

Altova MapForce Server 2018 Advanced Edition

User and Reference Manual

Altova MapForce Server 2018 Advanced Edition User & Reference Manual

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

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

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

Published: 2018

© 2018 Altova GmbH

Table of Contents

1	Altova MapForce Server 2018 Advanced Edition	3
1.1	How It Works	5
1.2	Preparing Mappings for Server Execution	8
1.3	Global Resources	12
1.4	Join Optimization	13
2	Setup on Windows	18
2.1	Installation on Windows	19
2.2	Licensing on Windows	21
3	Setup on Linux	26
3.1	Installation on Linux	27
3.2	Licensing on Linux	30
4	Setup on macOS	34
4.1	Installation on macOS	35
4.2	Licensing on macOS	38
5	MapForce Server Command Line	42
5.1	assignlicense (Windows only)	44
5.2	exportresourcestrings	46
5.3	help	48
5.4	licenseserver	50
5.5	run	52
5.6	setdeflang	56
5.7	verifylicense (Windows only)	58
6	MapForce Server API	62

6.1	About the .NET Interface	63
6.2	About the COM Interface	65
6.3	About the Java Interface	66
6.4	Code Examples	69
6.4.1	C#	70
6.4.2	C++	71
6.4.3	Java	73
6.4.4	VBScript	75
6.4.5	Visual Basic	76
6.4.6	Visual Basic for Applications (VBA)	77
6.5	API Reference (COM, .NET)	80
6.5.1	IServer Interface	81
6.6	API Reference (Java)	88
6.6.1	MapForceServer Class	89
6.6.2	MapForceServerException Class	93

7 Altova LicenseServer 96

7.1	Network Information	98
7.2	Updating LicenseServer	100
7.3	Installation (Windows)	101
7.4	Installation (Linux)	103
7.5	Installation (macOS)	106
7.6	Altova ServiceController	108
7.7	How to Assign Licenses	109
7.7.1	Start LicenseServer	111
7.7.2	Open LicenseServer's Config Page (Windows)	113
7.7.3	Open LicenseServer's Config Page (Linux)	116
7.7.4	Open LicenseServer's Config Page (Mac OS X)	118
7.7.5	Upload Licenses to LicenseServer	121
7.7.6	Register Product/s	124
	<i>Register Altova Desktop Products</i>	124
	<i>Register FlowForce Server</i>	125
	<i>Register MapForce Server</i>	130
	<i>Register MobileTogether Server</i>	132
	<i>Register RaptorXML(+XBRL) Server</i>	133
	<i>Register StyleVision Server</i>	134
7.7.7	Assign Licenses to Registered Products	136
7.8	Configuration Page Reference	142
7.8.1	License Pool	143

7.8.2	Client Management	150
7.8.3	Client Monitoring	155
7.8.4	Settings	156
7.8.5	Messages, Log Out	163
7.9	Password Reset	164

8 Digital Certificate Management 166

8.1	Trusting Server Certificates on Linux	169
8.2	Trusting Server Certificates on OS X / macOS	172
8.3	Trusting Server Certificates on Windows	173
8.4	Accessing the Certificate Stores on Windows	177
8.5	Exporting Certificates from Windows	178
8.6	Client Certificates on Linux	184
8.7	Client Certificates on OS X / macOS	186
8.8	Client Certificates on Windows	188

Index

Chapter 1

Altova MapForce Server 2018 Advanced Edition

1 Altova MapForce Server 2018 Advanced Edition

MapForce Server is an enterprise software solution that runs data mapping transformations on Windows, Linux, and OS X/ macOS operating systems. The data mappings themselves (or Mapping Design Files, *.mfd) are visually designed with Altova MapForce (<https://www.altova.com/mapforce>), 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 (<https://www.altova.com/flowforceserver>). 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 Advanced Edition 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
- Support for Altova Global Resources—a way of making file, folder, or database references configurable and portable across multiple environments and across multiple Altova applications, see [Altova Global Resources](#)
- Accelerates execution of mappings where join optimization is possible (see [About Join Optimization](#))
- Runs mappings that apply functions and defaults to multiple items simultaneously. Such mappings make it possible, for example, to easily replace all encountered null values with empty strings or custom text
- Runs mappings that perform bulk database inserts

Limitations

- XML digital signatures are not supported
- ADO, ADO.NET, and ODBC database connections are supported only on Windows. On Linux and OS X/ macOS, native database connectivity is available for SQLite and PostgreSQL databases. For other databases running on Linux or OS X/ macOS, JDBC should be used.

System Requirements

▼ Windows

Windows 7 SP1 with Platform Update, Windows 8, Windows 10

▼ Windows Server

Windows Server 2008 R2 SP1 with Platform Update 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

The following libraries are required as a prerequisite to install and run the application. If the packages below are not already available on your Linux machine, run the command `yum` (or `apt-get` if applicable) to install them.

Required by	CentOS, RedHat	Debian	Ubuntu
LicenseServer	krb5-libs	libgssapi-krb5-2	libgssapi-krb5-2
MapForce Server Advanced Edition	libidn, krb5-libs	libidn11, libgssapi-krb5-2	libidn11, libgssapi-krb5-2

▼ (Mac) OS X, macOS

OS X 10.10 or newer

Note: On Windows and macOS, it is possible to install multiple versions of MapForce Server on the same machine. In such cases, be aware that only one version can be registered to the same Altova LicenseServer at a time. Hence, two different versions of MapForce Server cannot run concurrently on the same machine. Multiple instances of the same product version can run in parallel, however. For more information, see [Assign Licenses to Registered Products](#).

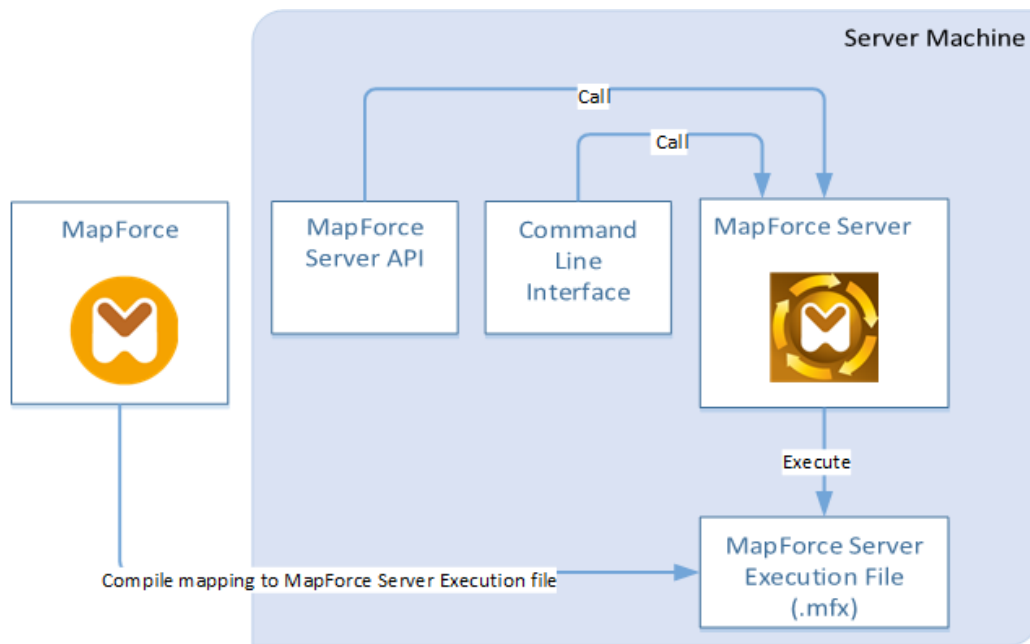
Last updated: 21 June 2018

1.1 How It Works

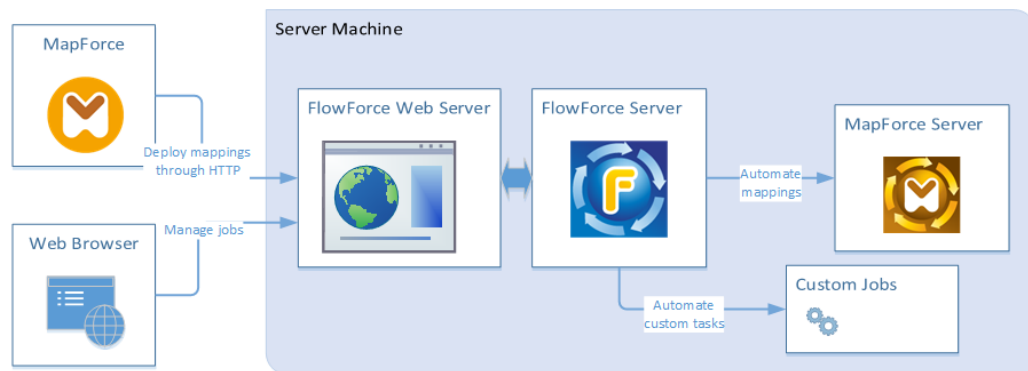
The role of MapForce Server is to execute data mappings created with Altova MapForce (<https://www.altova.com/mapforce.html>).

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 (<https://www.altova.com/documentation>).

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 examples, refer to the FlowForce Server documentation <https://www.altova.com/documentation>). 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, see [Preparing Mappings for Server Execution](#).

1.2 Preparing Mappings for Server Execution

A mapping designed and previewed with MapForce may refer to resources which are outside of the current machine and operating system (such as databases). In addition to this, in MapForce, all mapping paths follow Windows-style conventions by default. Thirdly, the machine where MapForce Server runs might not support the same database connections as the machine where the mapping was designed. For this reason, running mappings in a server environment typically requires some preparation, especially if the target machine is not the same as the source machine.

Note: The term "source machine" refers to the computer where the MapForce is installed and the term "target machine" refers to the computer where MapForce Server or FlowForce Server is installed. In the most simple scenario, this is the same computer. In a more advanced scenario, MapForce runs on a Windows machine whereas MapForce Server or FlowForce Server runs on a Linux or OS X/ macOS machine.

As best practice, always make sure that the mapping validates successfully in MapForce before deploying it to FlowForce Server or compiling it to a MapForce Server execution file.

If MapForce Server runs standalone (without FlowForce Server), the required licenses are as follows:

- On the source machine, MapForce Enterprise or Professional edition is required to design the mapping and compile it to a server execution file (.mfx).
- On the target machine, MapForce Server or MapForce Server Advanced Edition is required to run the mapping.

If MapForce Server runs under FlowForce Server management, the following requirements apply:

- On the source machine, MapForce Enterprise or Professional edition is required to design the mapping and deploy it to a target machine.
- Both MapForce Server and FlowForce Server must be licensed on the target machine. The role of MapForce Server is to run the mapping; the role of FlowForce is to make the mapping available as a job which benefits from features such as scheduled or on demand execution, execution as a Web service, error handling, conditional processing, email notifications, and others.
- FlowForce Server must be up and running at the configured network address and port. Namely, the "FlowForce Web Server" service must be started and configured to accept connections from HTTP clients (or HTTPS if configured) and must not be blocked by the firewall. The "FlowForce Server" service must also be started and running at the designated address and port.
- You have a FlowForce Server user account with permissions to one of the containers (by default, the **/public** container is accessible to any authenticated user).

General considerations

- If you intend to run the mapping on a target machine with standalone MapForce Server, all input and output files and schemas referenced by the mapping must be copied to the target machine as well. If MapForce Server runs under FlowForce Server management, there is no need to copy files manually. In this case, the instance and schema files are included in the package deployed to the target machine.

- If the mapping includes database components which require specific database drivers, such drivers must be installed on the target machine as well. For example, if your mapping reads data from a Microsoft Access database, then Microsoft Access or Microsoft Access Runtime (<https://www.microsoft.com/en-us/download/details.aspx?id=50040>) must be installed on the target machine as well.
- When you deploy a mapping to non-Windows platforms, ADO, ADO.NET and ODBC database connections are automatically changed to JDBC, see "Linux and OS X / macOS considerations" below.
- If the mapping contains custom function calls (for example, to .dll or .class files), such dependencies are not deployed together with the mapping, since they are not known before runtime. In this case, you can copy them manually to the target machine.
- Some mappings read multiple input files using a wildcard path. In this case, the input file names are not known before runtime and so they are not deployed. For the mapping to execute successfully, the input files must exist on the target machine.
- If the mapping output path includes directories, those directories must exist on the target machine. Otherwise, an error will be generated when you execute the mapping. This behavior is unlike MapForce, where non-existing directories are generated automatically if the option **Generate output to temporary files** is enabled.
- If the mapping calls a Web service that requires HTTPS authentication with a client certificate, the certificate must be transferred to the target machine as well, see [Digital Certificate Management](#).
- If the mapping connects to file-based databases such as Microsoft Access and SQLite, the database file must be manually transferred to the target machine or saved to a shared directory which is accessible to both the source and the target machine and referenced from there, see "File-based databases" below.

Linux and OS X / macOS considerations

If you intend to run the mapping on a Linux or OS X / macOS server, ensure that the mapping follows the applicable path conventions and uses a supported database connection.

To make paths portable to non-Windows operating systems, use relative instead of absolute paths when designing the mapping in MapForce. For example, you can copy all input or output files required by the mapping into the same directory as the mapping, and then reference them just by file name. Importantly, both MapForce Server and FlowForce Server support a so-called "working directory" against which all relative paths will be resolved. The working directory is specified at mapping runtime, as follows:

- In FlowForce Server, by editing the "Working-directory" parameter of any job.
- In MapForce Server API, through the `WorkingDirectory` property of the COM and .NET API, or through the `setWorkingDirectory` method of the Java API.
- In MapForce Server command line, the working directory is the current directory of the command shell.

As for database connections, be aware that ADO, ADO.NET, and ODBC connections are not supported on Linux and OS X / macOS machines. Therefore, if the target machine is Linux or OS X / macOS, such connections are converted to JDBC when you deploy the mapping to FlowForce or when you compile the mapping to a MapForce Server execution file. In this case, you have the following options before deploying the mapping or compiling it to a server execution file:

- In MapForce, create a JDBC connection to the database (see [Setting up a JDBC Connection](#))

- In Mapforce, fill the JDBC database connection details in the "JDBC-specific Settings" section of the database component .

If the mapping uses a native connection to a PostgreSQL or SQLite database, the native connection is preserved and no JDBC conversion takes place. If the mapping connects to a file-based database, such as Microsoft Access and SQLite, additional configuration is required, see "File-based databases" below.

File-based databases

File-based databases such as Microsoft Access and SQLite are not included in the package deployed to FlowForce Server or in the compiled MapForce Server execution file. Therefore, if the source and target machine are not the same, take the following steps:

1. In MapForce, right-click the mapping and clear the check box **Make paths absolute in generated code**.
2. Right-click the database component on the mapping and add a connection to the database file using a relative path. A simple way to avoid path-related issues is to save the mapping design (.mfd file) in the same directory as the database file and to refer to the latter from the mapping just by file name (thus using a relative path).
3. Copy the database file to a directory on the target machine (let's call it "working directory"). Keep this directory in mind since it will be required to run the mapping on the server, as shown below.

To run such mappings on the server, do one of the following:

- If the mapping will be run by MapForce Server under FlowForce Server control, configure the FlowForce Server job to point to the working directory created previously. The database file must reside in the working directory. For an example, see "Exposing a Job as a Web Service" (https://manual.altova.com/FlowForceServer/FlowForceServerAdvanced/index.html?fs_example_web_service.htm).
- If the mapping will be run by standalone MapForce Server at the command line, change the current directory to the working directory (for example, `cd path\to\working\directory`) before calling the `run` command of MapForce Server.
- If the mapping will be run by the MapForce Server API, set the working directory programmatically before running the mapping. To facilitate this, the property `WorkingDirectory` is available for the MapForce Server object in the COM and .NET API. In the Java API, the method `setWorkingDirectory` is available.

If both the source and the target machines are Windows machines running on the local network, an alternative approach is to configure the mapping to read the database file from a common shared directory, as follows:

1. Store the database file in a common shared directory which is accessible by both the source and the target machine.
2. Right-click the database component on the mapping and add a connection to the database file using an absolute path.

Global Resources

If a mapping includes references to Global Resources instead of direct paths or database

connections, such references are preserved when you compile the mapping to a server execution file (.mfx), or when you deploy the mapping to FlowForce Server. For information about running such mappings with MapForce Server, see [Global Resources](#).

Note: FlowForce Server does not currently support Global Resources. Do not use Global Resources if you intend to execute the mapping with MapForce Server running under FlowForce Server management.

XBRL Taxonomy Packages

When you deploy a mapping that references XBRL Taxonomy Packages to FlowForce Server, MapForce collects all external references from the mapping and then resolves them using the current configuration and currently installed taxonomy packages. If there are resolved external references that point to a taxonomy package, then the taxonomy package is deployed together with the mapping. FlowForce Server will use that package—as it was during deployment—to execute the mapping. To refresh the taxonomy package used by FlowForce Server, you will need to change it in MapForce and redeploy the mapping.

Note that the root catalog of MapForce Server influences the way taxonomies are resolved on the target machine. The root catalog is found at the following path relative to the MapForce Server installation directory: **etc/RootCatalog.xml**.

Taxonomy packages that were deployed with a mapping will be used if the root catalog of MapForce Server does not already contain such package or does not contain a package that is defined for the same URL prefix. The root catalog of MapForce Server has priority over the deployed taxonomy.

If MapForce Server runs standalone (without FlowForce Server), it is possible to specify the root catalog that should be used by the mapping as follows:

- At the command line, this is possible by adding the option `-catalog` to the `run` command.
- In the MapForce Server API, call the method `SetOption`, and supply the string `"catalog"` as first argument, and the path to the root catalog as second argument.

If a mapping uses XBRL components with table linkbases, the taxonomy package or the taxonomy package configuration file must be supplied to the mapping at runtime, as follows:

- At the MapForce Server command line, add the option `--taxonomy-package` or `--taxonomy-packages-config-file` to the `run` command.
- In the MapForce Server API, call the method `SetOption`. The first argument must be either `"taxonomy-package"` or `"taxonomy-packages-config-file"`. The second argument must be the actual path to the taxonomy package (or taxonomy package configuration) file.

1.3 Global Resources

Altova Global Resources represent a way to refer to files, folders, or databases so as to make these resources reusable, configurable and available across multiple Altova applications. A typical usage of Global Resources is to define a database connection once and reuse it across all Altova applications that support Global Resources. For example, you can create a database connection on the machine where the MapForce mapping was designed and then reuse the same connection on the machine where MapForce Server runs the mapping. Global Resources have various other uses, including the ability to make the input or output file or folder paths configurable. For an introduction to Global Resources, refer to the "Global Resources" chapter of MapForce documentation.

If a mapping designed with MapForce uses Altova Global Resources, the same mapping can be run in MapForce Server, and the same Global Resources can be called from the machine where MapForce Server runs. This requires the following:

1. The file where the Global Resource is defined. By default, all Global Resources are defined in a file called **GlobalResources.xml**, which is available in the folder **Documents \Altova** folder on the machine where MapForce is installed. It is possible to create multiple Global Resource files if necessary, in order to make them portable, see also the MapForce documentation.
2. The Global Resource configuration name. Each Global Resource is identified by a configuration name. The default name is "Default" but you can create additional configurations if so required.

In MapForce, the Global Resource file path and the Global Resource configuration name are set or changed from the graphical user interface. In MapForce Server, these are specified at mapping runtime.

