example, ensure the following prerequisites are met:

- MapForce Server is installed and licensed
- A reference to the MapForce Server DLL has been added in your Visual Studio project (see About the .NET Interface).
- If you have installed MapForce Server 64-bit, then the application which calls the API (such as the sample one below) must also be built for the 64-bit platform in Visual Studio. Also, the path to the MapForce server executable must be adjusted accordingly in the code.

```
namespace MapForceServerAPI_sample
{
    class Program
    {
        static void Main(string[] args)
        {
            //Create a MapForce Server object
            Altova.MapForceServer.Server objMFS = new
Altova.MapForceServer.Server();

            //Set a working directory used as a base for relative paths
            objMFS.WorkingDirectory = "c:\\temp";

            //Default path to the MapForce Server executable is the
installation path (same dir as the MapForceServer.dll)
            //If 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\
\MapForceServer2017\\bin\\MapForceServer.exe";

            //Prepare the parameters, if your mapping 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\
\MapForceServer2017\\etc\\Examples\\CompletePO.mfx"))
                System.Console.WriteLine("Success - finished execution");
            else
                System.Console.WriteLine(objMFS.LastExecutionMessage);
        }
    }
}
```

6.4.2 C++

The following example illustrates how to run a mapping execution file (.mfx) from C++ code.

Before testing this example, ensure the following prerequisites are met:

- MapForce Server is installed and licensed
- MapForce Server is available as a COM server object (normally, this process takes place automatically during MapForce Server installation; to check if registration was successful, see About the COM Interface).

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


// 32-bit MapForce Server
#import "progid:MapForce.Server"
// 64-bit MapForce Server
//#import "progid:MapForce_x64.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:\\temp";

		//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\
\MapForceServer2017\\bin\\MapForceServer.exe";

		//Prepare the parameters, if your mapping uses parameters
		//pMFS->AddParameter( "testparam1", "value 1" );

		//Run the mapping; the output will be stored at C:\temp
\ExpReport.rtf
		// NOTE Please adapt the path to the input file in order to run
the sample
		if (pMFS->Run("C:\\Program Files (x86)\\Altova\
```

```
\MapForceServer2017\\etc\\Examples\\CompletePO.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;

}
```

6.4.3 Java

The following example illustrates how to run a mapping execution file (.mfx) from Java code.

Before testing this example, ensure the following prerequisites are met:

- MapForce Server is installed and licensed
- MapForce Server libraries are added to the .classpath file in your project (for an example, see About the Java Interface).

```
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
         objMFS.setWorkingDirectory( "C:\\temp" );

         //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\
\MapForceServer2017\\bin\\MapForceServer.exe" );

         //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\
\MapForceServer2017\\etc\\Examples\\CompletePO.mfx" ) )
            System.out.println( "Success - finished execution" );
         else
            System.out.println( objMFS.getLastExecutionMessage() );
      }
      catch ( Exception e)
      {
         e.printStackTrace();
      }
   }

}
```

6.4.4 VBScript

The following example illustrates how to run a mapping execution file (.mfx) from VBScript code.

Before testing this example, ensure the following prerequisites are met:

- MapForce Server is installed and licensed
- MapForce Server is available as a COM server object (normally, this process takes place automatically during MapForce Server installation; to check if registration was successful, see About the COM Interface).

```
Option Explicit

'Create a MapForce Server object; use "MapForce_x64.Server" if you want to
use the 64-bit installation
Dim objMFS
Set objMFS = WScript.GetObject( "", "MapForce.Server" )

'Set a working directory - used as a base for relative paths
objMFS.WorkingDirectory = "C:\temp"

'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\MapForceServer2017\bin
\MapForceServer.exe"

'Prepare the parameters, if your mapping uses parameters
'Call 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\MapForceServer2017\etc
\Examples\CompletePO.mfx" ) ) Then
	WScript.Echo( "Success - finished execution" )
Else
	WScript.Echo( objMFS.LastExecutionMessage )
End If
```


6.4.5 Visual Basic

The following example illustrates how to run a mapping execution file (.mfx) from Visual Basic code.

Before testing this example, ensure the following prerequisites are met:

- MapForce Server is installed and licensed
- A reference to the MapForce Server DLL has been added in your Visual Studio project (see About the .NET Interface).

```
Option Explicit On

Module Program

    Sub Main()

        'Create a MapForce Server object; use "MapForce_x64.Server" if you
want to use the 64-bit installation
        Dim objMFS As Altova.MapForceServer.Server = New
Altova.MapForceServer.Server

        'Set a working directory - used as a base for relative paths
        objMFS.WorkingDirectory = "C:\temp"

        '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
\MapForceServer2017\bin\MapForceServer.exe"

        'Prepare the parameters, if your mapping 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\MapForceServer2017\etc
\Examples\CompletePO.mfx")) Then
            System.Console.WriteLine("Success - finished execution")
        Else
            System.Console.WriteLine(objMFS.LastExecutionMessage)
        End If
    End Sub

End Module
```

6.4.6 Visual Basic for Applications (VBA)

Microsoft Visual Basic for Applications (VBA) is primarily used for automating tasks in Microsoft Office. However, it is also possible to call the MapForce Server API from VBA and execute mappings. The following instructions have been tested on MapForce Server and the VBA for Microsoft Office 2013. Instructions may differ if you are using another VBA development environment.

Prerequisites

Before you can call the MapForce Server API functions from your VBA project, note the following prerequisites:

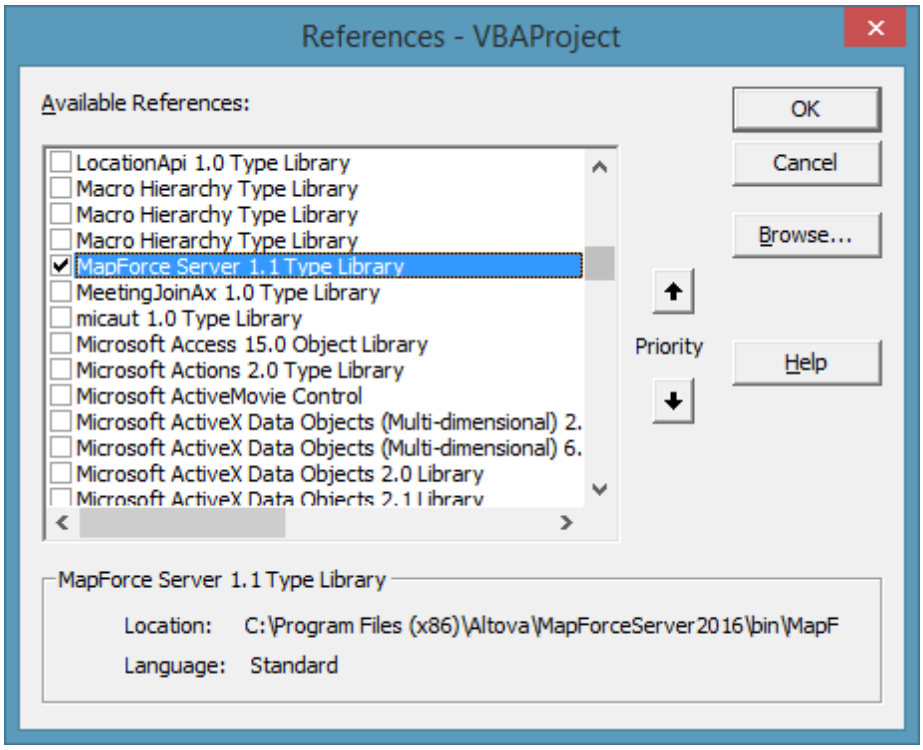
1. Microsoft Office and MapForce Server must be installed on the same machine.
2. The architecture of MapForce Server (32-bit or 64-bit) must match that of Microsoft Office. For example, if you run VBA on Microsoft Office 32-bit, make sure that you use MapForce Server 32-bit. To find out whether your Office product runs on 64-bit, click the **File** tab, click **Account**, and then click "About Excel" (or "About Word").
3. The MapForce Server library must be referenced from your VBA project (see instructions below).

How to add a reference to the MapForce Server Library from your VBA project

1. In a macro-enabled Microsoft Office document (.docm, .xlsm), on the **Developer** tab, click **Visual Basic**.

   > By default, the **Developer** tab is not enabled in Microsoft Office. To enable the **Developer** tab in an Office 2013 program, right-click the ribbon and select **Customize the Ribbon** from the context menu. Then, in the Options dialog box, select the **Developer** check box under "Main Tabs".

2. In the VBA development environment, in the **Tools** menu, click **References**.



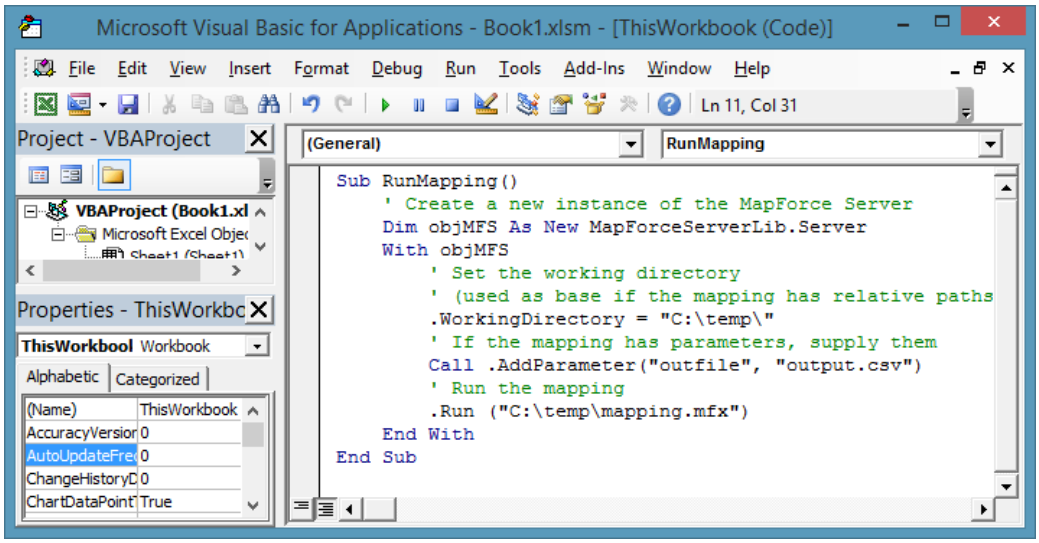
3. Click to select the **MapForce Server Type Library**.

How to call the MapForce Server API

Once you have added a reference to the MapForce Server Library in your VBA project, you can enter the VBA code in the Code Editor window. For example, the following sample code calls MapForce Server and runs a mapping executable file (**mapping.mfx**) that takes an input parameter called "outfile" having the value "output.csv".

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

Press **F5** to debug the VBA code and run the mapping.



6.5 API Reference (COM, .NET)

This section provides general reference to the MapForce Server API elements (such as interfaces and methods) applicable to code written for the COM or .NET platforms.

6.5.1 Interfaces

The MapForce Server API exposes the following interface:

- IServer

IServer Interface

Purpose

The `IServer` interface creates a new MapForce Server object instance, and provides access to the MapForce Server engine. The `IServer` interface has the following members:

- Methods
- Properties

Methods

AddParameter method

Assigns a value to a parameter defined in the mapping.

C#

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

C++

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

VB

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

ClearParameterList method

Clears the list of parameters.

C#