- If you run the mapping through the command line interface, set the options `--globalresourceconfig` and `--globalresourcefile` after the [run](#) command, for example:

```
C:\Program Files (x86)\Altova\MapForceServer2018\bin\MapForceServer.exe
run SomeMapping.mfx --globalresourcefile="C:\Users\me\Documents\Altova
\GlobalResources.xml" --globalresourceconfig="Default"
```


- If you run the mapping through the MapForce Server API, call the method [SetOption](#) two times before calling the [Run](#) method. The first call is required to supply the Global Resource XML file path as option, and the second one is required to supply the Global Resource configuration name (see also the [Code Examples](#) section).

1.4 Join Optimization

Join optimization accelerates execution of data mappings in which large sets of data are being filtered or joined.

Join optimization works by eliminating nested loops that occur internally as a mapping is being executed. A nested loop occurs when the mapping iterates each item of a set as many times as there are items in a second set. Note that it is normal for the mapping execution engine* to perform loops (iterations) over various sequences of items, by virtue of its design. When nested independent loops occur (that is, loops which iterate over other loops), the mapping can benefit from join optimization, which would significantly reduce the time required to execute the mapping. Nested loops are hardly noticeable when running mappings where the input data is not significantly large; however, this can become a challenge in case of mappings that process files or databases that consist of a very large number of records.

* The execution engine of a mapping can be MapForce, MapForce Server, or a C#, C++, or Java program generated by MapForce. Join optimization is available exclusively in the MapForce Server Advanced Edition.

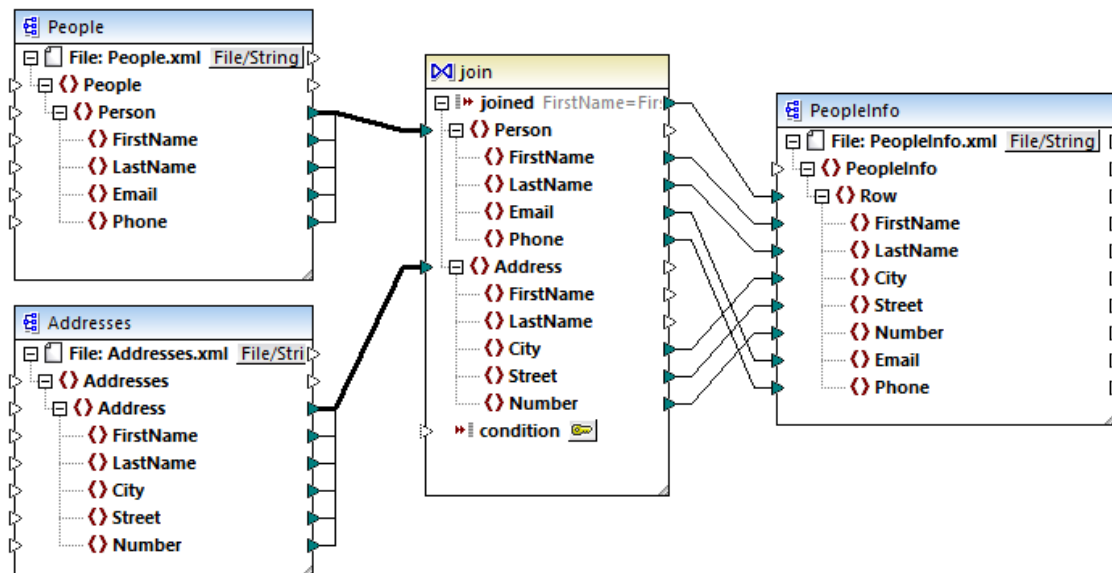
To designate MapForce Server as target execution engine, click the BUILT-IN () toolbar button in MapForce. This will also ensure your mapping benefits from most available features. If you select another transformation language, certain MapForce features might not be supported in that language.

As mentioned above, the primary concern of join optimization is to address nested loops in an efficient way. Let's now have a closer look at how nested loops occur in first place.

The typical case when nested loops occur is when the mapping contains at least one Join component, and SQL JOIN mode** is not possible.

** When certain conditions are met in MapForce, mappings could allow for a special execution mode called "SQL Join mode". SQL Join mode is possible only if the mapping reads data from a database. When data is joined this way, the join operation is undertaken by the database (that is, an SQL JOIN takes place), and this eliminates the need for nested loops in the mapping execution engine. For more information about SQL Join mode, refer to the MapForce documentation (<https://www.altova.com/documentation.html>).

For example, the image below illustrates a mapping (designed with Altova MapForce) which combines data from two XML files using a Join component. On the computer where MapForce is installed, this mapping is available at the following path: **..\Documents\Altova\MapForce2018\MapForceExamples\Tutorial\JoinPeopleInfo.mfd**. Some people data is available only in the first XML file (**Email**, **Phone**), while some other data is available only in the second XML file (**City**, **Street**, **Number**). The goal of the mapping is to write to the target XML file the merged data of all people where **FirstName** and **LastName** correspond in both source structures.



JoinPeopleInfo.mfd

In MapForce, a Join component pairs items in two sets according to some custom condition, which implies comparing each item in set 1 with each item in set 2. The total number of comparisons represents the cross-join (Cartesian product) of both sets. For example, if the first set contains 50 items, and if the second set contains 100 items, then a total of 5000 (50 x 100) comparisons will occur. In the mapping above, the sets that are being compared correspond to all instance items of the two XML structures connected to the Join component.

Note: Join optimization (a feature of MapForce Server Advanced Edition) should not be confused with Join components (a feature of MapForce). For more information about Join components, refer to the MapForce documentation (<https://www.altova.com/documentation.html>).

As expected, from a performance perspective, mappings that contain nested loops would need more time to run. Imagine a situation where both joined sets contain millions of records. This can easily affect performance, and this is where join optimization is useful. In very broad lines, join optimization behaves like a database engine that is optimized to look up (index) extremely large sets of data. Except that, as illustrated by the mapping above, join optimization deals not only with data originating from databases. Join optimization eliminates nested loops regardless of the data kind, by building, where possible, internal lookup tables which are queried at mapping runtime. This significantly improves the mapping performance and ultimately reduces the time required to execute the mapping.

Note: When join optimization occurs, running the mapping will take less time but typically require more memory as well. Be aware that memory usage patterns depend on various complex factors; therefore, observed behaviour may differ depending on the case.

Join optimization can accelerate not only mappings with joins, but also those which use filter components. In MapForce, a filter processes a sequence of items (that is, it checks a given Boolean condition for each instance of the item connected to the **node/row** input). If the Boolean condition is connected to a function which, in its turn, must iterate over another sequence of

items, and if the mapping context demands it, then a situation similar to a join happens. If the filter must perform a cross-comparison of each item in two sets, then it qualifies for join optimization.

In order for the mapping to benefit from join optimization, it must be run by MapForce Server Advanced Edition. To execute a mapping with MapForce Server Advanced Edition, open it in MapForce, and compile it to a mapping execution (.mfx) file using the menu command **File | Compile to MapForce Server Execution File**. Then run the .mfx file by using an API method in your language of choice, or the `run` command of the command line interface (see also [How It Works](#)).

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 7 SP1 with Platform Update, Windows 8, Windows 10

▼ Windows Server

Windows Server 2008 R2 SP1 with Platform Update or newer

▼ Installing MapForce Server

MapForce Server can be installed on Windows systems as follows:

- As a separate standalone server product called MapForce Server. To install MapForce Server, download and run the MapForce Server installer. Follow the on-screen instructions.
- As part of the FlowForce Server installation package. To install MapForce Server as part of the [FlowForce Server](#) package, download and run the FlowForce Server installer. Follow the on-screen instructions and make sure you check the option for installing MapForce Server.

The installers of both MapForce Server and [FlowForce Server](#) are available at the Altova Download Center (<http://www.altova.com/download.html>).

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

```
<ProgramFilesFolder>\Altova\MapForceServer2018\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 or [FlowForce Server](#) on Windows systems, an option is available that allows you to download and install [Altova LicenseServer](#) together with MapForce Server or [FlowForce 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 or [FlowForce 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: 2.7

▼ 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 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

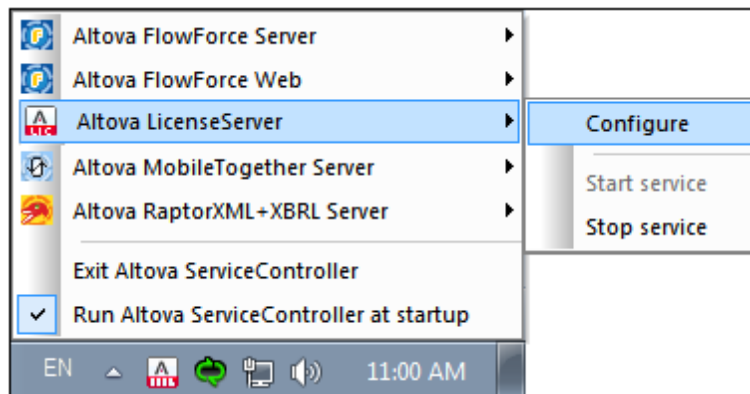
Altova ServiceController is started in order to start Altova LicenseServer.

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*).

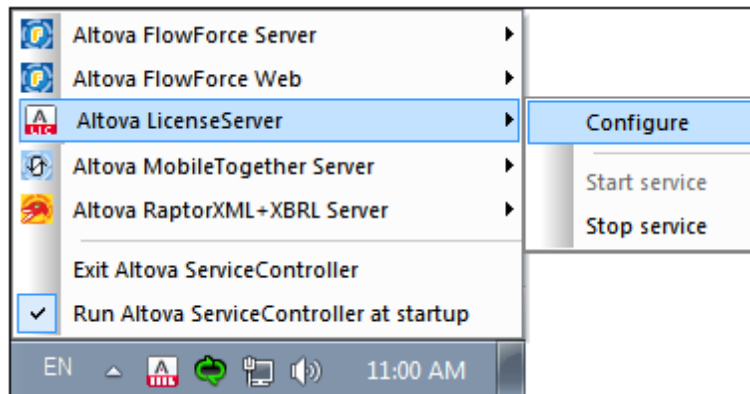


To specify that ServiceController starts automatically on logging in to the system, click the **ServiceController** icon in the system tray to display the **ServiceController** menu (*screenshot below*), and then toggle on the command **Run Altova ServiceController at Startup**. (This command is toggled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

If MapForce Server was installed as part of a [FlowForce Server](#) installation, registering FlowForce Server with LicenseServer will automatically also register MapForce Server. Essentially: (i) Start Altova FlowForce Web as a service via ServiceController (see *previous point*); (ii) Enter your password to access the Setup page; (iii) Select the LicenseServer name or address and click **Register with LicenseServer**. For more information, see [Register FlowForce Server](#).

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.

Note: FlowForce Server Advanced Edition and MapForce Server Advanced Edition will run only on machines with two or more cores.

When assessing the number of cores you should license, this decision should take into account the data volumes you need to process and the processing time your business environment is expected to allow for. In most scenarios, a larger number of cores means more volumes of data processed in less time. Below are just a few application-specific tips:

- FlowForce Server runs as a multi-threaded application. If the number of concurrent requests to the server is big, an insufficient number of cores will lead to latency (waiting times). For example, if you are exposing jobs as Web services, there may be hundreds of concurrent requests from clients. In this case, FlowForce Server will significantly benefit from a larger number of cores.
- MapForce Server will utilize a single core at a time, per mapping. Therefore, if you need to run multiple mappings simultaneously, a larger number of cores is highly recommended. For example, when MapForce Server runs under FlowForce Server management, several mapping jobs may overlap and run concurrently, depending also on the setup. Note, however, that if the volumes processed by your mappings are extremely big, latency could still occur.

In addition to the above, note that there are various external factors that typically influence the processing volumes and times that your server is capable to handle (for example, the hardware, the current load on the CPU, memory allocation of other applications running on the server). In order to get the most accurate performance measurements, it is generally advisable to first run the tools in your environment and expose them to actual factors and data specific to your business.

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 Server, 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](#)

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

The following libraries are required as a prerequisite to install and run the application. If the packages below are not already available on your Linux machine, run the command `yum` (or `apt-get` if applicable) to install them.

Required by	CentOS, RedHat	Debian	Ubuntu
LicenseServer	krb5-libs	libgssapi-krb5-2	libgssapi-krb5-2
MapForce Server Advanced Edition	libidn, krb5-libs	libidn11, libgssapi-krb5-2	libidn11, libgssapi-krb5-2

▼ 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/MapForceServer2018/etc/*.tool /opt/Altova/FlowForceServer2018/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 --get-selections | 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 mapforceserveradv
[CentOS, RedHat]:  sudo rpm -e mapforceserveradv
```

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 mapforceserveradv-2018-debian.deb
[Ubuntu]:  sudo dpkg --install mapforceserveradv-2018-ubuntu.deb
[CentOS]:  sudo rpm -ivh mapforceserveradv-2018-1.x86_64.rpm
[RedHat]:  sudo rpm -ivh mapforceserveradv-2018-1.x86_64.rpm
```

Note: You may need to adjust the name of the package above to match the current release or service pack version.

The MapForce Server package will be installed in the folder:

```
/opt/Altova/MapForceServer2018
```

▼ 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-2.7-debian.deb
[Ubuntu]:    sudo dpkg --install licenseserver-2.7-ubuntu.deb
[CentOS]:    sudo rpm -ivh licenseserver-2.7-1.x86_64.rpm
[RedHat]:    sudo rpm -ivh licenseserver-2.7-1.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](#). Also see the [LicenseServer documentation](#) for more detailed information.

▼ 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: 2.7

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

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/MapForceServer2018/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/MapForceServer2018/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/MapForceServer2018/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.

Note: FlowForce Server Advanced Edition and MapForce Server Advanced Edition will run only on machines with two or more cores.

When assessing the number of cores you should license, this decision should take into account the data volumes you need to process and the processing time your business environment is expected to allow for. In most scenarios, a larger number of cores means more volumes of data processed in less time. Below are just a few application-specific tips:

- FlowForce Server runs as a multi-threaded application. If the number of concurrent requests to the server is big, an insufficient number of cores will lead to latency (waiting times). For example, if you are exposing jobs as Web services, there may be hundreds of concurrent requests from clients. In this case, FlowForce Server will significantly benefit from a larger number of cores.
- MapForce Server will utilize a single core at a time, per mapping. Therefore, if you need to run multiple mappings simultaneously, a larger number of cores is highly recommended. For example, when MapForce Server runs under FlowForce Server management, several mapping jobs may overlap and run concurrently, depending also on the setup. Note, however, that if the volumes processed by your mappings are extremely big, latency could still occur.

In addition to the above, note that there are various external factors that typically influence the processing volumes and times that your server is capable to handle (for example, the hardware, the current load on the CPU, memory allocation of other applications running on the server). In order to get the most accurate performance measurements, it is generally

advisable to first run the tools in your environment and expose them to actual factors and data specific to your business.

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 4

Setup on macOS

4 Setup on macOS

This section describes the [installation](#) and [licensing](#) of MapForce Server on macOS systems.

[Installation on macOS](#)

- [System requirements](#)
- [Uninstall old versions of Altova server products](#)
- [Download the macOS package](#)
- [Install MapForce Server](#)
- [Altova LicenseServer](#)
- [LicenseServer versions](#)
- [Trial license](#)

[Licensing on macOS](#)

- [Start LicenseServer](#)
- [Register MapForce Server](#)
- [Assign a license](#)

4.1 Installation on macOS

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

- ▼ System requirements

- ▼ *(Mac) OS X, macOS*

- OS X 10.10 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/MapForceServer2018/etc/*.tool /usr/local/Altova/FlowForceServer2018/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/MapForceServer2018/
```

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/MapForceServer2018 (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 macOS](#).

▼ 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: 2.7

4.2 Licensing on macOS

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/MapForceServer2018/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/MapForceServer2018/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/MapForceServer2018/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.

Note: FlowForce Server Advanced Edition and MapForce Server Advanced Edition will run only on machines with two or more cores.

When assessing the number of cores you should license, this decision should take into account the data volumes you need to process and the processing time your business environment is expected to allow for. In most scenarios, a larger number of cores means more volumes of data processed in less time. Below are just a few application-specific tips:

- FlowForce Server runs as a multi-threaded application. If the number of concurrent requests to the server is big, an insufficient number of cores will lead to latency (waiting times). For example, if you are exposing jobs as Web services, there may be hundreds of concurrent requests from clients. In this case, FlowForce Server will significantly benefit from a larger number of cores.
- MapForce Server will utilize a single core at a time, per mapping. Therefore, if you need to run multiple mappings simultaneously, a larger number of cores is highly recommended. For example, when MapForce Server runs under FlowForce Server management, several mapping jobs may overlap and run concurrently, depending also on the setup. Note, however, that if the volumes processed by your mappings are extremely big, latency could still occur.

In addition to the above, note that there are various external factors that typically influence the processing volumes and times that your server is capable to handle (for example, the hardware, the current load on the CPU, memory allocation of other applications running on the server). In order to get the most accurate performance measurements, it is generally advisable to first run the tools in your environment and expose them to actual factors and data specific to your business.

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 Server, 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 5

MapForce Server Command Line

5 MapForce Server Command Line

This section:

- [Default location of MapForce Server executable](#)
- [Usage and list of CLI commands](#)

Default location of MapForce Server executable

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

<i>Windows</i>	<code><ProgramFilesFolder>\Altova\MapForceServer2018\bin MapForceServer.exe</code>
<i>Linux</i>	<code>/opt/Altova/MapForceServer2018/bin/mapforceserver</code>
<i>Mac</i>	<code>/usr/local/Altova/MapForceServer2018/bin/mapforceserver</code>

Usage and list of CLI commands

General command line syntax for MapForceServer is:

<i>Windows</i>	<code>MapForceServer --h --help --version <command> [options] [arguments]</code>
<i>Linux</i>	<code>mapforceserver --h --help --version <command> [options] [arguments]</code>
<i>Mac</i>	<code>mapforceserver --h --help --version <command> [options] [arguments]</code>

▼ 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

<code>--h --help</code>	Displays the help text.
<code>--version</code>	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: <code>help run</code>

5.1 assignlicense (Windows only)

This section:

- [Command syntax](#)
- [Command description](#)
- [Examples](#)
- [Command options](#)

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 <code>true false</code> . If <code>true</code> , then the license file is only uploaded to LicenseServer and validated. It is not assigned to MapForce Server. Form: <code>--test-only=true false</code> . Set to <code>true</code> if the option is specified without a value.
-----	-------------	--

5.2 exportresourcestrings

This section:

- [Command syntax](#)
- [Arguments](#)
- [Command description](#)
- [Examples](#)
- [Creating localized versions of MapForce Server](#)

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\MapForceServer2018\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 `MapForceServer2018_lc.dll`. The `_lc` part of the name contains the language code. For example, in `MapForceServer2018_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.

5.3 help

This section:

- [Command syntax](#)
- [Command description](#)
- [Examples](#)
- [The --help option](#)

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.

5.4 licenseserver

This section:

- [Command syntax](#)
- [Command description](#)
- [Examples](#)
- [Command options](#)

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 specify, 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: <code>--json=true/false</code>
-----	--------	---

5.5 run

The `run` command executes a MapForce Server execution file (.mfx file) supplied as argument. The MapForce Server execution file is created with MapForce; it essentially represents a mapping compiled for server execution.

Any input files required by the mapping are expected to be at the path specified at mapping design time in MapForce. If MapForce Server does not run on the same operating system as MapForce, the input files required by the mapping must be copied to the target machine alongside with the .mfx file, and must be referenced using a relative path. For information about configuring a mapping with respect to relative or absolute paths, refer to MapForce documentation (<https://www.altova.com/documentation#mapforce>). Other prerequisites may apply, depending on how you designed the mapping, see [Preparing Mappings for Server Execution](#).