```
void ClearParameterList()
```

C++

```
HRESULT ClearParameterList();
```

VB

```
Sub ClearParameterList()
```

GetOutputParameter method

Gets the string output generated by the last `run` command. Returns null if no output was generated. This function requires a string parameter which identifies the name of the output component in MapForce.

C#

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

C++

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

VB

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

Run method

Executes a MapForce Server Execution file (.mfx file). Returns TRUE in case of success; FALSE otherwise.

C#

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

C++

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

VB

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

Properties

APIMajorVersion property

Gets the major version of the MapForce Server API. This can be different from the product version if the API is connected to another server.

C#

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

C++

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

VB

```
ReadOnly Property APIMajorVersion As Integer
```

APIMinorVersion property

Gets the minor version of the MapForce Server API. This can be different from the product version if the API is connected to another server.

C#

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

C++

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

VB

```
ReadOnly Property APIMinorVersion As Integer
```

APIServicePackVersion property

Gets the service pack version of the MapForce Server API. This can be different from the product version if the API is connected to another server.

C#

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

C++

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

VB

```
ReadOnly Property APIServicePackVersion As Integer
```

Is64Bit property

Returns TRUE if the MapForce Server engine is a 64-bit executable.

C#

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

C++

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

VB

```
ReadOnly Property Is64Bit As Boolean
```

LastExecutionMessage property

Gets the message received during the last Run command.

C#

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

C++

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

VB

```
ReadOnly Property LastExecutionMessage As String
```

MajorVersion property

Gets the major version of the product (for example, "16" for Altova MapForce Server 2014 r2 sp1 (x64)).

C#

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

C++

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

VB

```
ReadOnly Property MajorVersion As Integer
```

MinorVersion property

Gets the minor version of the product (for example, "2" for Altova MapForce Server 2014 r2 sp1 (x64)).

C#

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

C++

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

VB

```
ReadOnly Property MinorVersion As Integer
```

ProductName property

Gets the name of the product (for example, "Altova MapForce Server" for Altova MapForce Server 2014 r2 sp1 (x64)).

C#

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

C++

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

VB

```
ReadOnly Property ProductName As String
```

ProductNameAndVersion property

Gets the complete name and version of the product (for example, "Altova MapForce Server 2014 r2 sp1 (x64)").

C#

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

C++

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

VB

```
ReadOnly Property ProductNameAndVersion As String
```

ServerPath property

Gets or sets the path of the MapForce Server executable.

C#

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

C++

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

VB

```
Property ServerPath As String
```

ServicePackVersion property

Gets the service pack version of the product (for example, "1" for Altova MapForce Server 2014 r2 sp1 (x64)).

C#

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

C++

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

VB

```
ReadOnly Property ServicePackVersion As Integer
```

WorkingDirectory property

Gets or sets the current directory for running jobs (relative paths will be evaluated against the working directory).

C#

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

C++

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

VB

```
Property WorkingDirectory As String
```

6.6 API Reference (Java)

This section provides general reference to the MapForce Server API elements (such as classes and methods) applicable to code written for the Java platform.

6.6.1 Package com.altova.mapforceserver

The package `com.altova.mapforceserver` consists of the following classes:

- MapForceServer Class
- MapForceServerException Class

MapForceServer Class

Purpose

The `MapForceServer` class creates a new MapForce Server object instance, and provides access to the MapForce Server engine.

Methods

addParameter method

Assigns a value to a parameter defined in the mapping.

```
public void addParameter(java.lang.String name,
                         java.lang.String value)
```

clearParameterList method

Clears the list of parameters.

```
public void clearParameterList()
```

getAPIMajorVersion method

Gets the major version of the MapForce Server API. This can be different from the product version if the API is connected to another server.

```
public int getAPIMajorVersion()
```

getAPIMinorVersion method

Gets the minor version of the MapForce Server API. This can be different from the product version if the API is connected to another server.

```
public int getAPIMinorVersion()
```

getAPIServicePackVersion method

Gets the service pack version of the MapForce Server API. This can be different from the product version if the API is connected to another server.

```
public int getAPIServicePackVersion()
```

getLastExecutionMessage method

Gets the message received during the last Run command.

```
public java.lang.String getLastExecutionMessage()
```

getMajorVersion method

Gets the major version of the product (for example, "16" for Altova MapForce Server 2014 r2 sp1 (x64)).

```
public int getMajorVersion()
                    throws MapForceServerException
```

getMinorVersion method

Gets the minor version of the product (for example, "2" for Altova MapForce Server 2014 r2 sp1 (x64)).

```
public int getMinorVersion()
                    throws MapForceServerException
```

getOutputParameter method

Gets the string output generated by the last `run` command. Returns null if no output was generated. This function requires a string parameter which identifies the name of the output component in MapForce.

```
public String getOutputParameter( String name )
```

getProductName method

Gets the name of the product (for example, "Altova MapForce Server" for Altova MapForce Server 2014 r2 sp1 (x64)).