If the mapping returns 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.

Syntax

Windows **MapForceServer** run [options] MfxFile

Linux **mapforceserver** run [options] MfxFile

Mac **mapforceserver** run [options] MfxFile

▼ 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.

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.

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.

	<code>--catalog</code>	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: <code>--catalog=FILE</code>
<code>--cert</code>	<code>--certificatespath</code>	This option 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: <code>--certificatespath=DIRECTORY</code> See also Digital Certificate Management .
<code>--gc</code>	<code>--globalresourceconfig</code>	This option is applicable if the mapping consumes Altova Global Resources (see Global Resources). It specifies the name of the global resource configuration (alias). This option must be used together with the <code>--globalresourcefile</code> option. Form: <code>--gc=VALUE</code>
<code>--gr</code>	<code>--globalresourcefile</code>	This option is applicable if the mapping consumes Altova Global Resources (see Global Resources). It specifies the path of the global resource definition file. This option must be used together with the <code>--globalresourceconfig</code> option. Form: <code>--gr=FILE</code> .
<code>--l</code>	<code>--lang</code>	The language used for displaying messages. Form: <code>--lang=VALUE</code> (<code>en, de, ja, es, fr</code>)
<code>--p</code>	<code>--param</code>	This option is applicable if the mapping was designed to take input parameters. It assigns a value to a parameter defined in the mapping. Form: <code>--param=ParamName:ParamValue</code> . The <code>--param</code>

		<p>switch must be used before each parameter. Use quotes if <i>ParamName</i> or <i>ParamValue</i> contains a space. For example: <code>--p=company: "Nanonull Inc"</code>.</p> <p>For more information about mappings that take input parameters, refer to MapForce documentation (https://www.altova.com/documentation#mapforce).</p>
	<code>--taxonomy-package</code>	<p>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.</p> <p>Form: <code>--taxonomy-package=FILE</code></p>
	<code>--taxonomy-packages-config-file</code>	<p>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.</p> <p>Form: <code>--taxonomy-packages-config-file=FILE</code></p>

Examples

This example shows you how to run a mapping execution file (.mfx) with MapForce Server on Windows. The mapping used in this example reads an input file, **Employees.xml**, and produces two output files (**PersonList.xml** and **Contacts.xml**).

First, let's generate the MapForce Server execution (.mfx) file, as follows:

1. Run MapForce and open the following MapForce design file (.mfd file): **C:\Users\<user>\Documents\Altova\MapForce2018\MapForceExamples\ChainedPersonList.mfd**.
2. On the **File** menu, click **Compile to MapForce Server Execution File**.
3. When prompted, save the .mfx file to **C:\temp** directory. This will be the working directory where the mapping will be executed by MapForce Server.

Next, let's open a command line prompt and change the working directory to **C:\temp**.

```
cd C:\temp
```

Finally, run the following command to execute **ChainedPersonList.mfx**. In this example, MapForce Server is called using an absolute path. (To call it with a relative path, add the executable's path to your system's `PATH` environment variable).

```
"C:\Program Files (x86)\Altova\MapForceServer2018\bin\MapForceServer.exe" run  
ChainedPersonList.mfx
```

The two output files (**PersonList.xml** and **Contacts.xml**) are generated in the working directory. Importantly, this mapping is configured to use absolute paths, which is why the mapping ran successfully and did not require that the input **Employees.xml** file exists in the working directory. The **Employees.xml** file actually exists in the MapForce Examples folder mentioned above and is referenced through an absolute path. To specify whether paths should be treated as absolute or relative, right-click the mapping in MapForce, select **Mapping Settings**, and then select or clear the **Make paths absolute in generated code** check box. Whenever you change the mapping settings, make sure to re-compile the mapping to .mfx. For more information, see [Preparing Mappings for Server Execution](#).

5.6 setdeflang

This section:

- [Command syntax](#)
- [Command description](#)
- [Examples](#)
- [Supported languages](#)

Command syntax

Windows `MapForceServer setdeflang | sdl LanguageCode`

Linux `mapforceserver setdeflang | sdl LanguageCode`

Mac `mapforceserver setdeflang | sdl LanguageCode`

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 verifylicense (Windows only)

This section:

- [Command syntax](#)
- [Command description](#)
- [Examples](#)
- [Command options](#)

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=ABCDEFGH-IJKLMNOP-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: <code>--license-key=VALUE</code>
-----	---------------	---

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`.

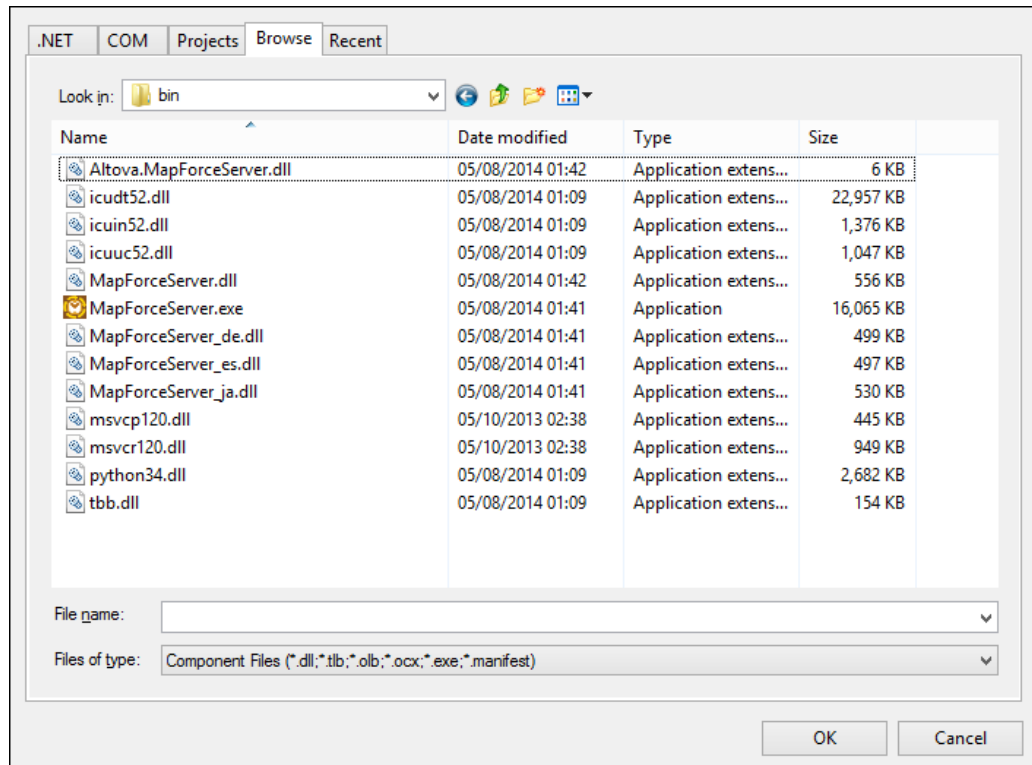
During installation, MapForce Server will be registered automatically as a COM server object, so there is no need for a manual registration. If you receive an access error, open the Component Services and give permissions to the same account that runs the application pool containing MapForce Server.

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.

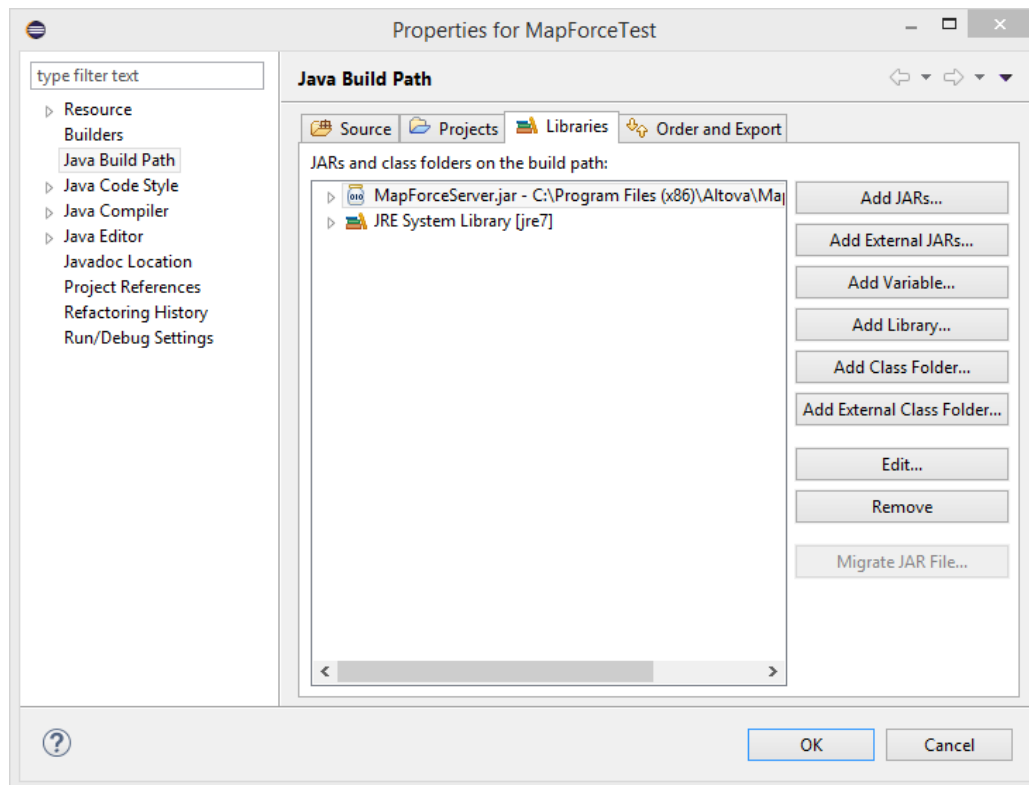
MapForceServer.jar	The library that communicates with MapForce Server.
MapForceServer_JavaDoc.zip	A Javadoc archive file containing documentation for the Java MapForce Server API.

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.

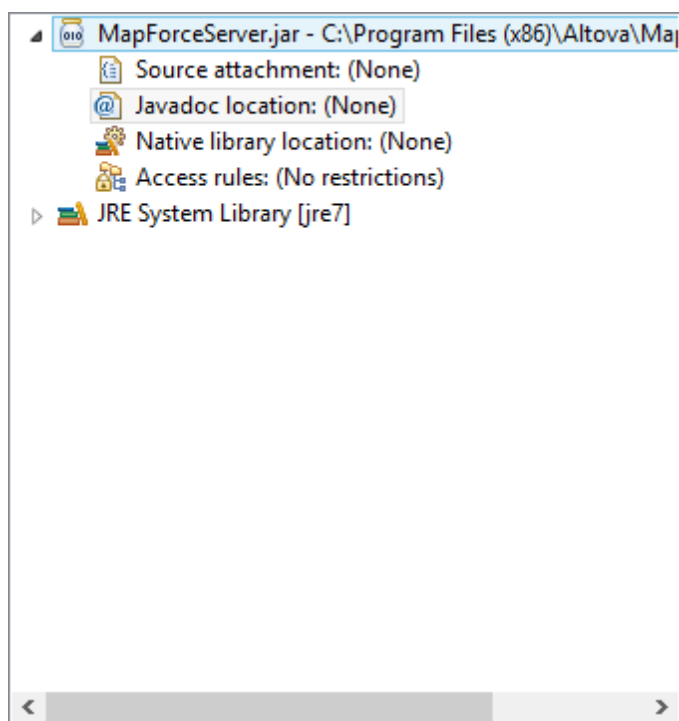
In Eclipse, you can add the classpath references by editing the properties of the Java project, as shown in the sample instructions below (notice the instructions apply to Eclipse 4.4).

To add the MapForce Server library references in Eclipse:

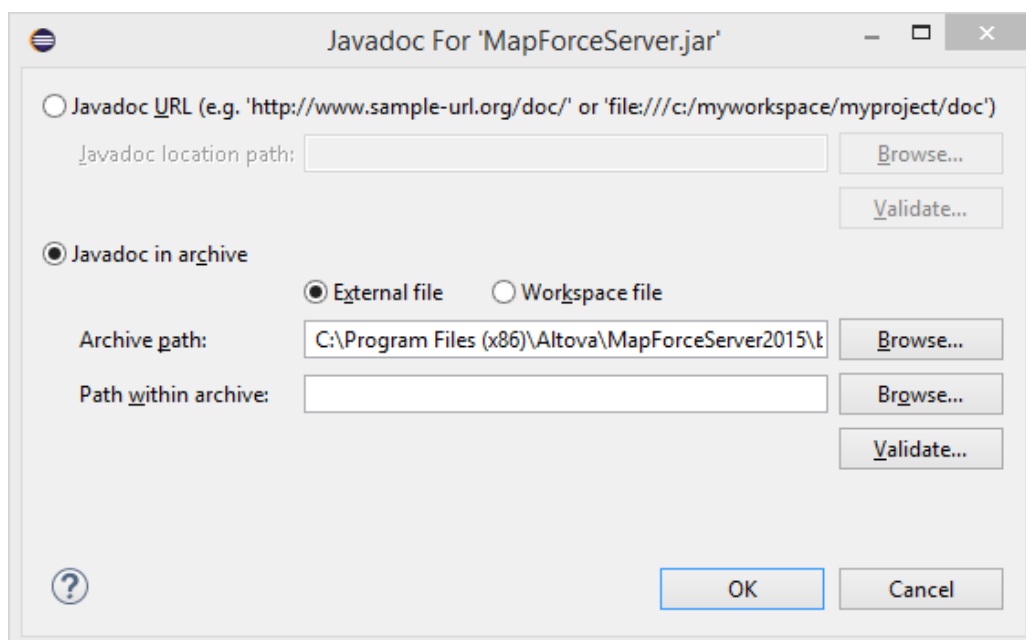
1. With the project open in Eclipse, on the **Project** menu, click **Properties**.
2. Click **Java Build Path**.
3. On the **Libraries** tab, click **Add External JARs**, and then browse for the `MapForceServer.jar` file located in the MapForce Server installation folder.



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



5. 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.



6. 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/
MapForceServer2018/bin/MapForceServer.jar">
    <attributes>
      <attribute name="javadoc_location"
value="jar:file:/C:/Program%20Files%20(x86)/Altova/MapForceServer2018/
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:\\Program Files (x86)\\Altova\\
\\MapForceServer2018\\etc\\Examples";

            //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\\
\\MapForceServer2018\\bin\\MapForceServer.exe";

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

            //Prepare the parameters, if your 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\\
\\MapForceServer2018\\etc\\Examples\\TokenizeString.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:\\Program Files (x86)\\Altova\\
        \\MapForceServer2018\\etc\\Examples";

        //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\\
        \\MapForceServer2018\\bin\\MapForceServer.exe";

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

        //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\\MapForceServer2018\\etc\\  
\\Examples\\TokenizeString.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:\\Program Files (x86)\\Altova\\
\\MapForceServer2018\\etc\\Examples" );

            //Default path to the MapForce Server executable is the installation
            path (same dir with the MapForceServer.jar)
            //In case you copied the JAR file to a new location, you need to
            explicitly set the path to the .exe file
            objMFS.setServerPath( "C:\\Program Files (x86)\\Altova\\
\\MapForceServer2018\\bin\\MapForceServer.exe" );

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

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

            //Run the mapping; input and output paths are stored inside the MFX
            file
            // NOTE Please adapt the path to the input file in order to run the
            sample
            if ( objMFS.run( "C:\\Program Files (x86)\\Altova\\MapForceServer2018
            \\etc\\Examples\\TokenizeString.mfx" ) )
                System.out.println( "Success - finished execution" );
            else
                System.out.println( objMFS.getLastExecutionMessage() );
        }
        catch ( Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

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


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:\Program Files (x86)\Altova\MapForceServer2018
\etc\Examples"

'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\MapForceServer2018\bin
\MapForceServer.exe"

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

'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\MapForceServer2018\etc
\Examples\TokenizeString.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:\Program Files (x86)\Altova
        \MapForceServer2018\etc\Examples"

        '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\MapForceServer2018
        \bin\MapForceServer.exe"

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

        '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\MapForceServer2018\etc
        \Examples\TokenizeString.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:

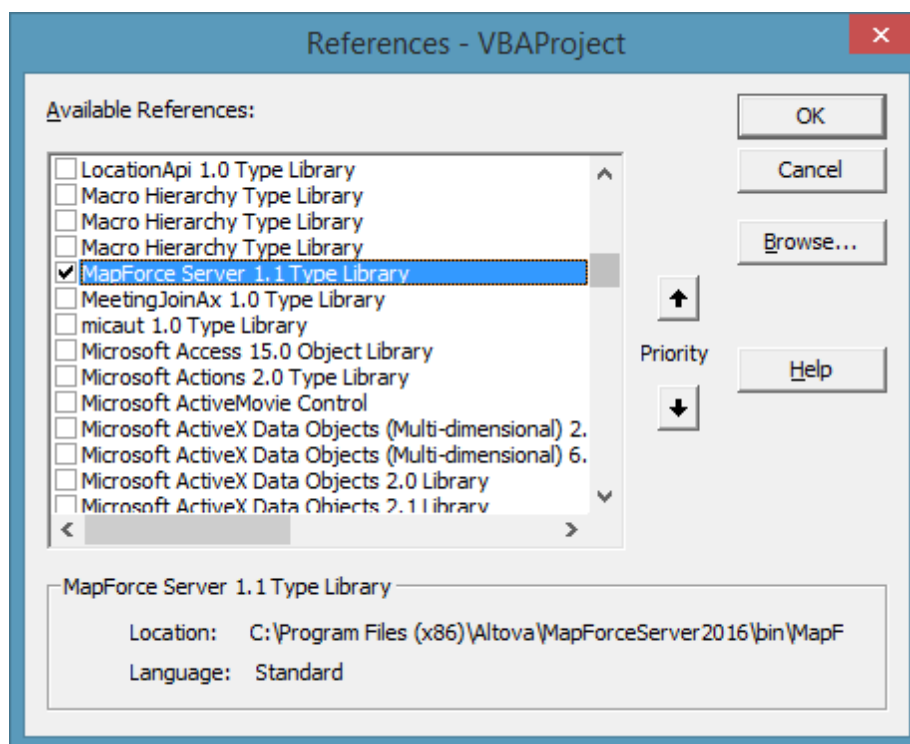
1. Microsoft Office and MapForce Server must be installed on the same machine.
2. The architecture of MapForce Server (32-bit or 64-bit) must match that of Microsoft Office. For example, if you run VBA on Microsoft Office 32-bit, make sure that you use MapForce Server 32-bit. To find out whether your 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**.



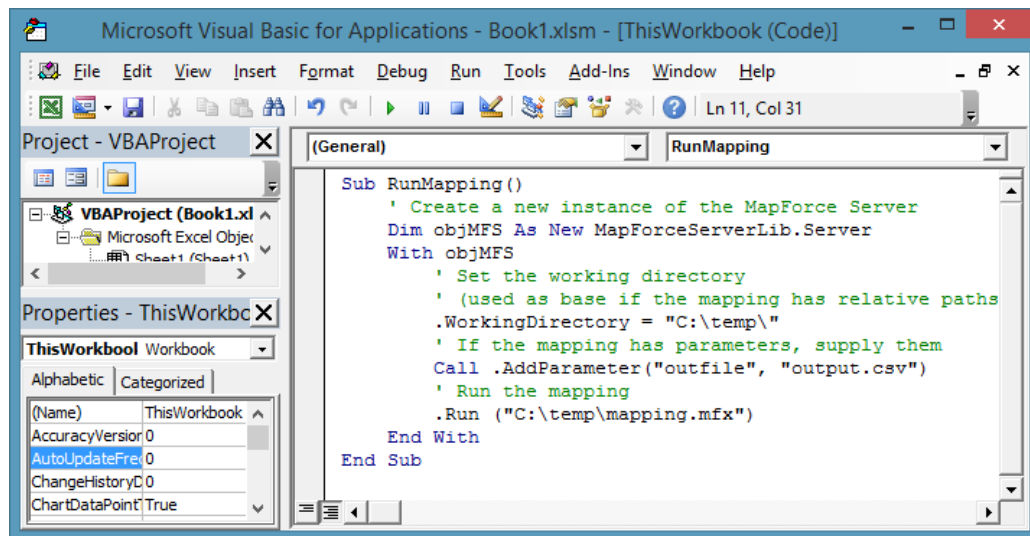
3. Click to select the **MapForce Server Type Library**.

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.

The MapForce Server API exposes the following interface:

- [IServer](#)

6.5.1 IServer Interface

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. The first argument specifies the name of the parameter as defined on the mapping; the second argument specifies the parameter value.

▼ 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)
```

ClearOptions method

Clears the list of options previously set through the `SetOption` method. All options set through the `SetOption` method are valid for the lifetime of the object. Call this method if you need to explicitly clear all of the previously set options.

▼ C#

```
void ClearOptions()
```

▼ C++

```
HRESULT ClearOptions();
```

▼ VB

```
Sub ClearOptions()
```

ClearParameterList method

Clears the list of parameters that were previously set using the `AddParameter` method.

▼ 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
```

SetOption method

Sets an option before running the mapping. The first argument specifies the name of the option, while the second argument specifies the option value. This method is particularly useful when a

mapping was designed to consume Global Resources (see [Altova Global Resources](#)). The currently supported options are as follows:

Option name	Description
globalresourcefile (or gr)	A Global Resource file path. (When this option is specified, then a Global Resource configuration name must also be specified, see next item).
globalresourceconfig (or gc)	A Global Resource configuration name. (When this option is specified, then a Global Resource file path must also be specified, see previous item).
catalog	The path to a custom RootCatalog.xml file. This option enables you to specify a custom catalog file used to resolve URLs used by the mapping. The default catalog is in the etc subdirectory of the program installation directory.
taxonomy-package	The path to a custom XBRL taxonomy package, if one is required by the mapping.
taxonomy-packages-config-file	The path to a custom XBRL taxonomy package configuration, if one is required by the mapping.

All set options are valid for the lifetime of the object. If you set an option with the same name twice, the previous option will be overridden. To explicitly clear all options, call the `ClearOptions()` method.

▼ C#

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

▼ C++

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

▼ VB

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

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, as number of years starting from 1998 (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.

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

- [MapForceServer Class](#)
- [MapForceServerException Class](#)

6.6.1 MapForceServer Class

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)
```

`clearOptions` method

Clears the list of options previously set through the `SetOption` method. All options set through the `SetOption` method are valid for the lifetime of the object. Call this method if you need to explicitly clear all of the previously set options.

```
public void clearOptions()
```

`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, as number of years starting from 1998 (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
```

setOption method

Sets an option before running the mapping. The first argument specifies the name of the option, while the second argument specifies the option value. This method is particularly useful when a mapping was designed to consume Global Resources (see [Altova Global Resources](#)). The currently supported options are as follows:

Option name	Description
globalresourcefile (or gr)	A Global Resource file path. (When this option is specified, then a Global Resource configuration name must also be specified, see next item).
globalresourceconfig (or gc)	A Global Resource configuration name. (When this option is specified, then a Global Resource file path must also be specified, see previous item).
catalog	The path to a custom RootCatalog.xml file. This option enables you to specify a custom catalog file used to resolve URLs used by the mapping. The default catalog is in the etc subdirectory of the program installation directory.
taxonomy-package	The path to a custom XBRL taxonomy package, if one is required by the mapping.
taxonomy-packages-config-file	The path to a custom XBRL taxonomy package configuration, if one is required by the mapping.

All set options are valid for the lifetime of the object. If you set an option with the same name twice, the previous option will be overridden. To explicitly clear all options, call the `ClearOptions()` method.

```
public void setOption(java.lang.String option_name,
                     java.lang.String value)
```

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)
```

6.6.2 MapForceServerException Class

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

```
public class MapForceServerException extends Exception
```


Chapter 7

Altova LicenseServer

7 Altova LicenseServer

Altova LicenseServer (hereafter also called **LicenseServer** for short) provides a central location for the management of licenses for Altova products. Altova applications running in a network can have licenses assigned to them from the LicenseServer, thus giving administrators the flexibility to manage and monitor licenses.

Current version: 2.7 *

** LicenseServer 2.7 can be used to license (i) Altova software products of version 2018r2 or older, and (ii) Altova MobileTogether Server version 4.1 or older. It cannot be used to license Altova product versions that are newer than those listed in this note. For more information about compatibility, see [Updating LicenseServer](#).*

Licensing process with Altova LicenseServer

To assign a license to an Altova product via Altova LicenseServer, you need to do the following:

1. [Start LicenseServer](#)
2. Open the [LicenseServer Configuration page](#), which is the Web UI of LicenseServer, on [Windows](#), [Linux](#), or [macOS](#).
3. [Upload the Altova product license/s](#) you have received from Altova to the license pool of LicenseServer. Do this in the [License Pool](#) tab of the Configuration page.
4. [Register the Altova product/s](#) with LicenseServer.
5. In the [Client Management](#) tab of the Configuration page, [assign licenses](#) to the Altova product/s.

Licenses can thereafter be conveniently monitored and managed centrally with LicenseServer. See the [Configuration Page Reference](#) for available functionality.

Note: The [LicenseServer Configuration page](#) does not support SSL.

▼ LicenseServer versions and their compatibility with Altova products

New versions of Altova products can only be licensed with the version of LicenseServer that is the latest at the time of the Altova product's release. However, older versions of Altova products will work with newer versions of LicenseServer.

So, if you are installing a new version of an Altova product and if your current LicenseServer version is not the latest, de-install this older LicenseServer version and install the latest version that is available on the Altova website. All registration and licensing information held in your older version of LicenseServer will be saved at the time of de-installation to a database on your machine, and will be imported automatically into the newer version of LicenseServer. When you install a newer version of LicenseServer, the older version will be de-installed before 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: 2.7

About this documentation

This documentation is organized into the following parts:

- Introductory information about: [network requirements](#); installation on [Windows](#), [Linux](#), and [macOS](#); and [Altova ServiceController](#).
- [How to Assign Licenses](#), which describes in a step-by-step way how to assign licenses with Altova LicenseServer.
- [Configuration Page Reference](#): A description of the administrator's interface with LicenseServer.

Last updated: 21 June 2018

7.1 Network Information

Altova LicenseServer must be installed on a server machine that is accessible by all clients running Altova products that require a license. Any firewall on both the client and server must allow the flow of network traffic to and from the LicenseServer that is necessary for the LicenseServer to operate correctly.

On the LicenseServer machine, **port 35355** is used to distribute licenses, and therefore it must be open for network traffic with client machines.

The following are the default networking parameters and requirements of LicenseServer:

- *For LicenseServer license distribution:*
Either one or both of
IPv4 TCP connection on port 35355
IPv6 TCP connection on port 35355

For administrative tasks, the LicenseServer is accessed by a web interface that uses port 8088. The port used can be [configured to suit your requirements](#).

Connection to the Master Licensing Server at altova.com

Altova LicenseServer needs to be able to communicate with the Master Licensing Server at altova.com to validate and authenticate license-related data and to ensure continuous compliance with the Altova license agreements. This communication occurs over HTTPS using port 443. If Altova LicenseServer, after making the initial verification with the altova.com Master Licensing Server, is unable to again connect with altova.com for a duration of more than 5 days (= 120 hours), then Altova LicenseServer will no longer permit the usage of any Altova software products connected to that Altova LicenseServer.

Any such loss of connection with the altova.com master servers will be logged in the [Messages tab](#) of the [Configuration page of Altova LicenseServer](#). In addition, the administrator can configure the Altova LicenseServer to automatically send an alert email when the connection to altova.com is lost. Alert Mail settings are available in the [Settings tab](#) of the [Configuration page](#).

How LicenseServer identifies client machines

When an Altova product is licensed via LicenseServer, the license is recorded by LicenseServer as having been issued to a specific client machine. LicenseServer identifies the client machine by its hostname, which LicenseServer obtains by a DNS-server-lookup of the client's IP address. If no hostname can be obtained from the lookup, then the client's IP address is used as its identifier. Subsequently, every time the Altova product is started on the client machine, LicenseServer checks whether the client-IP-address it receives matches the hostname or IP-address of any client for which a license was issued. If there is a match, then the license details are verified.

VPN connections with dynamic IP addresses

If a client machine connects to LicenseServer via a Virtual Private Network (VPN) service, then the

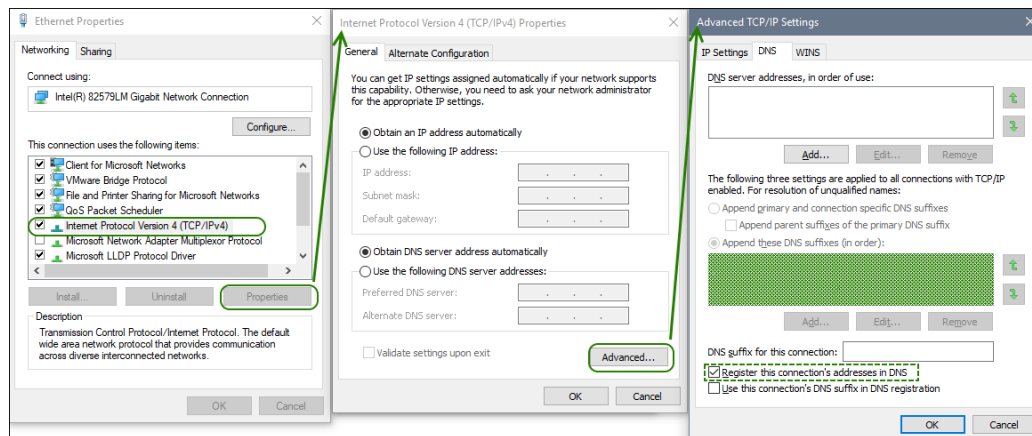
connection is often made with a dynamically assigned IP address. In this case, each new connection by the client will be recorded by LicenseServer as coming from a new IP address—and, therefore, from a previously unknown client (see *"How LicenseServer identifies client machines"* above).

This results in the following:

- If an additional license **is available** on LicenseServer at the time when the client connects, then a new license will be assigned to the client (which might already have had one or more licenses assigned to it previously). Licenses that were assigned previously to the client are not released. As a consequence, a single client would be consuming multiple licenses, leading to a shortage of licenses on the server.
- If an additional license **is not available** on LicenseServer, then the product on that client cannot be licensed—even though licenses might have been assigned to it previously (for IP addresses that the client is no longer using).

There are two ways to resolve this issue:

- *Register VPN clients with a DNS.* How to do this varies according to OS. On Windows 10 clients, for example, you can set this up via the properties of the client's network connection (see screenshot below).



- *Assign a static IP address to the VPN user.* How to do this differs from one VPN software to the other. You can get the relevant information from your VPN provider or administrator. For example, how to do this with OpenVPN is described [here](#).

7.2 Updating LicenseServer

Each new release of Altova server products (such as RaptorXML Server) is accompanied by the release of a newer version of LicenseServer (the corresponding version). **If you update your Altova server product, you must also update your LicenseServer** to the corresponding LicenseServer version. This is because the updated Altova server product cannot be licensed with an older LicenseServer version than the corresponding version. Note, however, that LicenseServer versions are backwards-compatible. (This means that LicenseServer can be used to license corresponding and older versions of Altova server products.)

Update LicenseServer as follows:

- *On Windows systems:* Double-click the new version's installer (executable file) or call the installer from the command line. The installer will uninstall the older version of LicenseServer and install the new version.
- *On Linux and macOS:* Uninstall the older version manually and then start the installer file of the new version.

The same conditions apply as when you originally installed LicenseServer (see for [Windows](#), [Linux](#), [macOS](#)).

7.3 Installation (Windows)

Altova LicenseServer can be installed on Windows systems in one of two ways:

- As an independent installation.
- As part of an Altova server product installation. (Altova server products are: Altova FlowForce Server, Altova MapForce Server, Altova StyleVision Server, Altova RaptorXML(+XBRL), and Altova MobileTogether Server.) If LicenseServer is not installed on your system at the time an Altova server product is installed, the option to install LicenseServer is selected by default during installation setup. If LicenseServer is already installed, the option to install it is deselected by default. You can change the default option if you like.

For information about assigning licenses via LicenseServer, see the section [How to Assign Licenses](#).

System requirements

▼ Windows

Windows 7 SP1 with Platform Update, Windows 8, Windows 10

▼ Windows Server

Windows Server 2008 R2 SP1 with Platform Update or newer

▼ LicenseServer versions and their compatibility with Altova products

New versions of Altova products can only be licensed with the version of LicenseServer that is the latest at the time of the Altova product's release. However, older versions of Altova products will work with newer versions of LicenseServer.

So, if you are installing a new version of an Altova product and if your current LicenseServer version is not the latest, de-install this older LicenseServer version and install the latest version that is available on the Altova website. All registration and licensing information held in your older version of LicenseServer will be saved at the time of de-installation to a database on your machine, and will be imported automatically into the newer version of LicenseServer. When you install a newer version of LicenseServer, the older version will be de-installed before 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: 2.7

The version number of the LicenseServer that is appropriate for any particular version of a server product is displayed during the installation of that version of the server product. You can choose to install this version of LicenseServer along with the server product, or you can install the newer version of LicenseServer separately. In both cases, the installer will automatically de-install the previous version and install the new version.

7.4 Installation (Linux)

Altova LicenseServer can be installed on Linux systems (Debian, Ubuntu, CentOS, RedHat).

System requirements

▼ Linux

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

The following libraries are required as a prerequisite to install and run the application. If the packages below are not already available on your Linux machine, run the command `yum` (or `apt-get` if applicable) to install them.

Required by	CentOS, RedHat	Debian	Ubuntu
LicenseServer	krb5-libs	libgssapi-krb5-2	libgssapi-krb5-2
MapForce Server Advanced Edition	libidn, krb5-libs	libidn11, libgssapi-krb5-2	libidn11, libgssapi-krb5-2

Uninstalling old versions of LicenseServer

On the Linux command line interface (CLI), you can check whether LicenseServer is installed with the following command:

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

If LicenseServer is not installed, go ahead with the installation as documented in the next steps. If LicenseServer is installed and you wish to install a newer version of it, uninstall the old version with the command:

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

Installing Altova LicenseServer

On Linux systems, LicenseServer must be installed independently of other Altova server products. It is not included as part of the installation packages of Altova server products. Download Altova LicenseServer from the [Altova website](#) and copy the package to any directory on the Linux system.

Distribution	Installer extension
--------------	---------------------

Debian	.deb
Ubuntu	.deb
CentOS	.rpm
RedHat	.rpm

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 LicenseServer as a root user. If you do not wish to be logged in as root, you can use the `sudo` command to temporarily exercise root-user privileges. Install LicenseServer with the following command:

```
[Debian]:  sudo dpkg --install licenseserver-2.7-debian.deb
[Ubuntu]:  sudo dpkg --install licenseserver-2.7-ubuntu.deb
[CentOS]:  sudo rpm -ivh licenseserver-2.7-1.x86_64.rpm
[RedHat]:  sudo rpm -ivh licenseserver-2.7-1.x86_64.rpm
```

- The LicenseServer package will be installed in: `/opt/Altova/LicenseServer/bin`
- A user named `altovalicenseserver` will be created, which will have the necessary rights to run LicenseServer. When LicenseServer is started as a daemon (or service), it will be started automatically as this user. (You can also, of course, (i) create a new user with privileges that will enable this user to access LicenseServer, or (ii) start LicenseServer as root user (which has maximum privileges), but neither of these steps is either required or recommended, because (i) they each have their disadvantages, and (ii) the `altovalicenseserver` user has all the rights that are needed to run LicenseServer and is a separate and dedicated user.)

Note: You can also run LicenseServer as `altovalicenseserver` in interactive mode. For example on CentOS 7, you can debug like this: `sudo runuser -l altovalicenseserver -c '/opt/Altova/LicenseServer/bin/licenseserver debug'`. The disadvantage of the interactive mode is that LicenseServer will stop running when the terminal session is closed.

Background information about running LicenseServer

Note the following points:

- It is best to run LicenseServer as a daemon (or service). If you run LicenseServer in interactive mode, then it will stop running when the terminal session is closed.
- When running LicenseServer as a daemon, it is best to manage LicenseServer via the `initctl` or `systemctl` command. Both of these commands can be run only with root-user privileges. So, if you are running as a non-root user, then use the `sudo` command to temporarily gain root-user privileges (`sudo initctl...` and `sudo systemctl...`).
- Once LicenseServer has been started as a daemon, it will automatically run as the

`altovalicenserver` user, which (i) was created by the LicenseServer installer at installation time, and (ii) has all the necessary rights for running LicenseServer. For more information, see the section about [starting LicenseServer](#).

For information about how to proceed with assigning licenses, see the section [How to Assign Licenses](#).

▼ LicenseServer versions and their compatibility with Altova products

New versions of Altova products can only be licensed with the version of LicenseServer that is the latest at the time of the Altova product's release. However, older versions of Altova products will work with newer versions of LicenseServer.

So, if you are installing a new version of an Altova product and if your current LicenseServer version is not the latest, de-install this older LicenseServer version and install the latest version that is available on the Altova website. All registration and licensing information held in your older version of LicenseServer will be saved at the time of de-installation to a database on your machine, and will be imported automatically into the newer version of LicenseServer. When you install a newer version of LicenseServer, the older version will be de-installed before 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: 2.7

7.5 Installation (macOS)

Altova LicenseServer can be installed on (Mac) OS X, macOS systems (see *system requirements below*). Since you might need to uninstall a previous version, uninstalling is described first.

System requirements

▼ (Mac) OS X, macOS

OS X 10.10 or newer

Uninstalling old versions of LicenseServer

Before uninstalling LicenseServer, stop the service 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 terminal and make sure that LicenseServer is not in the list.

In **Applications**, right-click the LicenseServer 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/LicenseServer
```

Installing Altova LicenseServer

Open the download page <http://www.altova.com/download.html> and locate the Altova LicenseServer installer under "Server Software Products" for Mac. After downloading the disk image (`.dmg`) file, click to open it. This mounts a new virtual drive on your computer. On the virtual drive, double-click the package (`.pkg`) file, and follow the on-screen instructions. You will need to accept the license agreement for installation to proceed.

Note the following points:

- The LicenseServer package will be installed in: `/usr/local/Altova/LicenseServer`
- A user named `altovalicenser` will be created, which will have the necessary rights to run LicenseServer. When LicenseServer is started as a daemon (or service), it will be started automatically as this user.

Background information about running LicenseServer

Note the following points:

- It is best to run LicenseServer as a daemon (or service). If you run LicenseServer in

- interactive mode, then it will stop running when the terminal session is closed.
- When running LicenseServer as a daemon, it is best to manage LicenseServer via the `launchctl` command. This command can be run only with root-user privileges. So, if you are running as a non-root user, then use the `sudo` command to temporarily gain root-user privileges (`sudo launchctl...`).
 - Once LicenseServer has been started as a daemon, it will automatically run as the `altovalicenser` user, which (i) was created by the LicenseServer installer at installation time, and (ii) has all the necessary rights for running LicenseServer. For more information, see the section about [starting LicenseServer](#).