```
public java.lang.String getProductName()
```

```
                                    throws MapForceServerException
```

getProductNameAndVersion method

Gets the complete name and version of the product (for example, "Altova MapForce Server 2014 r2 sp1 (x64)").

```
public java.lang.String getProductNameAndVersion()
                                                throws
MapForceServerException
```

getServicePackVersion method

Gets the service pack version of the product (for example, "1" for Altova MapForce Server 2014 r2 sp1 (x64)).

```
public int getServicePackVersion()
                           throws MapForceServerException
```

is64Bit method

Returns `true` if the MapForce Server engine is a 64-bit executable.

```
public boolean is64bit()
                throws MapForceServerException
```

run method

Executes a MapForce Server Execution file (.mfx file). Returns `true` in case of success; `false` otherwise.

```
public boolean run(java.lang.String mappingPath)
            throws MapForceServerException
```

setServerPath method

Sets the path of the MapForce Server executable.

```
public void setServerPath(java.lang.String serverFile)
```

setWorkingDirectory method

Sets the current directory for running jobs (relative paths will be evaluated against the

working directory).

```
public void setWorkingDirectory(java.lang.String workingDirectory)
```

MapForceServerException Class

Purpose

The `MapForceServerException` class provides programmatic access to exceptions thrown by the `MapForceServer` class.

```
public class MapForceServerException extends Exception
```

Chapter 7

Digital Certificate Management

7 Digital Certificate Management

Digital certificate management is an integral part of secure data exchange between a client computer and a Web server. Since mappings can be executed not only on Windows by MapForce, but also on a Windows, Linux or Mac server by MapForce Server (either standalone or in FlowForce Server execution), this section deals with managing HTTPS certificates on various platforms.

In the context of secure HyperText Transport Protocol (HTTPS), it is important to distinguish between server and client certificates.

Server certificates

A server certificate is what identifies a server as a trusted entity to a client application such as MapForce. The server certificate may be digitally signed by a commercial Certificate Authority, or it may be self-signed by your organization. In either case, while designing the mapping in MapForce, you can specify the following settings:

- Whether the server certificate must be checked.
- Whether the request must proceed if a mismatch has been detected between the name certificate and the name of the host.

These settings are available on the HTTP Security Settings dialog box of MapForce. When you enable server certificate checks, consider the following:

- If you are calling a Web server whose certificate is signed by a trusted Certificate Authority, your operating system will likely be already configured to trust the server certificate, and no additional configuration is necessary.
- If you are calling a Web server which provides a self-signed certificate (for example, a local network server within your organization), you will need to configure your operating system as well to trust that certificate.

In most cases, you can check the level of trust between your operating system and the Web server by typing the URL of the Web service in the browser's address bar. If the server is not trusted, or if your operating system is not configured to trust the server, your browser will display a message such as "This connection is untrusted", or "There is a problem with this website's certificate". Note that you cannot use the browser to check the level of trust with a Web server if the browser uses a certificate database other than that of the operating system (for example, Firefox 35.0.1 on Ubuntu 14.04).

On Windows, you can establish trust with the server by following the browser's instructions and importing or installing the required certificates into your system's Trusted Root Authorities store (see Trusting Server Certificates on Windows). On Mac, you can do the equivalent operation in Safari (see Trusting Server Certificates on Mac). For instructions applicable to Linux, see Trusting Server Certificates on Linux.

Client certificates

While server certificates are used to identify a server as a trusted entity, client certificates are primarily used to authenticate the caller against the Web server. If you intend to call a Web server which requires client certificates, you may need to contact the administrator of the Web server for the client configuration instructions. Taking IIS (Internet Information Services) as an example, the

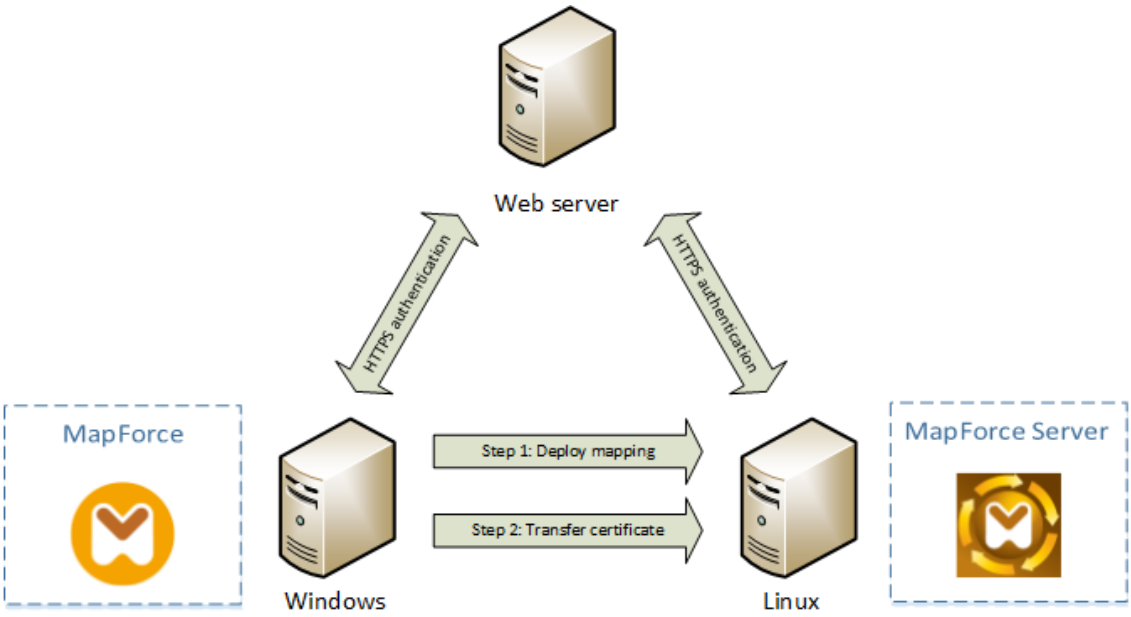
Web server may be configured to handle HTTPS and client certificates in one of the following ways:

- Require HTTPS and ignore client certificate
- Require HTTPS and accept client certificate
- Require HTTPS and require client certificate

The success or failure of the Web service request depends both on the configuration of the Web server and the client application. For example, if the Web server is configured to require a client certificate, then, for the call to be successful, the calling application must present a valid client certificate.

From a MapForce perspective, the same is true for mappings which include Web service calls through HTTPS. In particular, to run such mappings successfully, it is assumed that the Web server has been configured to accept or require the client certificate, and that the operating system where the mapping runs provides the correct client certificate to the Web server.

The diagram below illustrates a scenario where a client certificate used in MapForce is transferred to a Linux server running MapForce Server. Once the certificate has been transferred to the target operating system, MapForce Server can use it to authenticate itself against the Web server and execute the mapping successfully.



Deploying mappings with client certificates to another computer

For HTTPS authentication in Web service calls, MapForce is capable of using Transport Layer Security (TLS) on top of HTTP, which is the successor of Secure Sockets Layer (SSL) protocol. Note that fallback to SSL may occur if either the client implementation or the server does not support TLS.

To support Web calls with client certificate authentication on multiple platforms, MapForce (and MapForce Server) relies on the certificate management implementation of each platform, thus ensuring that certificate management is always in the scope of the underlying operating system. Each operating system provides different support for certificate management, as shown in the

table below.

Platform	Certificate management and implementation
Windows	On Windows, you can manage certificates using the Certificate snap-in (see Accessing the Certificate Stores on Windows). TLS support is available through the *Secure Channel* (also known as *SChannel*) library.
Linux	On Linux, you can manage certificates using the OpenSSL (`openssl`) command line tool and library. If OpenSSL support is not already available on the Linux machine where MapForce Server is installed, you will need to download and install it before you can manage certificates. TLS support is available through the OpenSSL library (https://www.openssl.org/).
Mac	On Mac, you can manage certificates using the *Keychain Access Manager*, located under **Finder > Applications > Utilities**. TLS support is provided by the *Secure Transport* library native to the operating system.

If you execute the mapping on a Windows operating system where you can already successfully consume the same Web service that you intend to call from MapForce, no additional certificate configuration is normally required (for the conditions to run the mapping successfully on Windows, see Client Certificates on Windows). However, if you design mappings with MapForce on a Windows computer, and then deploy them to another computer (which may run a different operating system), the client certificate is not stored or copied together with the deployed package. For the Web service call (and the mapping) to execute successfully, the client certificate must exist on the target operating system as well.

To transfer a certificate from a Windows system to another Windows-based computer, export the required certificate (with private key) from the source system (see Exporting Certificates from Windows). Then import the same certificate to the **Current User\Personal** store on the target operation system (see Client Certificates on Windows).

For instructions on how to transfer client certificates to the Linux and Mac platforms, see Client Certificates on Linux and Client Certificates on Mac, respectively.

7.1 Trusting Server Certificates on Linux

To establish trust with a Web server on Linux, obtain the certificate file of the Web server, copy it to the system's certificate store, and then update the latter (see instructions below). One of the ways to obtain the server certificate is by using the Firefox browser, as shown in the example below.

> Perform the following steps only if you are sure of the authenticity of the Web server certificate.

Debian, Ubuntu

1. Copy the certificate file of the Web server to the following directory.

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

2. Update the certificate store as follows:

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

Cent OS

1. Install the `ca-certificates` package:

```
yum install ca-certificates
```

2. Enable the dynamic certificate authority configuration feature:

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

3. Copy the server certificate to the following directory:

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

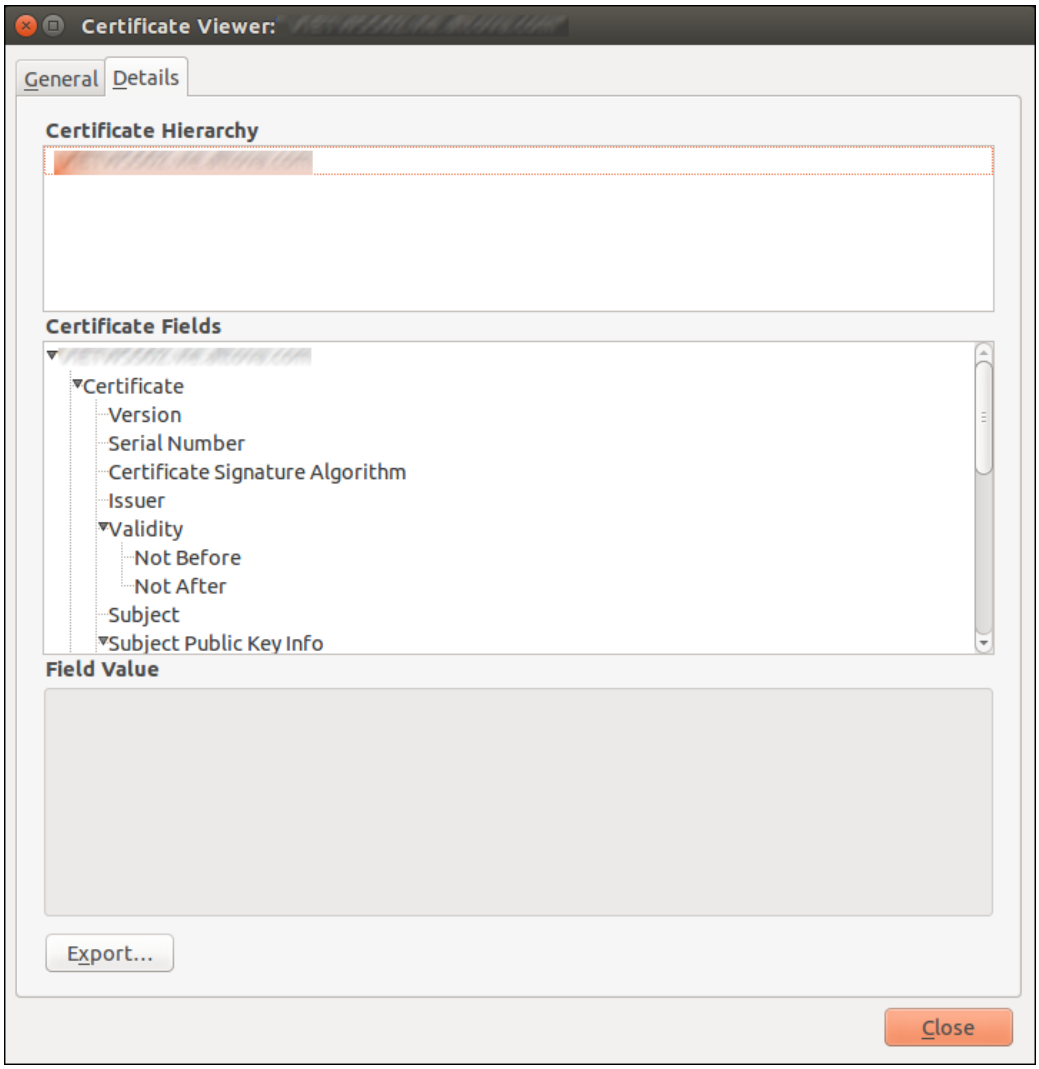
4. Use the command:

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

Example: Exporting the certificate of the Web server with Firefox on Ubuntu

1. Run Firefox and access the URL of the Web server.
2. When prompted with the message "This connection is untrusted", click **Add Security Exception**.

3. Click **View**.
4. Click the **Details** tab.



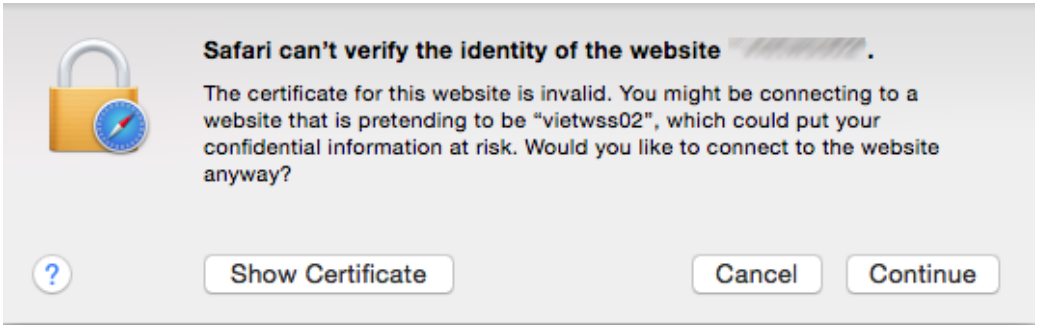
5. Click **Export** and save the certificate file to a local directory.

7.2 Trusting Server Certificates on Mac

To establish trust with a Web server using Safari:

> Perform the following steps only if you are sure of the authenticity of the Web server certificate.

1. In the browser address bar, enter the HTTPS address of the Web server.
2. When prompted to connect to the website, click **Show Certificate**.



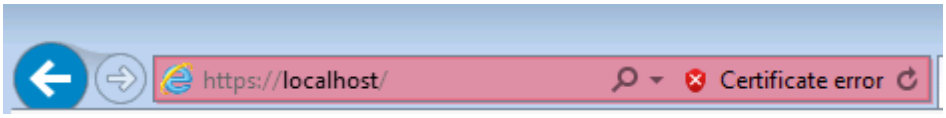
3. Select the option **Always trust {certificate} when connecting to {website}**.
4. Click **Continue** and enter your password when prompted.

7.3 Trusting Server Certificates on Windows

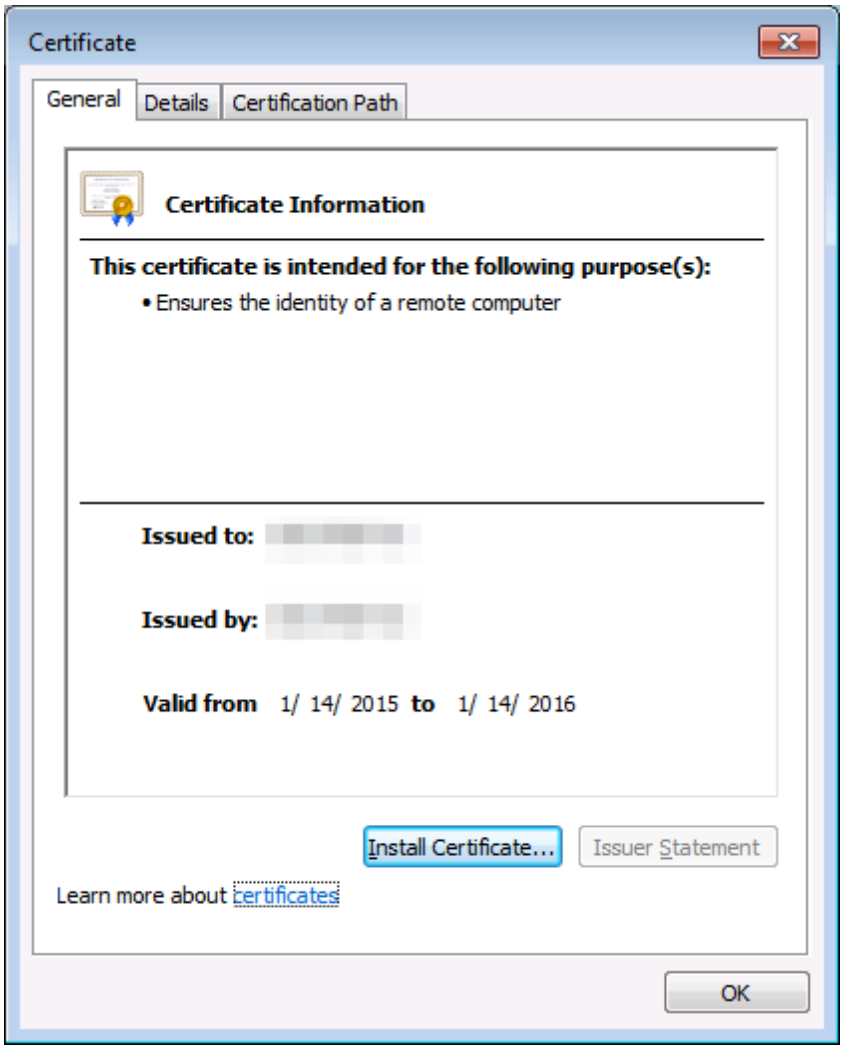
To establish trust with a Web server using Internet Explorer 11:

> Perform the following steps only if you are sure of the authenticity of the Web server certificate.

1. In the browser address bar, enter the HTTPS address of the Web server.
2. Click **Continue to this website (not recommended)**.
3. Click the Certificate error area, and then click **View certificates**.

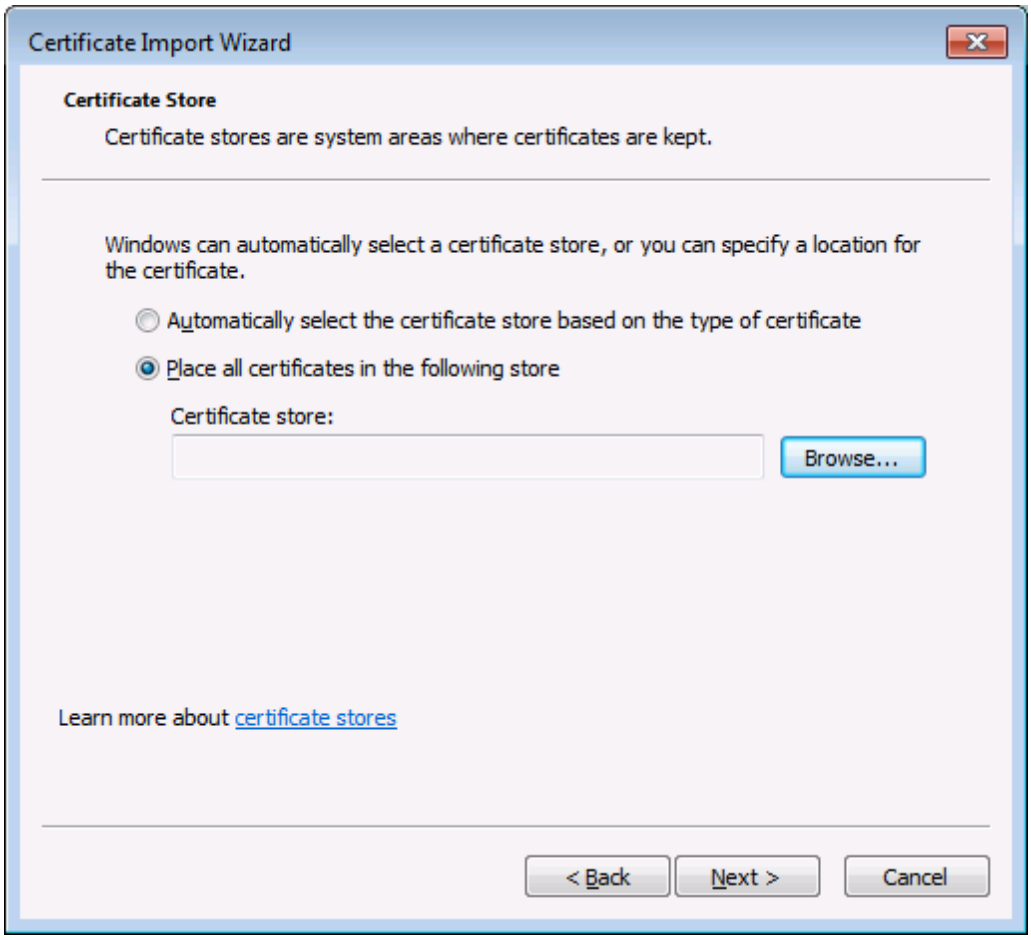


4. Click **Install Certificate**.

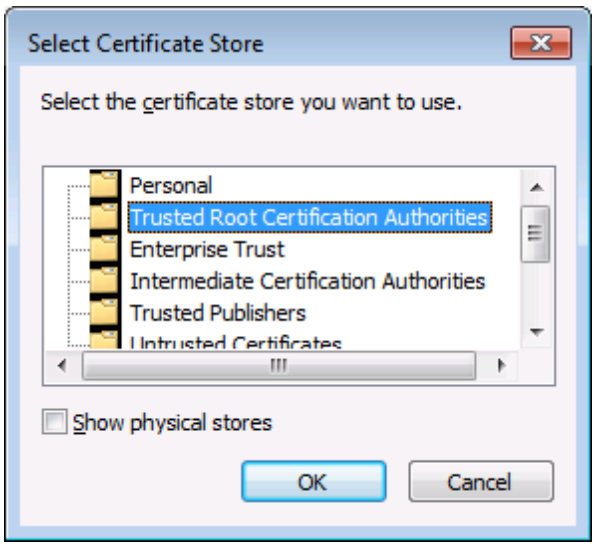


5. Click **Next**.

6. Choose to select a store manually.



7. Browse for the Trusted Root Certification Authorities, and then click OK.



8. When prompted to confirm your action, click OK.