To eject the virtual drive after installation, right-click it, and select **Eject**.

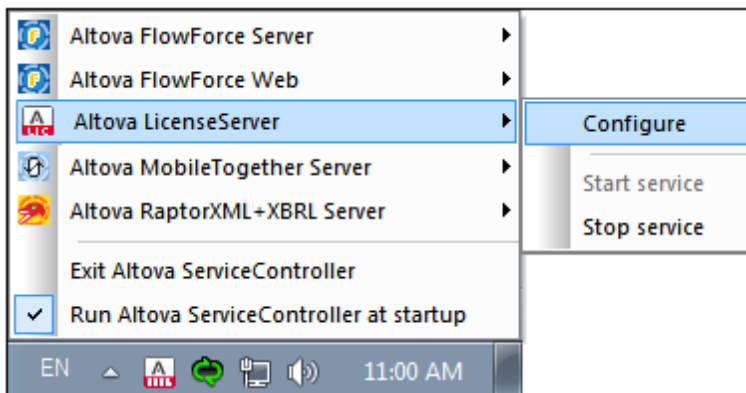
7.6 Altova ServiceController

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**.



Starting and stopping Altova services

Each installed Altova service component will have an entry in the ServiceController menu (*see screenshot above*). An Altova service can be started or stopped via a command in its ServiceController sub-menu. Additionally, important administration tasks of individual services can be accessed via the ServiceController menu. In the screenshot above, for example, Altova LicenseServer service has a sub-menu in which you can choose to access LicenseServer's Configuration page via the **Configure** command.

7.7 How to Assign Licenses

To assign a license to an Altova product via Altova LicenseServer, do the following:

1. [Start LicenseServer](#)
2. Open the [LicenseServer Configuration page](#), which is the administrator's interface with LicenseServer, on [Windows](#), [Linux](#), or [macOS](#).
3. [Upload the license/s](#) you have received from Altova to the license pool of your Altova LicenseServer. Do this in the [License Pool](#) tab of the LicenseServer Configuration page.
4. [Register the Altova product](#) with LicenseServer. Depending on the product's type, the method of registering it with LicenseServer will be different: respectively, via the product's Software Activation dialog, Web UI, or command line. See the documentation of your Altova product for additional information.
5. In the [Client Management](#) tab of the [LicenseServer Configuration page](#), [assign a license](#) to the Altova product.

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.

Note: FlowForce Server Advanced Edition and MapForce Server Advanced Edition will run only on machines with two or more cores.

When assessing the number of cores you should license, this decision should take into account the data volumes you need to process and the processing time your business environment is expected to allow for. In most scenarios, a larger number of cores means more volumes of data processed in less time. Below are just a few application-specific tips:

- FlowForce Server runs as a multi-threaded application. If the number of concurrent requests to the server is big, an insufficient number of cores will lead to latency (waiting times). For example, if you are exposing jobs as Web services, there may be hundreds of concurrent requests from clients. In this case, FlowForce Server will significantly benefit from a larger number of cores.
- MapForce Server will utilize a single core at a time, per mapping. Therefore, if you need to

run multiple mappings simultaneously, a larger number of cores is highly recommended. For example, when MapForce Server runs under FlowForce Server management, several mapping jobs may overlap and run concurrently, depending also on the setup. Note, however, that if the volumes processed by your mappings are extremely big, latency could still occur.

In addition to the above, note that there are various external factors that typically influence the processing volumes and times that your server is capable to handle (for example, the hardware, the current load on the CPU, memory allocation of other applications running on the server). In order to get the most accurate performance measurements, it is generally advisable to first run the tools in your environment and expose them to actual factors and data specific to your business.

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.

7.7.1 Start LicenseServer

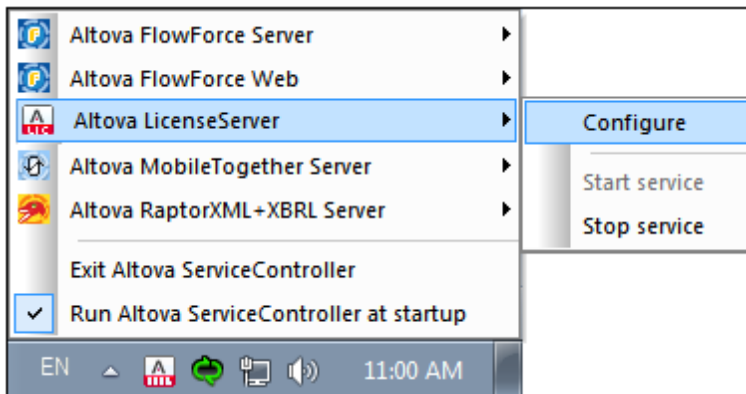
This section:

- How to start LicenseServer on [Windows systems](#)
- How to start LicenseServer on [Linux systems](#)
- How to start LicenseServer on [macOS systems](#)
- Note about [Connection to altova.com](#)

Windows systems

You can start LicenseServer via the Altova ServiceController, which is available in the system tray.

First, click **Start | All Programs | Altova LicenseServer | Altova ServiceController** to start Altova ServiceController and display its icon in the system tray (see screenshot below). If you select the *Run Altova ServiceController at Startup* option, Altova ServiceController will start up on system start and its icon will be available in the system tray from then onwards.



To start LicenseServer, click the Altova 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.

Linux systems

To start LicenseServer as a service on Linux systems, run the following command in a terminal window.

```
[Debian 7]:      sudo /etc/init.d/licenseserver start
[Debian >=8]:   sudo systemctl start licenseserver
[Ubuntu <=14]:  sudo initctl start licenseserver
[Ubuntu >=15]:  sudo systemctl start licenseserver
[CentOS 6]:     sudo initctl start licenseserver
[CentOS >=7]:   sudo systemctl start licenseserver
[RedHat]:       sudo initctl start licenseserver
```

(If you need to stop LicenseServer, replace **start** with **stop** in the above command.)

For more information, see [Installation \(Linux\)](#).

macOS systems

To start LicenseServer as a service on macOS systems, run the following command in a terminal window:

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

If at any time you need to stop LicenseServer, use:

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

For more information, see [Installation \(macOS\)](#).

Connection to the Master Licensing Server at altova.com

Altova LicenseServer needs to be able to communicate with the Master Licensing Server at `altova.com` to validate and authenticate license-related data and to ensure continuous compliance with the Altova license agreements. This communication occurs over HTTPS using port 443. If Altova LicenseServer, after making the initial verification with the `altova.com` Master Licensing Server, is unable to again connect with `altova.com` for a duration of more than 5 days (= 120 hours), then Altova LicenseServer will no longer permit the usage of any Altova software products connected to that Altova LicenseServer.

Any such loss of connection with the `altova.com` master servers will be logged in the [Messages tab](#) of the [Configuration page of Altova LicenseServer](#). In addition, the administrator can configure the Altova LicenseServer to automatically send an alert email when the connection to `altova.com` is lost. Alert Mail settings are available in the [Settings tab](#) of the [Configuration page](#).

7.7.2 Open LicenseServer's Config Page (Windows)

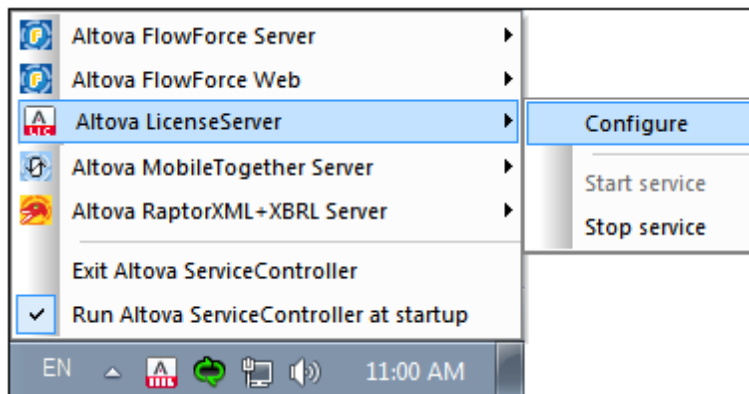
This section:

- [Opening the Configuration page if LicenseServer is on the same machine](#)
- [Opening the Configuration page if LicenseServer is on another machine](#)
- [Logging in with the initial password](#)
- [Setting a fixed port for the Configuration page](#)

Opening the Configuration page if LicenseServer is on the same machine

On Windows systems, if LicenseServer is on the same machine, you can open the [Configuration page](#) of LicenseServer in one of two ways:

- Click **Start | All Programs | Altova LicenseServer | LicenseServer Configuration Page**. The Configuration page opens in a new tab of your Internet browser.
- Click the Altova ServiceController icon in the system tray, mouse over **Altova LicenseServer** in the menu that pops up (see screenshot below), and then select **Configure** from the LicenseServer submenu.



The [Configuration page](#) opens in a new browser window, and its login mask is displayed (screenshot below).

Opening the Configuration page if LicenseServer is on another machine

To open the LicenseServer [Configuration page](#) from some other Windows machine on the local network (than that on which LicenseServer is installed), enter the URL of the LicenseServer [Configuration page](#) in the address bar of a browser and press **Enter**. By default, the URL of the Configuration page will be:

```
http://<serverIPAddressOrName>:8088/
```

The URL is present in the HTML code of the Configuration page itself, which is named `WebUI.html` and is located at:

```
C:/ProgramData/Altova/LicenseServer/WebUI.html
```

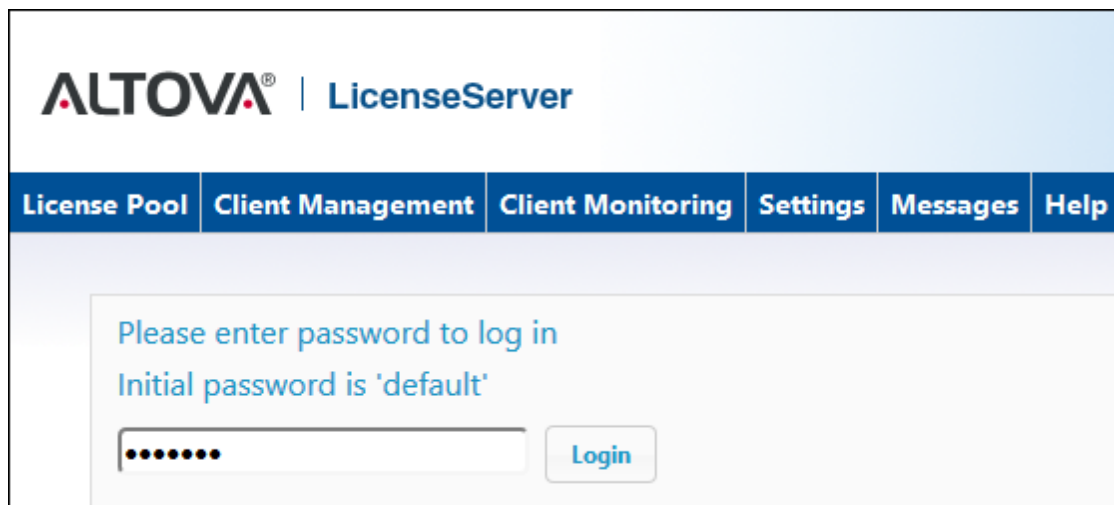
If you have [set the URL of the Configuration page](#) to be generated dynamically (in the Settings tab

of the Configuration page), then a new URL is generated each time LicenseServer is started. You will need to check the current version of `webUI.html` to find out the current URL of the [Configuration page](#).

The dynamically generated URL in `webUI.html` will have a form something like:
`http://127.0.0.1:55541/optionally-an-additional-string`, and it is located in the function `checkIfServiceRunning()` in a script near the end of the `<head>` element. While the port number in the URL is dynamically assigned, the IP address part identifies the server on which LicenseServer has been installed. If you wish to access the LicenseServer [Configuration page](#) from another machine, make sure that the IP address part of the URL has the correct IP address or name of the server on which LicenseServer has been installed. For example, the URL could be something like: `http://SomeServer:55541`.

Logging in with the initial password

After going through the steps above, the [Configuration page](#) is opened with the login mask displayed (*screenshot below*). You can log in with the initial password of `default`. After you have logged in, you can change your password in the [Settings](#) tab.



Setting a fixed or dynamic port for the Configuration page

The port of the Configuration page (Web UI)—and consequently its address—can be specified in the [Settings page](#). By default the port is 8088. You can set any other port you want for the LicenseServer [Configuration page](#) (see *screenshot below*). Alternatively, you allow the port to be selected dynamically each time LicenseServer starts up. In this case, you will need to find out the URL of the Configuration page from the file `webUI.html` (see [Open LicenseServer Config Page \(Windows\)](#), [Open LicenseServer Config Page \(Linux\)](#), and [Open LicenseServer's Config Page \(macOS\)](#)).

Web UI
Changing these settings will cause the LicenseServer to restart and any currently running and licensed applications will be shut down!

Configure the host addresses where the web UI is available to administrators.

☒ All interfaces and assigned IP addresses

☐ Only the following hostname or IP address:
Ensure this hostname or IP address exists or LicenseServer will fail to start!

Configure the port used for the web UI.

☐ Dynamically chosen by the operating system

☒ Fixed port
Ensure this port is available or LicenseServer will fail to start!

The advantage of a fixed port is that the page URL is known in advance and therefore can be accessed easily. If the port is assigned dynamically, the port part of the URL will have to be looked up in the file `WebUI.html` each time LicenseServer is started.

7.7.3 Open LicenseServer's Config Page (Linux)

This section:

- [Opening the Configuration page for the first time with the returned URL](#)
- [URL of the LicenseServer Configuration page](#)
- [Logging in with the initial password](#)
- [Setting a fixed port for the Configuration page](#)

Opening the Configuration page for the first time with the returned URL

On Linux systems, when you register your Altova server product with LicenseServer via the CLI, the URL of the LicenseServer Configuration page is returned. On opening this URL in a browser, you are prompted to read and accept the license agreement. After accepting the license agreement, the Configuration page's login mask is displayed (*screenshot below*).

Note: Altova desktop products are available for Windows only.

URL of the LicenseServer Configuration page

To open the LicenseServer [Configuration page](#) at any time, enter its URL in the address bar of a browser and press **Enter**. By default, the URL of the Configuration page will be:

```
http://<serverIPAddressOrName>:8088/
```

The URL is present in the HTML code of the Configuration page itself, which is named `webUI.html` and is located at:

```
/var/opt/Altova/LicenseServer/webUI.html
```

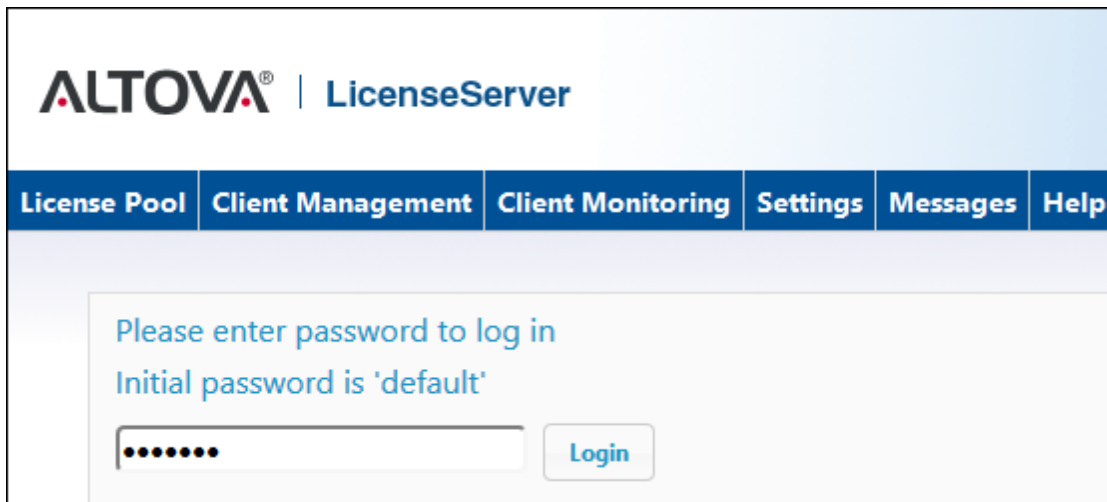
If you have [set the URL of the Configuration page](#) to be generated dynamically (in the Settings tab of the Configuration page), then a new URL is generated each time LicenseServer is started. You will need to check the current version of `webUI.html` to find out the current URL of the [Configuration page](#).

The dynamically generated URL in `webUI.html` will have a form something like:

`http://127.0.0.1:55541`, and it is located in the function `checkIfServiceRunning()` in a script near the end of the `<head>` element. While the port number in the URL is dynamically assigned, the IP address part identifies the server on which LicenseServer has been installed. If you wish to access the LicenseServer [Configuration page](#) from another machine, make sure that the IP address part of the URL has the correct IP address or name of the server on which LicenseServer has been installed. For example, the URL could be something like: `http://MyServer:55541`.

Logging in with the initial password

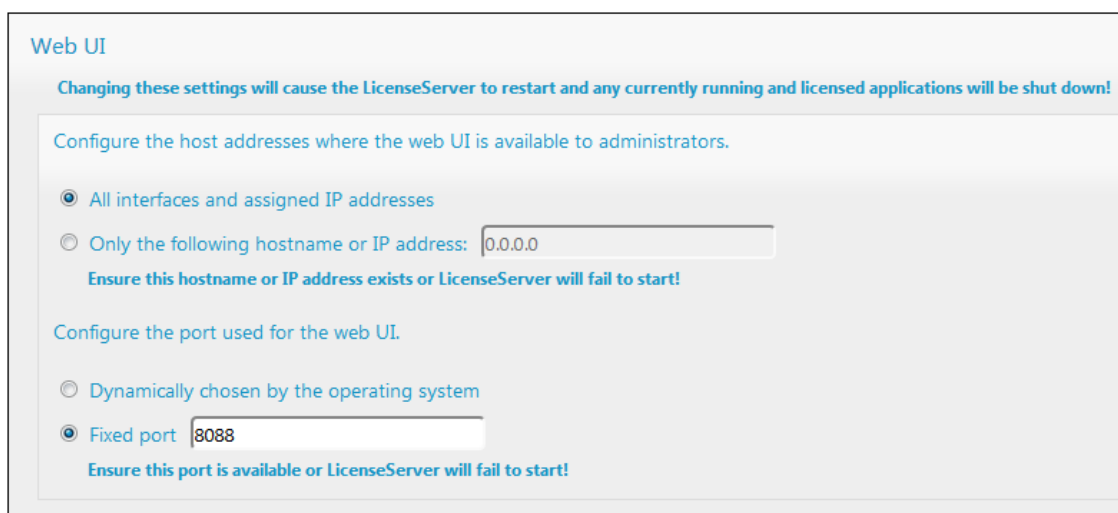
After going through the steps above, the [Configuration page](#) is opened with the login mask displayed (*screenshot below*). You can log in with the initial password of `default`. After you have logged in, you can change your password in the [Settings](#) tab.



The screenshot shows the Altova LicenseServer login interface. At the top, the logo 'ALTOVA® | LicenseServer' is displayed. Below the logo is a navigation bar with six tabs: 'License Pool', 'Client Management', 'Client Monitoring', 'Settings', 'Messages', and 'Help'. The 'Settings' tab is currently selected. The main content area contains a login prompt: 'Please enter password to log in' and 'Initial password is 'default''. Below this text is a password input field with a masked password '.....' and a 'Login' button.

Setting a fixed or dynamic port for the Configuration page

The port of the Configuration page (Web UI)—and consequently its address—can be specified in the [Settings page](#). By default the port is 8088. You can set any other port you want for the LicenseServer [Configuration page](#) (see *screenshot below*). Alternatively, you allow the port to be selected dynamically each time LicenseServer starts up. In this case, you will need to find out the URL of the Configuration page from the file `WebUI.html` (see [Open LicenseServer Config Page \(Windows\)](#), [Open LicenseServer Config Page \(Linux\)](#), and [Open LicenseServer's Config Page \(macOS\)](#)).



The screenshot shows the 'Web UI' configuration page. At the top, it says 'Web UI'. Below this, a warning message states: 'Changing these settings will cause the LicenseServer to restart and any currently running and licensed applications will be shut down!'. The page is divided into two sections. The first section is 'Configure the host addresses where the web UI is available to administrators.' It has two radio button options: 'All interfaces and assigned IP addresses' (which is selected) and 'Only the following hostname or IP address:'. The second section is 'Configure the port used for the web UI.' It also has two radio button options: 'Dynamically chosen by the operating system' and 'Fixed port'. The 'Fixed port' option is selected, and the port number '8088' is entered in the adjacent text field. Both sections include a warning: 'Ensure this hostname or IP address exists or LicenseServer will fail to start!' and 'Ensure this port is available or LicenseServer will fail to start!' respectively.

The advantage of a fixed port is that the page URL is known in advance and therefore can be accessed easily. If the port is assigned dynamically, the port part of the URL will have to be looked up in the file `WebUI.html` each time LicenseServer is started.

7.7.4 Open LicenseServer's Config Page (Mac OS X)

This section:

- [Opening the Configuration page for the first time with the returned URL](#)
- [URL of the LicenseServer Configuration page](#)
- [Logging in with the initial password](#)
- [Setting a fixed port for the Configuration page](#)

Opening the Configuration page for the first time with the returned URL

On macOS systems, when you register your Altova server product with LicenseServer via the CLI, the URL of the LicenseServer Configuration page is returned. On opening this URL in a browser, you are prompted to read and accept the license agreement. After accepting the license agreement, the Configuration page's login mask is displayed (*screenshot below*).

Note: Altova desktop products are available for Windows only.

URL of the LicenseServer Configuration page

To open the LicenseServer [Configuration page](#) at any time, enter its URL in the address bar of a browser and press **Enter**. By default, the URL of the Configuration page will be:

```
http://<serverIPAddressOrName>:8088/
```

The URL is present in the HTML code of the Configuration page itself, which is named `webUI.html` and is located at:

```
/var/Altova/LicenseServer/webUI.html
```

If you have [set the URL of the Configuration page](#) to be generated dynamically (in the Settings tab of the Configuration page), then a new URL is generated each time LicenseServer is started. You will need to check the current version of `webUI.html` to find out the current URL of the [Configuration page](#).

The dynamically generated URL in `webUI.html` will have a form something like:

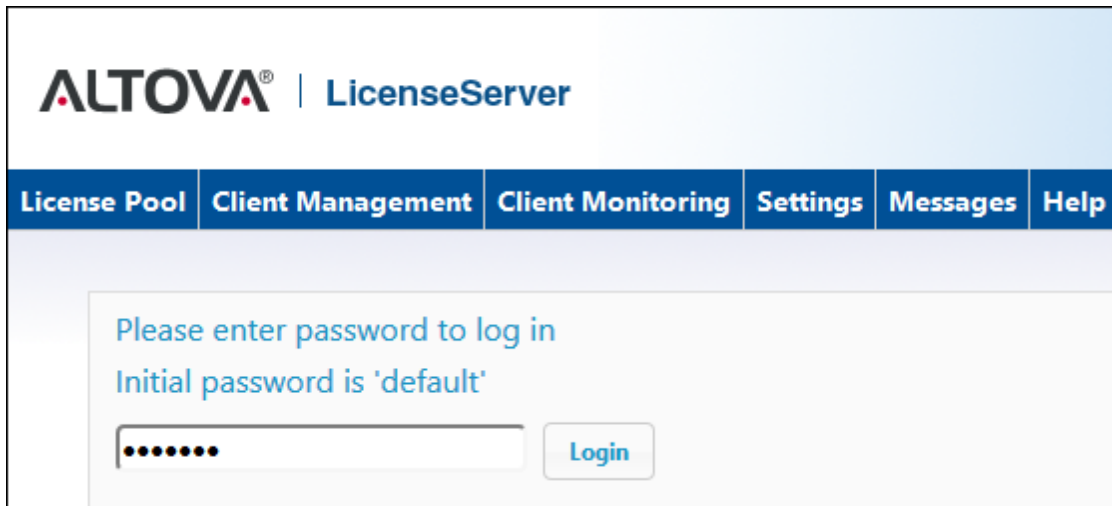
`http://127.0.0.1:55541`, and it is located in the function `checkIfServiceRunning()` in a script near the end of the `<head>` element. While the port number in the URL is dynamically assigned, the IP address part identifies the server on which LicenseServer has been installed. If you wish to access the LicenseServer [Configuration page](#) from another machine, make sure that the IP address part of the URL has the correct IP address or name of the server on which LicenseServer has been installed. For example, the URL could be something like: `http://MyServer:55541`.

Note: The [Configuration page](#) can also be accessed directly via the **Finder | Applications | Altova License Server** icon.

Logging in with the initial password

After going through the steps above, the [Configuration page](#) is opened with the login mask

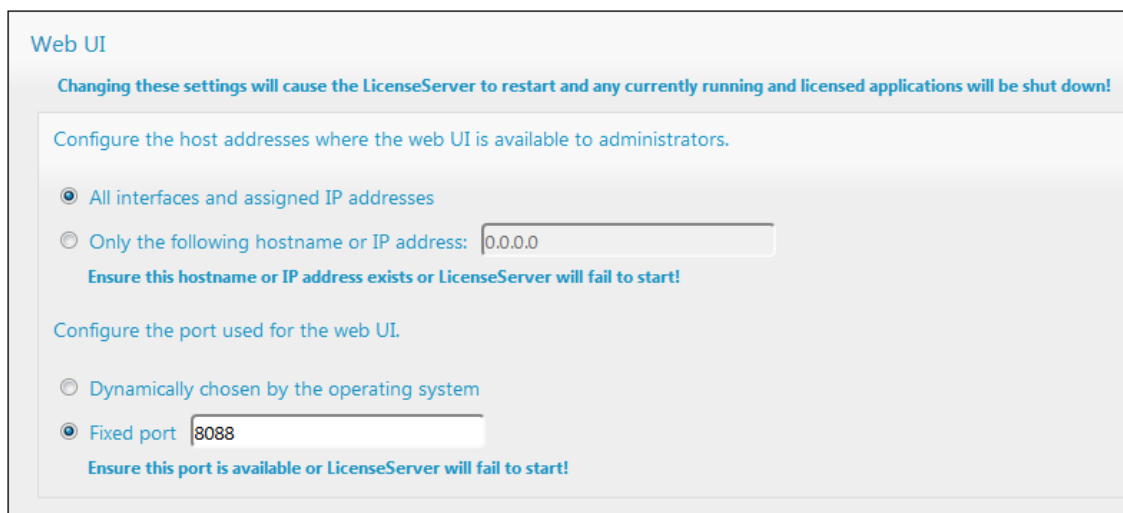
displayed (*screenshot below*). You can log in with the initial password of `default`. After you have logged in, you can change your password in the [Settings](#) tab.



The screenshot shows the Altova LicenseServer login interface. At the top, the logo 'ALTOVA® | LicenseServer' is displayed. Below the logo is a navigation bar with tabs: 'License Pool', 'Client Management', 'Client Monitoring', 'Settings', 'Messages', and 'Help'. The main content area contains a login form with the text 'Please enter password to log in' and 'Initial password is 'default''. There is a password input field with a masked password '.....' and a 'Login' button.

Setting a fixed or dynamic port for the Configuration page

The port of the Configuration page (Web UI)—and consequently its address—can be specified in the [Settings page](#). By default the port is 8088. You can set any other port you want for the LicenseServer [Configuration page](#) (see *screenshot below*). Alternatively, you allow the port to be selected dynamically each time LicenseServer starts up. In this case, you will need to find out the URL of the Configuration page from the file `WebUI.html` (see [Open LicenseServer Config Page \(Windows\)](#), [Open LicenseServer Config Page \(Linux\)](#), and [Open LicenseServer's Config Page \(macOS\)](#)).



The screenshot shows the 'Web UI' configuration settings page. At the top, it says 'Web UI' and 'Changing these settings will cause the LicenseServer to restart and any currently running and licensed applications will be shut down!'. Below this, there are two sections: 'Configure the host addresses where the web UI is available to administrators.' and 'Configure the port used for the web UI.' The first section has two radio buttons: 'All interfaces and assigned IP addresses' (selected) and 'Only the following hostname or IP address: 0.0.0.0'. The second section has two radio buttons: 'Dynamically chosen by the operating system' and 'Fixed port 8088' (selected). Both sections have a warning: 'Ensure this hostname or IP address exists or LicenseServer will fail to start!' and 'Ensure this port is available or LicenseServer will fail to start!'.

The advantage of a fixed port is that the page URL is known in advance and therefore can be accessed easily. If the port is assigned dynamically, the port part of the URL will have to be looked up in the file `WebUI.html` each time LicenseServer is started.

7.7.5 Upload Licenses to LicenseServer

This section:

- [Uploading a license file to the license pool of LicenseServer](#)
- [License status](#)
- [Activating the licenses you wish to use](#)
- [Next steps](#)

Uploading a license file to the license pool of LicenseServer

After you have obtained a license file (file extension **.altova_licenses**) from Altova, save this file to a suitable location and upload this file to Altova LicenseServer. Each license file can contain one or more licenses and depends on your purchase. When you upload a license file (a **.altova_licenses** file), all the licenses in it will be uploaded to the license pool of LicenseServer and can be assigned to an Altova product that has been registered with that LicenseServer. All the uploaded licenses, from one or more license files and for all Altova products, are collected in a license pool on the LicenseServer. The license pool is displayed in the License Pool tab of the LicenseServer Configuration page (*screenshot below*).

ALTOVA® | LicenseServer

License Pool Client Management Client Monitoring Settings Messages(0) Log Out Help

Licenses

<input type="checkbox"/>	Status	Name	Company	Product	Edition	Version	Key Code	Bundle ID	Start Date	End Date	Expires in days	SMP days left	#	License Type	Clients
				All Products ▾	All ▾	All ▾									
<input type="checkbox"/>	Active		Altova GmbH	DatabaseS	Enterprise Editi	2015 rel. 4	GWS36BI-	{D5FC74C	2015-06	-	-	355	50	Installed User	0/50 users
<input type="checkbox"/>	Active	Altova Document	Altova GmbH	FlowForce Ser		2015 rel. 4	9FJUP0P-	-	2015-05	-	-	328	8	CPU Cores	1/50 machin
<input type="checkbox"/>	Active		Altova GmbH	MapForce	Enterprise Editi	2015 rel. 4	BCEB4BI-	{D5FC74C	2015-06	-	-	355	50	Installed User	0/50 users
<input type="checkbox"/>	Active	Altova Document	Altova GmbH	MapForce Ser		2015 rel. 4	23A8TT1-	-	2015-05	-	-	328	8	CPU Cores	1/50 machin
<input checked="" type="checkbox"/>	Active	Altova Document	Altova GmbH	RaptorXML+X		2015 rel. 4	M2L0CMY-	-	2015-05	-	-	328	16	CPU Cores	running assigned
<input type="checkbox"/>	Active	Altova Document	Altova GmbH	RaptorXML Se		2015 rel. 4	847AXW4-	-	2015-05	-	-	328	16	CPU Cores	
<input type="checkbox"/>	Active		Altova GmbH	SchemaAg		2015 rel. 4	GWVBWBI-	{D5FC74C	2015-06	-	-	355	50	Installed User	0/50 users

Activate Deactivate Delete

Upload License File Browse... No files selected. Upload

License files are uploaded to the LicenseServer using the *Upload License File* function of the License Pool tab (see *screenshot above*). Click the **Browse** button and select the license file you want. (The license file to upload is the file you received as an attachment in your License Email from Altova; it has a **.altova_licenses** file extension.) The license file will appear in the *Upload License File* text field and the **Upload** button will be enabled. Click the **Upload** button to upload the license file. All the licenses in the file are uploaded and displayed in the License Pool tab. The screenshot above shows multiple licenses, uploaded from multiple license files.

License status

License status values are as follows:

- *Activating*: When a license is uploaded to the license pool of LicenseServer, the server will transmit license-related data to the `altova.com` master licensing server to validate, authenticate, and activate the license that was supplied. This is necessary to ensure compliance with the Altova license agreements. During this initial activation and authentication transaction—which typically lasts between 30 seconds and a couple of minutes, depending on your Internet connection, speed, and overall network traffic—the status of the license will be indicated as *Activating...*
- *Failed Verification*: If a connection with the `altova.com` master licensing server cannot be made, then the status of the license in the pool will be shown as *Failed Verification*. If this happens, check your Internet connection and firewall rules to ensure that LicenseServer is able to communicate with the `altova.com` master licensing server.
- *Active*: Once the license has been authenticated and activated, the status will change to *Active*.
- *Inactive*: If a license has been verified, but is present on another LicenseServer on the network, the status will be shown as *Inactive*. An *Inactive* status also results when a license is manually deactivated in the license pool by the administrator.
- *Pending*: If the start date of a license is a date in the future, then the license is shown as *Pending*. The status will change to *Active* at 00:00 hrs on the start date. A *Pending* license can be assigned to a product and ensures that the licensing of a product continues uninterrupted before an existing license expires. The changeover to the new license is designed to be a smooth transition, with no running client processes being interrupted.
- *Blocked*: A license is shown in the license pool as *Blocked* if there was a problem authenticating the license and the `altova.com` master licensing server has not granted permission to the LicenseServer to use this license. This could be the result of a license agreement violation, over-usage of a license, or other compliance issues. After fixing the issue, delete, re-upload, and re-activate the license. Please see the table below for additional information.

These statuses are summarized in the table below:

Status	Meaning
<i>Activating...</i>	On upload, license information is sent to <code>altova.com</code> for verification. Refresh the browser to view the updated status. Verification and activation can take a few minutes.
<i>Failed Verification</i>	A connection to <code>altova.com</code> could not be made. After establishing a connection, either restart the service or activate the license (with the Activate button).
<i>Active</i>	Verification was successful, the license is active.
<i>Inactive</i>	Verification was successful, but the license is also on another LicenseServer on the network. Licenses can be made inactive with the Deactivate button.

<i>Pending</i>	A <i>Pending</i> license has a start date in the future, and it turns <i>Active</i> on that start date. It can be assigned to a product and ensures the automatic renewal of a product's licensing before an existing license expires.
<i>Blocked</i>	Verification was not successful. License is invalid and is blocked. After the issue that was causing the license to be blocked has been resolved, delete the license, re-upload it, and re-activate it. Every time a license is uploaded, it contacts the Altova master server, which will start a new verification. If you do not re-upload, then the license verification will have to wait till the next scheduled communication with the Altova master server—and that might not happen for up to a day. Contact Altova Support if the problem persists.

Note: After a license has been sent to `altova.com` for verification, the browser must be refreshed to see the updated status. Verification and activation can take a few minutes.

Note: If a connection to `altova.com` could not be made, the status will be *Failed Verification*. After establishing a connection, either restart the service or try activating the license with the **Activate** button.

Note: When a license is given a status of *Inactive* or *Blocked*, a message explaining the status is also added to the Messages log.

Only an active or pending license can be assigned to a product installation. An inactive license can be activated or deleted from the license pool. If a license is deleted from the license pool, it can be uploaded again to the pool by uploading the license file containing it. When a license file is updated, only those licenses in it that are not already in the pool will be uploaded to the pool. To activate, deactivate, or delete a license, select it and then click the **Activate**, **Deactivate**, or **Delete** button, respectively.

Activate the license/s you wish to use

Before you can assign a license to an Altova product, it must be active. So do ensure it is active. If it is inactive, select it and click **Activate**.

Next Steps

After you have uploaded the license file to the LicenseServer and checked that the license you want is active, do the following:

1. Register the Altova server product ([FlowForce Server](#), [MapForce Server](#), [StyleVision Server](#)) with LicenseServer. (If you have already done this prior to uploading the license file, you can now start assigning licenses.)
2. [Assign a license](#) to your Altova product that has been registered with the LicenseServer.

7.7.6 Register Product/s

Before you can [assign a license](#) to an Altova product, you must register the product installation with LicenseServer. The registration is done from the Altova product, and the process differs according to the type of product.

- *Desktop products:* Registration is via the the product's Software Activation dialog.
- *Server products that have Web UIs:* Registration of FlowForce Server and MobileTogether Server is via the Web UI's *Setup* tab or the product's CLI.
- *Server products that have **no** Web UI:* Registration of MapForceServer, RaptorXML(+XBRL) Server, and StyleVisionServer is via the CLI of these products. You will need the server name or IP Address of the machine on which LicenseServer is installed to carry out the registration.

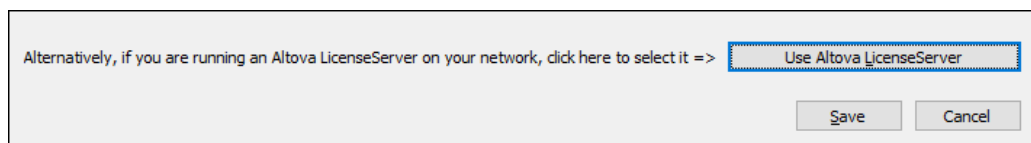
This section describes how to register different Altova products:

- [Register Altova Desktop Products](#)
- [Register FlowForce Server](#)
- [Register MapForce Server](#)
- [Register MobileTogether Server](#)
- [Register RaptorXML\(+XBRL\) Server](#)
- [Register StyleVision Server](#)

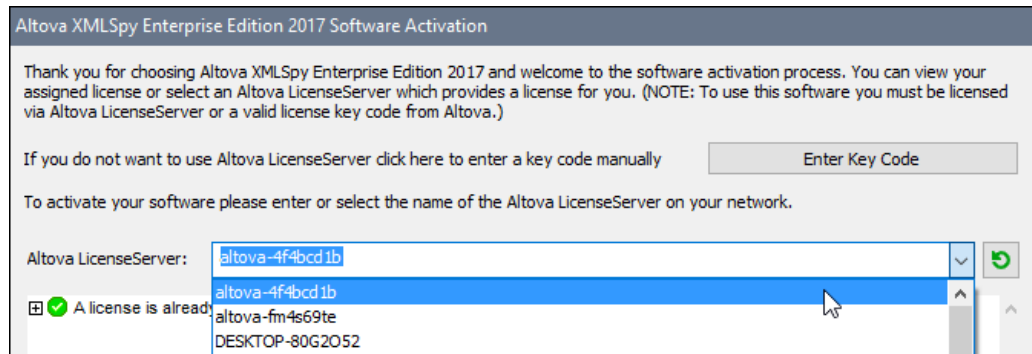
Register Altova Desktop Products

To register an Altova Desktop product with an Altova LicenseServer, do the following:

1. Select the menu command **Help | Software Activation** to go to the product's Software Activation dialog. You can activate your software either (i) via Altova LicenseServer, or (ii) by entering your product's key code details. In this documentation we describe only licensing via Altova LicenseServer.
2. To license your Altova product via LicenseServer, click **Use Altova LicenseServer** (located at the bottom of the dialog; see *screenshot below*)



3. This switches the dialog to LicenseServer activation mode (*screenshot below*). In the *Altova LicenseServer* combo box, select a LicenseServer from the dropdown list.