7.4 Accessing the Certificate Stores on Windows

On Windows, you can manage certificates either from the *Certificates* Microsoft Management Console (MMC) snap-in, or from Internet Explorer.

To open the Certificates snap-in (for the current Windows user):

- Run `certmgr.msc` at the command line.

To open the certificate management dialog box in Internet Explorer:

1. On the **Tools** menu, click **Internet Options**.
2. Click the **Content** tab, and then click **Certificates**.

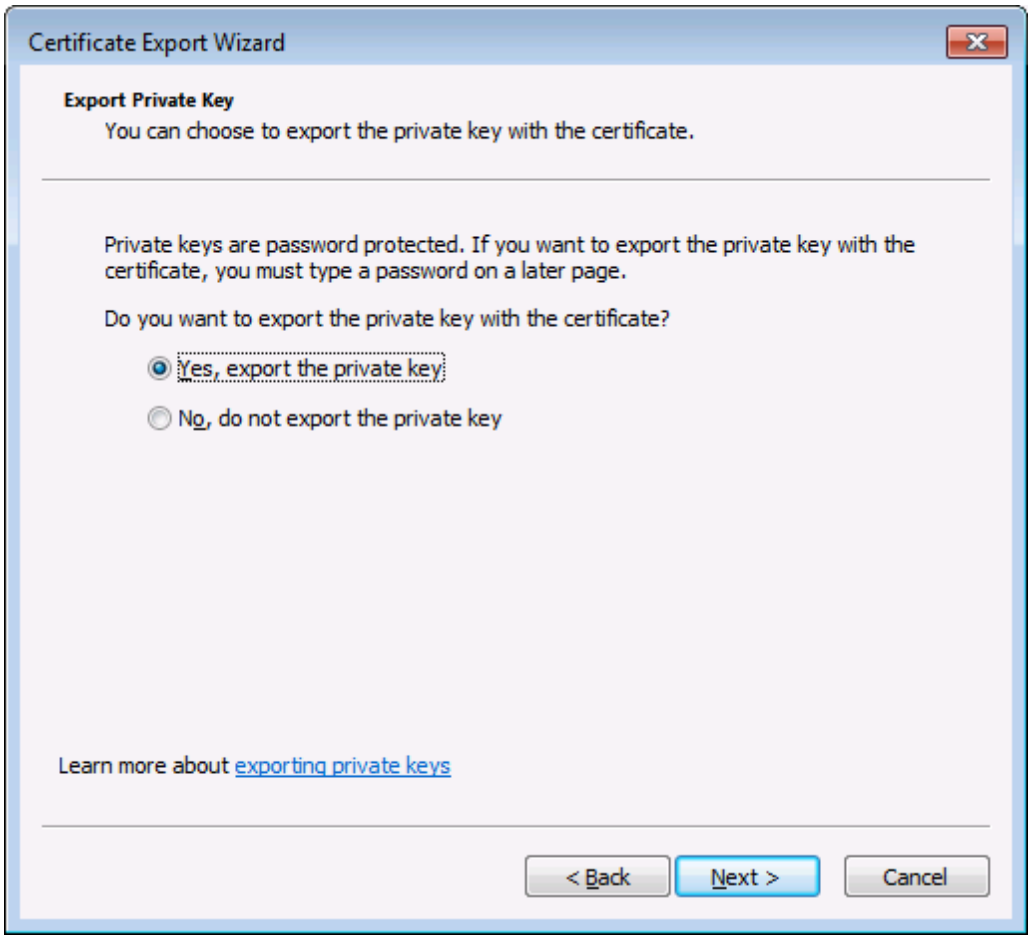
7.5 Exporting Certificates from Windows

For mappings that call Web services through HTTPS and are deployed to a Mac or Linux server running MapForce Server or FlowForce Server, the same client certificate must be available on the non-Windows operating system as the one used on Windows to design and test the mapping. To execute such mappings on a non-Windows operating system with MapForce Server, export the required certificate with private key from Windows and then import it into the target operating system.

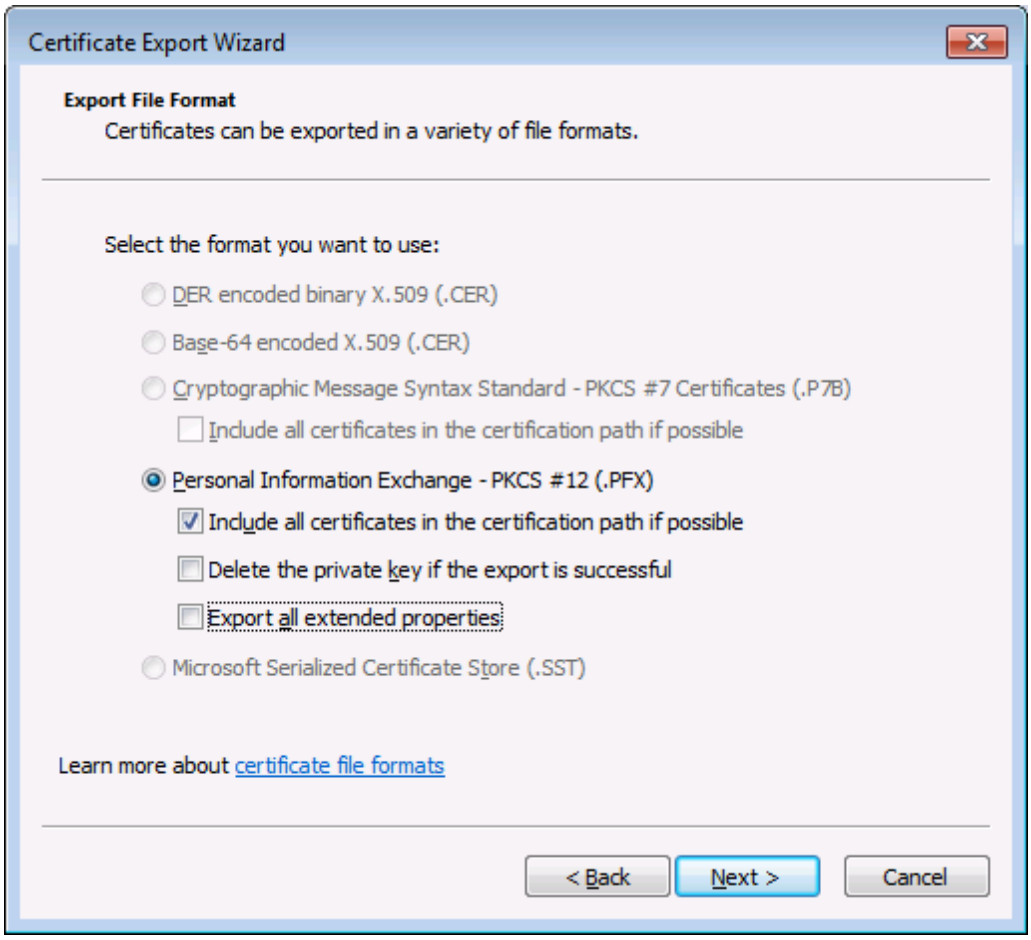
To export a certificate with private key from Windows:

1. On Windows, open the Certificates snap-in (see Accessing the Certificate Stores on Windows).
2. Right-click the certificate that you want to export, point to **All Tasks**, and then click **Export**.
3. Click **Next**.

4. Choose to export from Windows the certificate together with its private key, and then click **Next**.

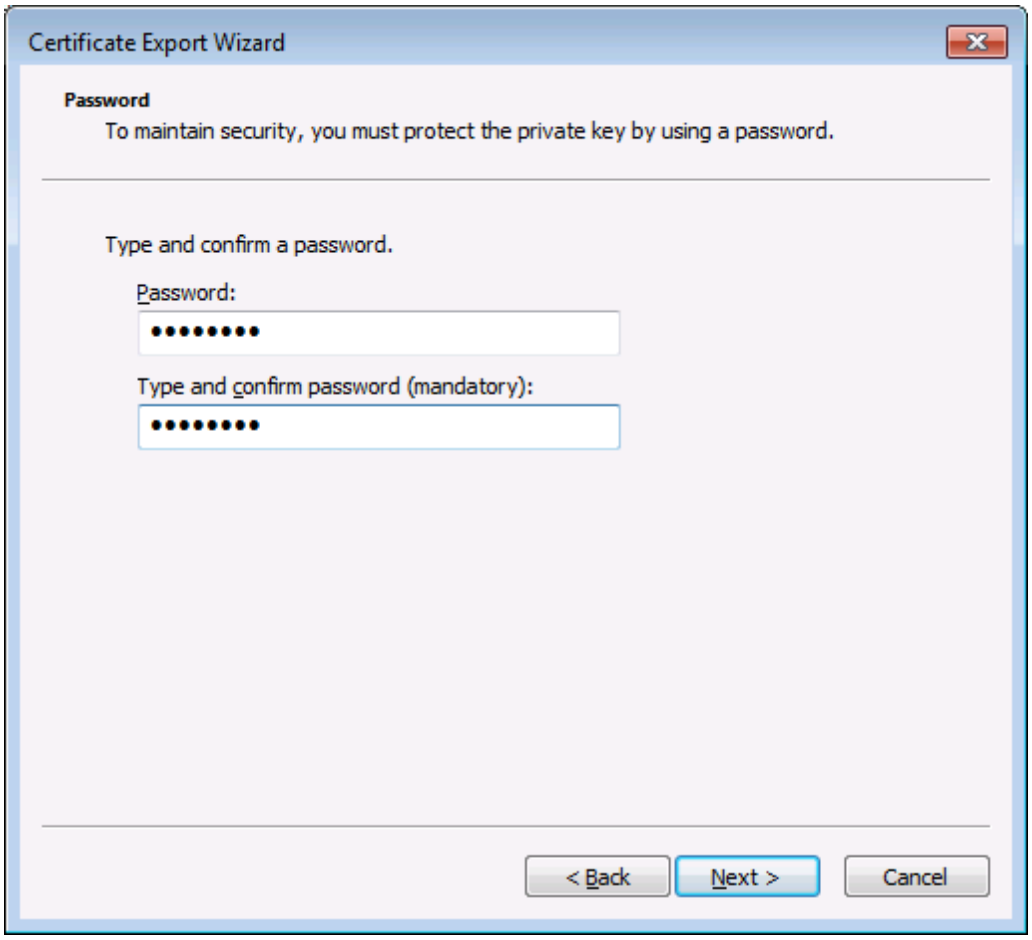


5. Choose the *Personal Information Exchange - PKCS #12 (.pfx)* file format, and then click **Next**.

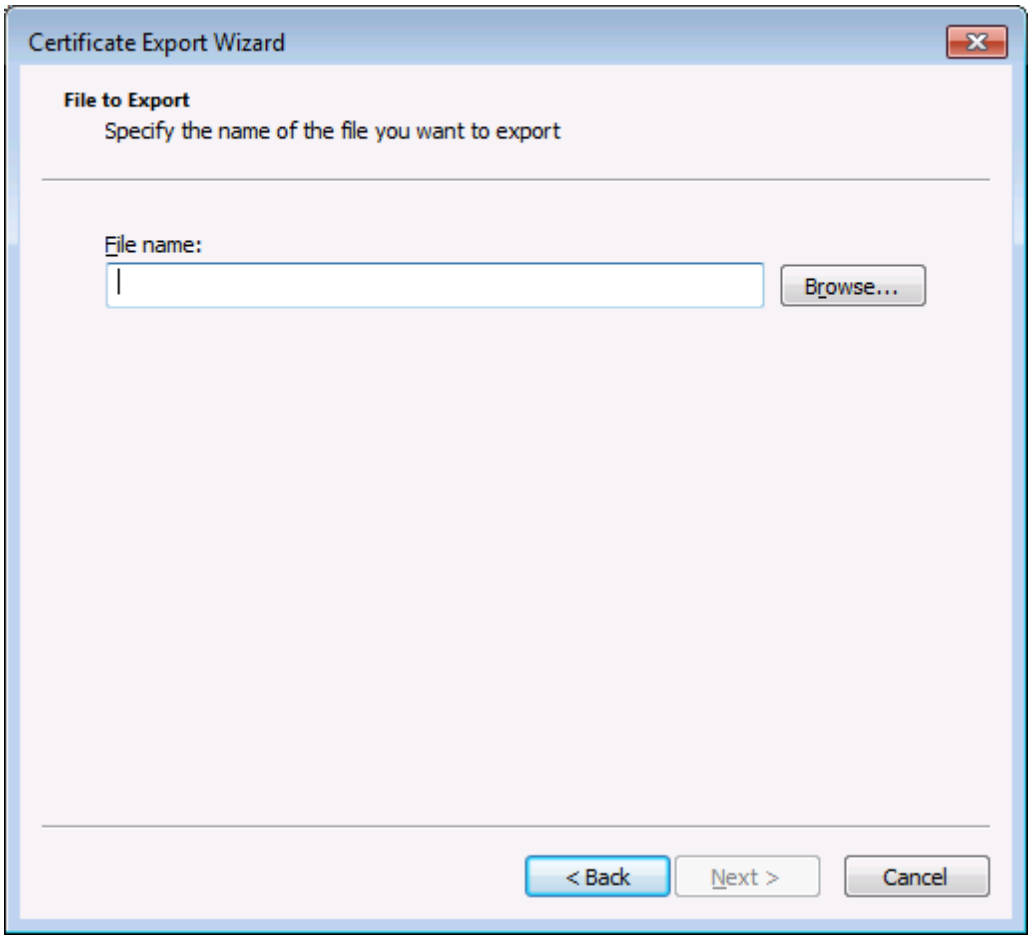


Note: Make sure not to select the option **Delete the private key if the export is successful**, otherwise you will not be able to make use of the certificate after it is exported.

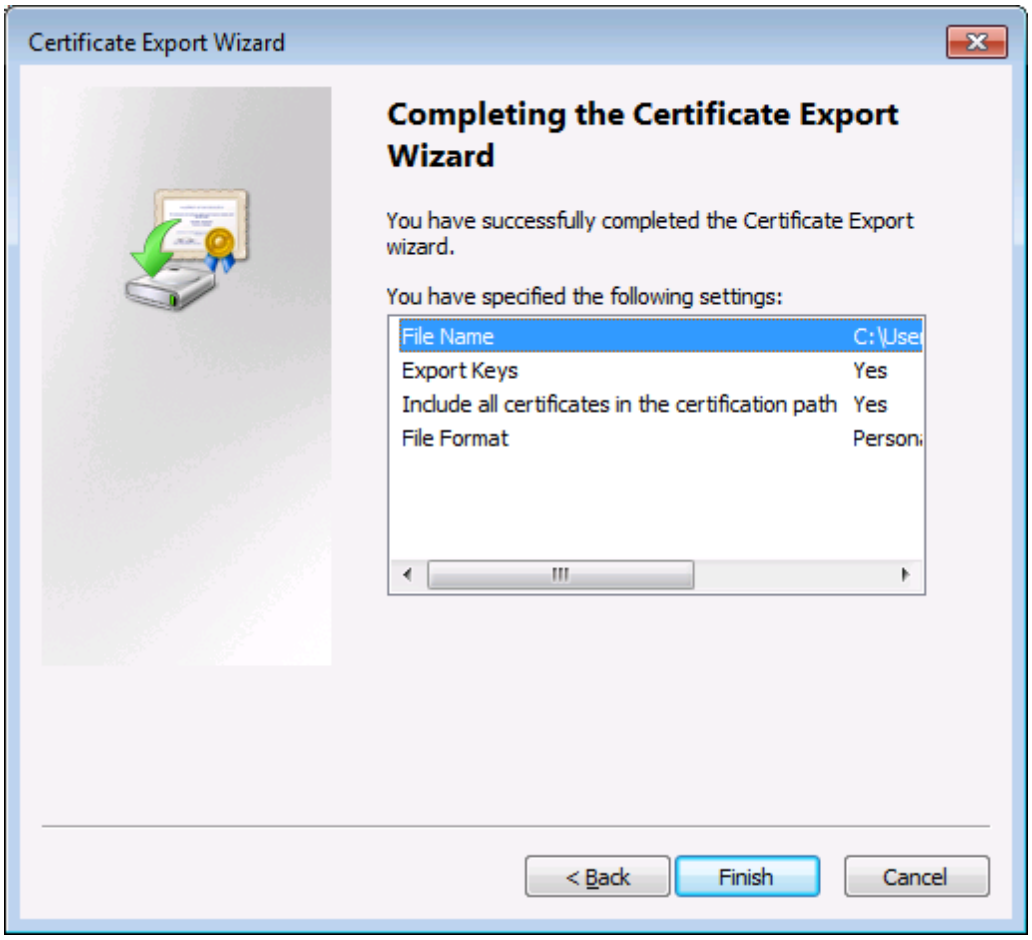
6. Enter a password, and then click **Next**. You will need this password after you copy the certificate to the target operating system.