When the connection to the selected LicenseServer is made, the product is immediately registered with the selected LicenseServer, and, in the [Client Management tab](#), the product is displayed in that client machine's product list.

Unregistering a desktop product

To unregister a desktop product, go to the [Client Management tab](#) of LicenseServer and, in the right-hand *Product licensing* pane, click the product's **Unregister Product** button.

Register FlowForce Server

This section:

- [Methods of registering FlowForce Server with LicenseServer](#)
- [Accessing the FlowForce Server Setup page \(Windows\)](#)
- [Accessing the FlowForce Server Setup page \(Linux\)](#)
- [Registering FlowForce Server via the Setup page](#)
- [Registering FlowForce Server via the FlowForce CLI \(Windows\)](#)
- [Registering FlowForce Server via the FlowForce CLI \(Linux\)](#)
- [Next steps](#)

Methods of registering FlowForce Server

FlowForce Server can be registered with LicenseServer using any of the following methods:

- [Via the FlowForce Server Setup page](#)
- [Via the FlowForce CLI \(Windows\)](#)
- [Via the FlowForce CLI \(Linux\)](#)

Accessing the FlowForce Server Setup page (Windows)

The FlowForce Server Setup page can be accessed in one of the following ways:

- Via the **Start** menu:
Start | Altova FlowForce Server 2018 | FlowForce Server Setup Page
- Via [Altova ServiceController](#): Click the ServiceController icon in the system tray. In the

menu that pops up, select *Altova FlowForce Web | Setup*.

This pops up the FlowForce Server Setup page (*screenshot above*).

Accessing the FlowForce Server Setup page (Linux)

After you have installed FlowForce Server on Linux (see the FlowForce Server user documentation for information about how to do this), start FlowForce Web Server as a service with the following command:

```
sudo /etc/init.d/flowforcewebserver start
```

A message containing the URL of the FlowForce Server Setup appears in the terminal window:

```
FlowForceWeb running on http://127.0.1.1:3459/setup?key=52239315203
```

Enter the URL in the address field of a browser and hit **Enter** to access the FlowForce Server Setup page (*screenshot below*).

Registering FlowForce Server via the Setup page

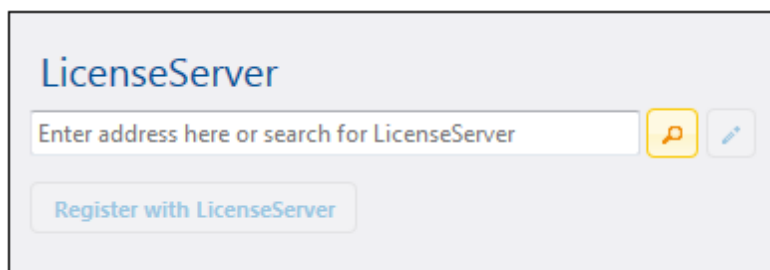
In the Setup page (*screenshot below*)—how to access it is described above—the LicenseServer field specifies the Altova LicenseServer to be used for registration.



The screenshot shows the 'Setup LicenseServer' window of FlowForce SERVER 2014. At the top is the Altova logo and 'FlowForce SERVER 2014'. Below is a navigation bar with 'Home' and 'Help'. The main title is 'Setup LicenseServer'. There is a text input field with the placeholder 'Enter address here or search for LicenseServer' and two icons (magnifying glass and pencil) to its right. Below this is a button labeled 'Register with LicenseServer'. The next section is 'FlowForce Web Server' with 'Bind address' set to 'All interfaces (0.0.0.0)' and 'Port' set to '8082'. Below that is 'Default time zone' set to 'Europe/Berlin'. The next section is 'FlowForce Server' with 'Bind address' set to 'All interfaces (0.0.0.0)' and 'Port' set to '4646'. At the bottom is a button labeled 'Apply settings and restart FlowForce services'.

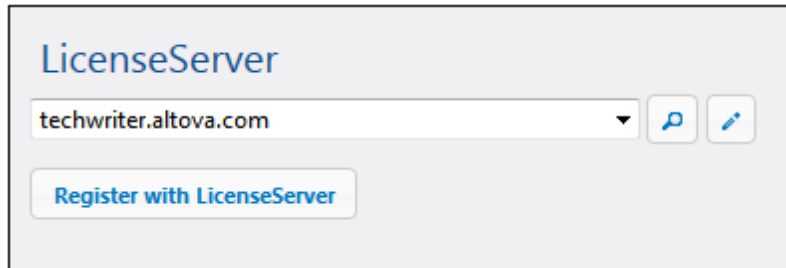
The LicenseServer can be specified in one of two ways.

- You can search for Altova LicenseServers that are currently available on the network—that is, those that are currently running. Do this by clicking the **Search for Altova LicenseServers** button (*highlighted yellow in the screenshot below*).



The screenshot shows a close-up of the 'LicenseServer' section. It features a text input field with the placeholder 'Enter address here or search for LicenseServer'. To the right of the input field is a yellow button with a magnifying glass icon, which is highlighted. To the right of the yellow button is a pencil icon. Below the input field is a button labeled 'Register with LicenseServer'.

The search returns a list of available Altova LicenseServers on the network. One LicenseServer will be selected (*screenshot below*) and the others will be available in the dropdown list of the combo box. Select the LicenseServer on which your FlowForce license is stored.



- Alternatively, you can enter the address of the LicenseServer in the LicenseServer field. If the currently running LicenseServers are available as a dropdown list, you must click the **Manually Enter Address** button to be able to enter an address in the LicenseServer field.

After you have specified the LicenseServer, click **Register with LicenseServer**. The Altova server application will be registered with the specified LicenseServer, and that LicenseServer's [Configuration page](#) will open in a browser with its [Client Management tab](#) active (*screenshot below*).

Note: You may need to allow pop-ups in order for the LicenseServer Configuration page to be displayed.

The screenshot displays the Altova LicenseServer web interface. At the top, the Altova logo and 'LicenseServer' text are visible. Below this is a navigation bar with tabs: License Pool, Server Management, Server Monitoring, Settings, Messages(0), Log Out, and Help. A dropdown menu shows the selected server: DOC.altova.com. The main content area lists three registered servers:

- Altova FlowForce Server 2014**: This server has 2 CPU core(s). Licenses for 2 CPU core(s) are required. ☐ Limit to single thread execution. Max licensed CPU cores: 0.
- Altova StyleVision Server 2014**: This server has 2 CPU core(s). Licenses for 2 CPU core(s) are required. ☐ Limit to single thread execution. Max licensed CPU cores: 0.
- Altova MapForce Server 2014**: This server has 2 CPU core(s). Licenses for 2 CPU core(s) are required. ☐ Limit to single thread execution. Max licensed CPU cores: 0.

At the bottom, there are two buttons: 'Request evaluation licenses' and 'Unregister server and all products'.

In the screenshot below, three Altova products have been registered with the Altova LicenseServer at DOC.altova.com. How to assign licenses is described in the next section, [Assign Licenses to Registered Products](#).

Registering FlowForce Server via the FlowForce CLI (Windows)

On Windows machines, FlowForce Server can also be registered with an Altova LicenseServer on your network via the command line (CLI) by using the `licenseserver` command:

```
FlowForceServer licenseserver Server-Or-IP-Address
```

For example, if LicenseServer is running on `http://localhost:8088`, then register FlowForce Server with:

```
FlowForceServer licenseserver localhost
```

If FlowForce Server was installed with other Altova server products as sub-packages, registering

FlowForce Server will automatically also register the Altova server products. After successfully registering FlowForce Server, you can go to LicenseServer and assign a license to FlowForce Server. How to do this is described in the section [Assign Licenses to Registered Products](#).

Registering FlowForce Server via the FlowForce CLI (Linux)

On Linux machines, FlowForce Server can be registered with LicenseServer by using the `licenseserver` command of the FlowForce Server CLI. Note that FlowForce Server must be started with root rights.

```
sudo /opt/Altova/FlowForceServer2018/bin/flowforceserver 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 FlowForce Server executable is:

```
/opt/Altova/FlowForceServer2018/bin
```

After successfully registering FlowForce Server, you can go to LicenseServer and assign a license to FlowForce Server. How to do this is described in the section [Assign Licenses to Registered Products](#).

Next Steps

After you have registered your Altova product with LicenseServer, do the following:

1. If you have not already uploaded your license file/s to the LicenseServer (see previous section, [Upload the license/s](#)), upload the license file now and check that the license you want is active. If you have already done this, carry on to the next step, [Assign Licenses](#).
2. [Assign a license](#) to your Altova product that has been registered with the LicenseServer.

Register MapForce Server

This section:

- [Registering MapForce Server from FlowForce Server \(Windows\)](#)
- [Registering a standalone MapForce Server \(Windows\)](#)
- [Registering MapForce Server \(Linux\)](#)
- [Next steps](#)

MapForce Server can be installed as part of the FlowForce Server package, or as a standalone server product. In either case, it must be registered with Altova LicenseServer. After it has been registered with LicenseServer, you can [assign a license](#) to it from LicenseServer. On Windows systems, you can install MapForce Server as part of the FlowForce installation. If MapForce Server is installed as part of the FlowForce Server package, it will automatically be registered with LicenseServer when FlowForce is registered with LicenseServer. On Linux systems, the two products must be installed separately. If MapForce Server is installed after FlowForce Server, then

it will be registered automatically when FlowForce Server is registered. But if MapForce Server is installed before FlowForce Server, you will have to register both products separately.

Registering MapForce Server from FlowForce Server (Windows)

MapForce Server is packaged with FlowForce Server, so when FlowForce Server is registered with an Altova LicenseServer on your network, MapForce Server will automatically also be registered with LicenseServer. How to register FlowForce Server is described in the FlowForce Server documentation and in the section, [Register FlowForce Server with LicenseServer](#).

After the registration, you can go to LicenseServer and assign a MapForce Server license to MapForce Server. How to do this is described in the section, [Assign Licenses to Registered Products](#).

Registering a standalone MapForce Server (Windows)

If you have installed MapForce Server as a standalone package, you must register it with an Altova LicenseServer on your network and then license it from the Altova LicenseServer. You can register MapForce Server via its command line interface (CLI) by using the `licenseserver` command:

```
MapForceServer licenseserver Server-Or-IP-Address
```

For example, if LicenseServer is running on `http://localhost:8088`, then register MapForce Server with:

```
MapForceServer licenseserver localhost
```

After successfully registering MapForce Server, you can go to LicenseServer and assign a license to MapForce Server. How to do this is described in the section, [Assign Licenses to Registered Products](#).

Registering MapForce Server (Linux)

On Linux machines, MapForce Server can be registered with LicenseServer by using the `licenseserver` command of the MapForce Server CLI. Note that MapForce Server must be started with root rights.

```
sudo /opt/Altova/MapForceServer2018/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/MapForceServer2018/bin
```

After successfully registering MapForce Server, you can go to LicenseServer and assign a license

to MapForce Server. How to do this is described in the section [Assign Licenses to Registered Products](#).

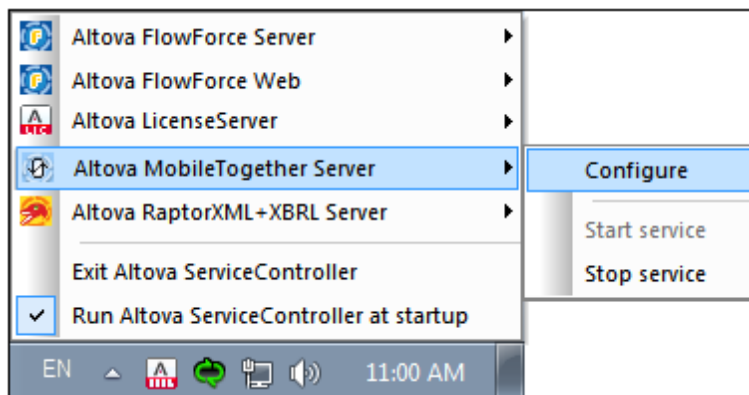
Next Steps

After you have registered your Altova product with LicenseServer, do the following:

1. If you have not already uploaded your license file/s to the LicenseServer (see previous section, [Upload the license/s](#)), upload the license file now and check that the license you want is active. If you have already done this, carry on to the next step, [Assign Licenses](#).
2. [Assign a license](#) to your Altova product that has been registered with the LicenseServer.

Register MobileTogether Server

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



Register MobileTogether Server via:

- The Settings tab of the MobileTogether Server Web UI: (i) Start MobileTogether Server via ServiceController (see *previous point*); (ii) Enter your password to access the Configuration page; (iii) Select the Settings tab; (iv) Go to the LicenseServer pane at the bottom of the page, select the LicenseServer name or address, and click **Register with LicenseServer**.
- its CLI, using the `licenseserver` command:
`MobileTogetherServer licenseserver [options] ServerName-Or-IP-Address`
 For example, if `localhost` is the name of the server on which LicenseServer is installed:
`MobileTogetherServer licenseserver localhost`

After successful registration, go to the [Client Management tab of LicenseServer's configuration page](#) to assign a license to MobileTogether Server.

Register RaptorXML(+XBRL) Server

This section:

- [Registering RaptorXML\(+XBRL\) Server \(Windows\)](#)
- [Registering RaptorXML\(+XBRL\) Server \(Linux\)](#)
- [Next steps](#)

RaptorXML(+XBRL) Server must be installed on the server machine or network to which LicenseServer is connected and then be started as a service. It must then be registered with LicenseServer. Only after registration can a [license be assigned](#) to it from LicenseServer. This section describes how to register RaptorXML(+XBRL) Server with LicenseServer.

Registering RaptorXML(+XBRL) Server (Windows)

You can register RaptorXML(+XBRL) Server via its command line interface (CLI) by using the `licenseserver` command:

```
RaptorXML Server:    RaptorXML licenseserver Server-Or-IP-Address
RaptorXML+XBRL      RaptorXMLXBRL licenseserver Server-Or-IP-Address
Server:
```

For example, if LicenseServer is running on `http://localhost:8088`, then register RaptorXML(+XBRL) Server with:

```
RaptorXML Server:    RaptorXML licenseserver localhost
RaptorXML+XBRL      RaptorXMLXBRL licenseserver localhost
Server:
```

After successfully registering RaptorXML(+XBRL) Server, you can go to LicenseServer and assign a license to RaptorXML(+XBRL) Server. How to do this is described in the section [Assign Licenses to Registered Products](#).

Registering RaptorXML(+XBRL) Server (Linux)

On Linux machines, RaptorXML(+XBRL) Server can be registered with LicenseServer by using the `licenseserver` command of the RaptorXML(+XBRL) Server CLI. Note that RaptorXML(+XBRL) Server must be started with root rights.

```
sudo /opt/Altova/RaptorXMLServer2018/bin/raptorxmlserver licenseserver
localhost
sudo /opt/Altova/RaptorXMLXBRLServer2018/bin/raptorxmlxbmlserver 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 RaptorXML(+XBRL) Server executable is:

```
/opt/Altova/RaptorXMLServer2018/bin  
/opt/Altova/RaptorXMLXBRLServer2018/bin
```

After successfully registering RaptorXML(+XBRL) Server, you can go to LicenseServer and assign a license to RaptorXML(+XBRL) Server. How to do this is described in the section [Assign Licenses to Registered Products](#).

Next Steps

After you have registered your Altova product with LicenseServer, do the following:

1. If you have not already uploaded your license file/s to the LicenseServer (see previous section, [Upload the license/s](#)), upload the license file now and check that the license you want is active. If you have already done this, carry on to the next step, [Assign Licenses](#).
2. [Assign a license](#) to your Altova product that has been registered with the LicenseServer.

Register StyleVision Server

This section:

- [Registering StyleVision Server from FlowForce Server \(Windows\)](#)
- [Registering a standalone StyleVision Server \(Windows\)](#)
- [Registering StyleVision Server \(Linux\)](#)
- [Next steps](#)

StyleVision Server can be installed as part of the FlowForce Server package or as a standalone server product. In either case, it must be registered with Altova LicenseServer. Only after it has been registered with LicenseServer can a [license be assigned](#) to it from LicenseServer. On Windows systems, if StyleVision Server was installed as part of the FlowForce Server package, it will automatically be registered when FlowForce is registered. On Linux systems, only if StyleVision Server is installed after FlowForce Server will it be registered automatically when FlowForce Server is registered subsequently.

Registering StyleVision Server from FlowForce (Windows)

StyleVision Server is packaged with FlowForce Server, so when FlowForce Server is registered with an Altova LicenseServer on your network, StyleVision Server will automatically also be registered with LicenseServer. How to register FlowForce Server is described in the FlowForce Server documentation and in the section, [Register FlowForce Server with LicenseServer](#).

After the registration, you can go to LicenseServer and assign a StyleVision Server license to StyleVision Server. How to do this is described in the section [Assign Licenses to Registered Products](#).

Registering a standalone StyleVision Server (Windows)

If you have installed StyleVision Server as a standalone package on Windows, you must register it with an Altova LicenseServer on your network and then license it from the Altova LicenseServer. You can register StyleVision Server via its command line interface (CLI) by using the `licenseserver` command:

```
StyleVisionServer licenseserver Server-Or-IP-Address
```

For example, if LicenseServer is running on `http://localhost:8088`, then register StyleVision Server with:

```
StyleVisionServer licenseserver localhost
```

After successfully registering StyleVision Server, you can go to LicenseServer and assign a license to StyleVision Server. How to do this is described in the section [Assign Licenses to Registered Products](#).