7. Browse for the location of the file to export, and then click **Next**.



8. Click **Finish**.



7.6 Client Certificates on Linux

If your mappings include Web service authentication through HTTPS by means of client certificates, follow these steps to deploy such mappings to a Linux machine running MapForce Server:

1. In the HTTP Security Settings dialog box, click **Client Certificate**, and then select the required certificate from the **Current User\Personal** store on Windows .
2. Save and deploy the mapping to the target operating system.
3. Transfer the client certificate required by the Web service call to the target operating system. Make sure that the certificate has a private key, and that the **Enhanced Key Usage** property of the certificate includes "Client authentication" as purpose.

To transfer the client certificate to Linux:

1. Export the client certificate with private key from Windows, in the *Personal Information Exchange - PKCS #12 (.pfx)* file format (see Exporting Certificates from Windows).
2. Copy the certificate file to the Linux machine.
3. Convert the .pfx file to .pem format using the command:

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

This command parses the .pfx file and outputs a .pem file, without encrypting the private key. Certificates with an encrypted private key prompt for password and are not supported in server execution.

Executing the mapping

To instruct MapForce Server to use the .pem file as client certificate, set the `--certificatespath` parameter when running the mapping. The `--certificatespath` parameter defines the path of the directory where all certificates required by the current mapping are stored. For example, if the certificate file path is `/home/John/John Doe.pem`, then `--certificatespath` must be set to `/home/John`.

By default, if the `--certificatespath` parameter is not provided, MapForce Server looks for certificates in the directory `$HOME/.config/altova/certificates` of the current user.

> For the mapping to execute successfully, the certificate file is expected to have the .pem extension and the file name must match the Common Name (CN) of the certificate, including spaces (for example, **John Doe.pem**). If the CN contains a forward slash (/), it must be replaced with an underscore (_) character.

If you intend to execute the mapping as a FlowForce Server job, copy the certificate file to the `$HOME/.config/altova/certificates` directory. When running the job, FlowForce Server will use this directory to look for any certificate files required by the mapping.

> For security considerations, make sure that certificate files are not readable by other users, since they contain sensitive information.

7.7 Client Certificates on Mac

If your mappings include Web service authentication through HTTPS client certificates, follow these steps to deploy such mappings to a OS X running MapForce Server:

1. In the HTTP Security Settings dialog box of MapForce, click **Client Certificate**, and then select the required certificate.
2. If the certificate name does not match exactly the host name of the server, select **Allow name mismatch between certificate and request**.
3. Save and deploy the mapping to the target operating system .
4. Transfer the client certificate required by the Web service call to the target operating system. Make sure that the certificate has a private key, and that the **Enhanced Key Usage** property of the certificate includes "Client authentication" as purpose.

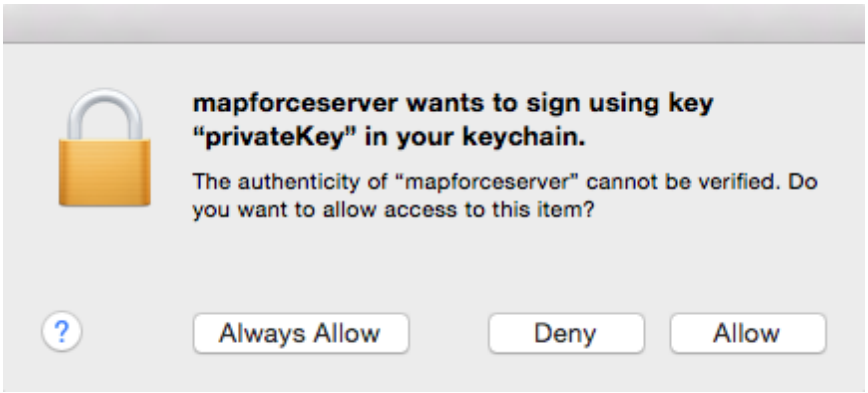
To transfer the client certificate to Mac:

1. Export the client certificate with private key from Windows, in the *Personal Information Exchange - PKCS #12 (.pfx)* file format (see Exporting Certificates from Windows) and copy the .pfx file to the Mac.
2. If this hasn't been done already, make sure that the operating system trusts the server certificate (see Trusting Server Certificates on Mac OS).
3. Run Keychain Access from **Finder > Applications > Utilities**.
4. On the **File** menu, click **Import Items**.
5. Browse for the the client certificate exported from Windows in step 1 and select a destination keychain.
6. Click **Open** and enter the password with which the certificate was encrypted.

Executing the mapping

You are now ready to run the mapping using the MapForce Server `run` command. Note the following:

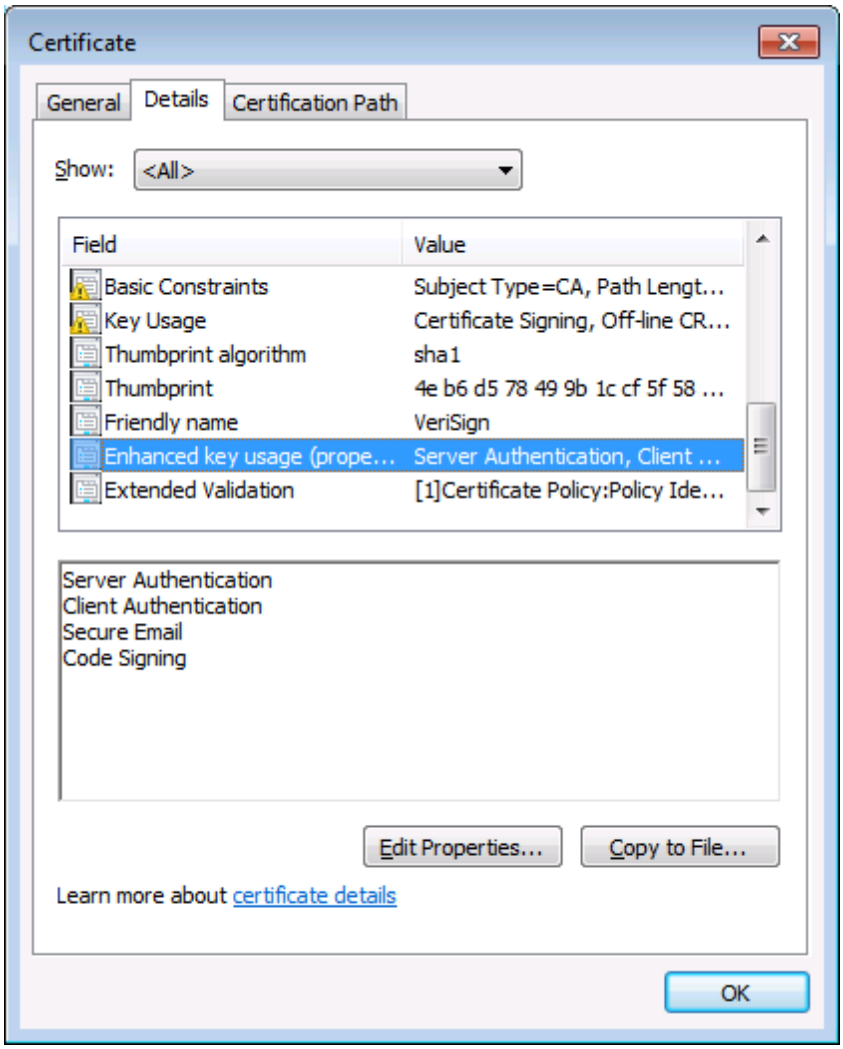
- If you execute the mapping remotely through SSH, first unlock the keychain with the `security unlock-keychain` command.
- If you execute the mapping through the Mac OS graphical user interface, when prompted to allow MapForce Server access to the keychain, click **Allow**.



7.8 Client Certificates on Windows

When you run on Windows a mapping which requires client certificates, the conditions to run the mapping successfully are as follows:

- The client certificate must exist in the **Current User\Personal** certificate store (also referred to as the **My** store). For the certificate to exist in this store, it must be imported through the Certificate Import Wizard. For instructions, see http://windows.microsoft.com/en-us/windows/import-export-certificates-private-keys#1TC=windows-7.
- The certificate must have a private key.
- The **Enhanced Key Usage** property of the certificate must include "Client authentication" as purpose.



In the current version of MapForce, due to a limitation of the library used by MapForce, Windows will select the required certificate automatically from the certificate store when you run the mapping. The mapping will execute successfully if, after filtering the **Current User\Personal** certificate store, the server finds a suitable certificate. Note that the HTTPS authentication (and the certificate selection operation) is managed by Windows and is not controlled by MapForce or MapForce Server. In some cases, if multiple certificates exist in the **Current User\Personal**

store, an unsuitable certificate may be selected automatically by the operating system, which causes the mapping execution to fail. This situation can be avoided by limiting the number of certificates available in the **Current User\Personal** store.

Index

W