Registering StyleVision Server (Linux)

On Linux machines, StyleVision Server can be registered with LicenseServer by using the `licenseserver` command of the StyleVision Server CLI. Note that StyleVision Server must be started with root rights.

```
sudo /opt/Altova/StyleVisionServer2018/bin/stylevisionserver 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 StyleVision Server executable is:

```
/opt/Altova/StyleVisionServer2018/bin
```

After successfully registering StyleVision Server, you can go to LicenseServer and assign a license to StyleVision Server. How to do this is described in the section [Assign Licenses to Registered Products](#).

Next Steps

After you have registered your Altova product with LicenseServer, do the following:

1. If you have not already uploaded your license file/s to the LicenseServer (see previous section, [Upload the license/s](#)), upload the license file now and check that the license you want is active. If you have already done this, carry on to the next step, [Assign Licenses](#).
2. [Assign a license](#) to your Altova product that has been registered with the LicenseServer.

7.7.7 Assign Licenses to Registered Products

This section:

- [Before assigning a license](#)
- [The Client Management tab](#)
- [Icons in the Client Management tab](#)
- [Note on cores and licenses](#)
- [Assigning a license](#)
- [Unregistering products from LicenseServer](#)

Before assigning a license

Before you assign a license to an Altova product, make sure that:

- The relevant license has been uploaded to the [license pool of LicenseServer](#) and that the license is active.
- Your Altova product has been registered with LicenseServer.

The Client Management tab

Licenses are assigned in the [Client Management tab](#) of the LicenseServer Configuration page (*screenshot below*). The screenshot shows, in the left pane, that there is one client machine, which has three Altova products that have been registered with LicenseServer.

ALTOVA® | LicenseServer

License Pool **Client Management** **Client Monitoring** **Settings** **Messages(0)** **Log Out** **Help**

Registered Clients

Address	User	Registered Products
doc-aab	adoc	All Products RaptorXML+XBRL Server 2016 rel. 2 MobileTogether Server 2.2 XMLSpy Enterprise Edition 2016 rel. 3

Request evaluation licenses **Unregister client and all products**

RaptorXML+XBRL Server 2016 rel. 2

Key Code	State	CPU Cores
M2L0CMY-W78MPXJ-A8H3C40-W5X55XY-C9C93D1	Active	16
Max licensed CPU cores		16

This server has 6 CPU core(s). Licenses for 6 CPU core(s) are required.

☐ Limit to single thread execution

Unregister Product

Note the following points about the Client Management tab:

- In the left pane, each product is listed under the name of its client machine. In the screenshot above, one client machine is listed. This client machine has three Altova products registered with LicenseServer. If an Altova product on a different client machine is registered with this LicenseServer, then that client machine, with its registered

- products, will also be listed in the left pane.
- When you select a client machine in the left pane, the licensing details of that machine's registered products are displayed in the right-hand pane. Here, you can edit the license assignments of each product.
 - Each registered Altova product on a client machine has its own *Key Code* entry, which takes the key code of a license. A registered product is assigned a license by clicking its **Edit Assigned Licenses** button (see *icon list below*) and selecting the required license from those available for that product in the license pool. This procedure is explained in more detail below.
 - Server products also have a line stating how many CPU cores need to be licensed to run that product on that client. If the number of licensed cores is less than the number required, then the information is marked in red (see *screenshot below*). (The number of CPU cores that need to be licensed is the number of CPU cores on that client and is obtained from the client machine by LicenseServer.)

RaptorXML+XBRL Server 2016 rel. 2

Key Code	State	CPU Cores
Max licensed CPU cores		0

This server has 6 CPU core(s). Licenses for 6 CPU core(s) are required.




☐ Limit to single thread execution

Unregister Product

- If **multiple versions** of a single product (for example, StyleVision Server 2013 and StyleVision Server 2014) have been installed on one machine and if each of these installations has been registered with a single LicenseServer, then the multiple registrations are consolidated in a single registration in the Client Management tab and displayed as a single registration. When a license is assigned to this single registration, all the installations indicated by that registration will be licensed. However, multiple instances of only one installation can be run simultaneously on the client machine. For example, multiple instances of StyleVision Server 2013 or multiple instances of StyleVision Server 2014 can be run simultaneously, but not one instance of StyleVision Server 2013 and one instance of StyleVision Server 2014. Note that newly installed versions must be registered for them to run.
- New versions of Altova server products can only be licensed with the version of LicenseServer that is the latest at the time of the product's release. Older Altova server products will work with newer versions of LicenseServer. So, if you are installing a new version of an Altova server product and if your current LicenseServer version is not the latest, de-install the older version of LicenseServer and install the latest version. All registration and licensing information held in your older version of LicenseServer will be saved, at the time of de-installation, to a database on the client machine, and will be imported automatically into the newer version. (The version number of the LicenseServer that is appropriate for any particular version of a server product is displayed during the installation of that server product. You can choose to install this version along with the

server product. The version of the currently installed LicenseServer is given at the bottom of the [LicenseServer configuration page](#).)

Icons in the Client Management tab

-  *Edit Assigned Licenses.* Available with each product listing. Pops up the [Edit Assigned Licenses dialog](#), in which new licenses can be assigned and already assigned licenses can be edited.
-  *Show Licenses.* Appears with each license listing. Switches to the [License Pool tab](#) and highlights the selected license. Details of the license can be read here.
-  *Unregister Product.* The selected product will be unregistered from LicenseServer.

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.

Note: FlowForce Server Advanced Edition and MapForce Server Advanced Edition will run only on machines with two or more cores.

When assessing the number of cores you should license, this decision should take into account the data volumes you need to process and the processing time your business environment is expected to allow for. In most scenarios, a larger number of cores means more volumes of data processed in less time. Below are just a few application-specific tips:

- FlowForce Server runs as a multi-threaded application. If the number of concurrent requests to the server is big, an insufficient number of cores will lead to latency (waiting times). For example, if you are exposing jobs as Web services, there may be hundreds of concurrent requests from clients. In this case, FlowForce Server will significantly benefit from a larger number of cores.

- MapForce Server will utilize a single core at a time, per mapping. Therefore, if you need to run multiple mappings simultaneously, a larger number of cores is highly recommended. For example, when MapForce Server runs under FlowForce Server management, several mapping jobs may overlap and run concurrently, depending also on the setup. Note, however, that if the volumes processed by your mappings are extremely big, latency could still occur.

In addition to the above, note that there are various external factors that typically influence the processing volumes and times that your server is capable to handle (for example, the hardware, the current load on the CPU, memory allocation of other applications running on the server). In order to get the most accurate performance measurements, it is generally advisable to first run the tools in your environment and expose them to actual factors and data specific to your business.

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 Server, 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.

Assigning a license

To assign a license to a registered product, click the **Edit Assigned Licenses** button of that product. This pops up the Manage Licenses dialog (*screenshot below*).

Edit Assigned Licenses

Product: RaptorXML+XBRL Server 2016 rel. 2

Registered at: doc-aab

Requires licenses for 6 CPU cores Max licensed CPU cores: **16**

Available licenses						
	#	License Type	State	Key Code	Expires in days	SMP days left
<input checked="" type="checkbox"/>	16	CPU Cores	Active	M2L0CMY-W78MP		336

Apply Changes **Go to License Pool**


Note the following points about the licenses displayed in the Manage Licenses dialog:


- The product to be licensed is listed at the top left of the dialog. In the screenshot above

- the product is Altova RaptorXML+XBRL Server.
- The machine on which the server is installed (doc-aab in the screenshot above) is listed next.
 - The dialog displays all the currently active licenses for that product that are in the license pool. In our screenshot, there is one currently active RaptorXML+XBRL Server license in the license pool. (LicenseServer will automatically detect from each license in the pool the product for which it has been issued.)
 - The license type can be either *Cores* (all Altova server products, including MobileTogether Server) or *Users* (MobileTogether Server only). The license type is shown in the *License Type* column. The license in the screenshot above has been licensed for 16 CPU cores.
 - You need to know the number of processor cores on the server on which the Altova server product has been installed. If the machine has a dual-core processor, you need a two-core (the CPU Cores count) license. How many cores the registered server product requires is listed below the machine name. The license you assign to this server must be valid for a sufficient number of cores. Note that you can combine licenses to reach the required number of cores. So, if the machine's processor is octa-core (eight-core), you can combine two 4-core licenses.
 - The Edit Assigned Licenses dialog will list only currently active licenses for **that** product. Licenses for other Altova products will not be listed.
 - Licenses that have been assigned already—for example, to another installation of the product on the network—will have their check boxes checked. So only unchecked licenses may be selected.
 - The *CPU Cores* (or # for MobileTogether Server users) column indicates for how many CPU cores (or MobileTogether Clients) a license is valid.
 - If you wish to make modifications to the license pool—for example, to upload, activate, deactivate, or delete a license—click the **Go to License Pool** button.

Select the license you wish to assign. The license's check box will be checked. Also, the total number of CPU cores licensed for that product on that client is listed near the top left of the dialog as *Max licensed CPU cores* (see screenshot above). You can select more licenses if you wish to increase the number of licensed CPU cores for that product on that client. The *Max licensed CPU cores* in this case will be the sum of the CPU cores of all the selected licenses.


After selecting the license/s, click **Apply Changes**. The license/s will be assigned to that product and displayed in the Client Management tab (see screenshot below). The screenshot shows that a 16-CPU-core license for Altova RaptorXML+XBRL has been assigned to the client machine.

 **RaptorXML+XBRL Server 2016 rel. 2**

Key Code	State	CPU Cores	
M2L0CMY-W78MPXJ-A8H3C40-W5X55XY-C9C93D1	Active	16	
Max licensed CPU cores		16	

This server has 6 CPU core(s). Licenses for 6 CPU core(s) are required.

☐ Limit to single thread execution

 **Unregister Product**

Unregistering products

Each Altova product registered with LicenseServer is listed in the right-hand (*Product licensing*) pane under its client machine name and has an **Unregister Product** button at the bottom of its entry (see *screenshot above*). Click this button to unregister the product from LicenseServer. If a license was assigned to the product, the assignment will be terminated when the product is unregistered. To unregister all products, click the **Unregister client and all products** button at the top of the right-hand (*Product licensing*) pane (see *first screenshot in this section*).

To re-register a product with LicenseServer, do the following:

- *Server products:* Go to the Settings page in the server product's Web UI. If the server product has no Web UI, open a command prompt window and use the product's CLI to register it. The steps for each server product. are described here: [Register FlowForce Server](#), [Register MapForce Server](#), [Register MobileTogether Server](#), [Register StyleVision Server](#), and [Register RaptorXML\(+XBRL\) Server](#).
- *Desktop products:* Open the product's [Software Activation dialog \(Help | Software Activation\)](#), switch to Activation via LicenseServer mode, and, in the *Altova LicenseServer* field, select the LicenseServer you want to register the product with. The product will be registered, and it will appear in LicenseServer's Client Management tab, in that client's registered-product list.

7.8 Configuration Page Reference

The LicenseServer Configuration page is the administrator's interface with LicenseServer (Web UI). It allows the management of LicenseServer and the licensing of Altova products that have been registered with LicenseServer. The LicenseServer Configuration page is viewed in a web browser. How to open the Configuration page is described in the sections, [Open LicenseServer Config Page \(Windows\)](#), [Open LicenseServer Config Page \(Linux\)](#) and [Open LicenseServer's Config Page \(macOS\)](#).

This section is a user reference for the Configuration page and is organized by the tabs of the Configuration page:

- [License Pool](#)
- [Client Management](#)
- [Client Monitoring](#)
- [Settings](#)
- [Messages, Log Out](#)

For a step-by-step guide of how to assign licenses with LicenseServer, see the section [How to Assign Licenses](#).

7.8.1 License Pool

This section:

- [Uploading a license](#)
- [License status](#)
- [Activating, de-activating, and deleting a license](#)
- [Icons in the License Pool tab](#)
- [License information](#)
- [Note on desktop product licenses](#)
- [Note on cores and licenses](#)

The **License Pool** tab displays information about the licenses that are currently available on LicenseServer (see *screenshot below*). When a license file is uploaded to LicenseServer with the **Upload** button on this page, all the licenses contained in that license file are placed in the license pool on LicenseServer. The License Pool page therefore provides an overview of all Altova product licenses that are currently available on LicenseServer, together with the details of each of these licenses. On this page, in addition to being able to upload licenses to the license pool, you can activate, deactivate, and delete selected licenses.

ALTOVA® | LicenseServer

License Pool | Client Management | Client Monitoring | Settings | Messages(0) | Log Out | Help

Licenses

<input type="checkbox"/>	Status	Name	Company	Product	Edition	Version	Key Code	Bundle ID	Start Date	End Date	Expires in days	SMP days left	#	License Type	Clients
				All Products ▾	All ▾	All ▾									
<input type="checkbox"/>	Active		Altova GmbH ⓘ	DatabaseS ⓘ	Enterprise Editi	2015 rel. 4	GWS36BI-	{D5FC74C	2015-06	-	-	355	50	Installed User	0/50 users
<input type="checkbox"/>	Active	Altova Document	Altova GmbH ⓘ	FlowForce Ser		2015 rel. 4	9FUUP0P-	-	2015-05	-	-	328	8	CPU Cores	1/50 machin ⓘ
<input type="checkbox"/>	Active		Altova GmbH ⓘ	MapForce ⓘ	Enterprise Editi	2015 rel. 4	BCEB4BI-	{D5FC74C	2015-06	-	-	355	50	Installed User	0/50 users
<input type="checkbox"/>	Active	Altova Document	Altova GmbH ⓘ	MapForce Ser		2015 rel. 4	23A8TT1-	-	2015-05	-	-	328	8	CPU Cores	1/50 machin ⓘ
<input checked="" type="checkbox"/>	Active	Altova Document	Altova GmbH ⓘ	RaptorXML+X		2015 rel. 4	M2L0CMY-	-	2015-05	-	-	328	16	CPU Cores	running assigned ⓘ
<input type="checkbox"/>	Active	Altova Document	Altova GmbH ⓘ	RaptorXML Se		2015 rel. 4	847AXW4-	-	2015-05	-	-	328	16	CPU Cores	
<input type="checkbox"/>	Active		Altova GmbH ⓘ	SchemaAg ⓘ		2015 rel. 4	GWVBWBI-	{D5FC74C	2015-06	-	-	355	50	Installed User	0/50 users

Activate

Deactivate

Delete

Upload License File

Browse...

No files selected.

Upload

Uploading a license

To upload a license file (a `.altova_licenses` file which you receive from Altova GmbH for your Altova product), click the **Browse** button, browse for the license file, and select it. On clicking **Upload**, all the licenses contained in the license file are placed in the license pool and displayed on the License Pool page (*screenshot above*).

License status

License status values are as follows:

- *Activating*: When a license is uploaded to the license pool of LicenseServer, the server will transmit license-related data to the `altova.com` master licensing server to validate, authenticate, and activate the license that was supplied. This is necessary to ensure compliance with the Altova license agreements. During this initial activation and authentication transaction—which typically lasts between 30 seconds and a couple of minutes, depending on your Internet connection, speed, and overall network traffic—the status of the license will be indicated as *Activating...*
- *Failed Verification*: If a connection with the `altova.com` master licensing server cannot be made, then the status of the license in the pool will be shown as *Failed Verification*. If this happens, check your Internet connection and firewall rules to ensure that LicenseServer is able to communicate with the `altova.com` master licensing server.
- *Active*: Once the license has been authenticated and activated, the status will change to *Active*.
- *Inactive*: If a license has been verified, but is present on another LicenseServer on the network, the status will be shown as *Inactive*. An *Inactive* status also results when a license is manually deactivated in the license pool by the administrator.
- *Pending*: If the start date of a license is a date in the future, then the license is shown as *Pending*. The status will change to *Active* at 00:00 hrs on the start date. A *Pending* license can be assigned to a product and ensures that the licensing of a product continues uninterrupted before an existing license expires. The changeover to the new license is designed to be a smooth transition, with no running client processes being interrupted.
- *Blocked*: A license is shown in the license pool as *Blocked* if there was a problem authenticating the license and the `altova.com` master licensing server has not granted permission to the LicenseServer to use this license. This could be the result of a license agreement violation, over-usage of a license, or other compliance issues. After fixing the issue, delete, re-upload, and re-activate the license. Please see the table below for additional information.

These statuses are summarized in the table below:

Status	Meaning
<i>Activating...</i>	On upload, license information is sent to <code>altova.com</code> for verification. Refresh the browser to view the updated status. Verification and activation can take a few minutes.
<i>Failed Verification</i>	A connection to <code>altova.com</code> could not be made. After establishing a connection, either restart the service or activate the license (with the Activate button).
<i>Active</i>	Verification was successful, the license is active.

<i>Inactive</i>	Verification was successful, but the license is also on another LicenseServer on the network. Licenses can be made inactive with the Deactivate button.
<i>Pending</i>	A <i>Pending</i> license has a start date in the future, and it turns <i>Active</i> on that start date. It can be assigned to a product and ensures the automatic renewal of a product's licensing before an existing license expires.
<i>Blocked</i>	Verification was not successful. License is invalid and is blocked. After the issue that was causing the license to be blocked has been resolved, delete the license, re-upload it, and re-activate it. Every time a license is uploaded, it contacts the Altova master server, which will start a new verification. If you do not re-upload, then the license verification will have to wait till the next scheduled communication with the Altova master server—and that might not happen for up to a day. Contact Altova Support if the problem persists.

Note: After a license has been sent to [altova.com](#) for verification, the browser must be refreshed to see the updated status. Verification and activation can take a few minutes.

Note: If a connection to [altova.com](#) could not be made, the status will be *Failed Verification*. After establishing a connection, either restart the service or try activating the license with the **Activate** button.

Note: When a license is given a status of *Inactive* or *Blocked*, a message explaining the status is also added to the Messages log.

Only an active or pending license can be assigned to a product installation. An inactive license can be activated or deleted from the license pool. If a license is deleted from the license pool, it can be uploaded again to the pool by uploading the license file containing it. When a license file is updated, only those licenses in it that are not already in the pool will be uploaded to the pool. To activate, deactivate, or delete a license, select it and then click the **Activate**, **Deactivate**, or **Delete** button, respectively.

Connection to the Master Licensing Server at [altova.com](#)





Altova LicenseServer needs to be able to communicate with the Master Licensing Server at [altova.com](#) to validate and authenticate license-related data and to ensure continuous compliance with the Altova license agreements. This communication occurs over HTTPS using port 443. If Altova LicenseServer, after making the initial verification with the [altova.com](#) Master Licensing Server, is unable to again connect with [altova.com](#) for a duration of more than 5 days (= 120 hours), then Altova LicenseServer will no longer permit the usage of any Altova software products connected to that Altova LicenseServer.

Any such loss of connection with the [altova.com](#) master servers will be logged in the [Messages tab](#) of the [Configuration page of Altova LicenseServer](#). In addition, the administrator can configure the Altova LicenseServer to automatically send an alert email when the connection to [altova.com](#) is lost. Alert Mail settings are available in the [Settings tab](#) of the [Configuration page](#).

Activating, deactivating, and deleting a license

An active license can be deactivated by selecting the license and clicking **Deactivate**. An inactive license can be activated (**Activate** button) or deleted (**Delete** button). When a license is deleted it is removed from the license pool. A deleted license can be added again to the license pool by uploading the license file containing it. If a license file is re-uploaded, only licenses that are not already in the license pool will be added to the license pool; licenses that are already in the pool will not be re-added.

Icons in the License Pool tab

-  *Altova MissionKit logo.* Appears next to individual Altova desktop product names when the desktop product license is part of a MissionKit license. See [Note on desktop product licenses](#) below.
-  *Show Assigned Client.* Appears in the *Clients* column of a license that has been assigned. Goes to the [Client Management](#) tab, in which the licenses of a client's registered products can be managed.
-  *Show Running Client.* Appears in the *Clients* column of a license that has been assigned to software that is currently running. Goes to the [Client Monitoring](#) tab of the client machine running that software. Here, information about the selected client and its registered software is displayed.
-  *Show Info.* Appears in the *Clients* column of a license that has not been assigned. Displays information about the license, such as the user count and whether the license is part of a license bundle.

License information

The following license information is displayed:

- *Status:* Can be one of the following values: *Activating* | *Failed Verification* | *Active* | *Inactive* | *Blocked*. See [License status](#) above.
- *Name, Company:* The name and company of the licensee. This information was submitted at the time of purchase.
- *Product, Edition, Version:* The version and edition of the licensed products. At the top of each column is a combo box to filter licenses by category.
- *Key Code, Bundle ID:* The license key to unlock the product. All products in a single Altova MissionKit bundle have the same Bundle ID. Non-bundled products have no Bundle ID.
- *Start Date, End Date:* Together they give the validity period of the license. Permanent licenses have no end date.
- *Expires in days, SMP (days left):* The number of days left before the license expires. Each licensed purchase comes with a Support & Maintenance Package, which is valid for a certain number of days. The *SMP* column notes how many SMP days are still left.
- *#, License Type:* The number of users or CPU cores that are allowed for the license is listed in the *#* column. Whether it is users or cores that the license regulates is given in the *License Type* column. In the case of Altova's MobileTogether Server product, licenses are also assigned on the basis of the number of MobileTogether client **devices** that connect to MobileTogether Server. In the case of all other Altova server products, licenses are assigned on the basis of **CPU cores** only (see *note below*). In the case of Altova

desktop products, licenses are assigned on the basis of **users**. See [Note on desktop product licenses](#) below.

- **Clients:** This column has entries only for [MobileTogether Server \(MTS\) licenses](#) and [desktop product licenses](#). It does not have entries for non-MTS [server product licenses](#). For [MobileTogether Server device licenses](#), this column displays whether the license is assigned. For Desktop products, the column displays the machine count and user count as described below.

Desktop products: machine count and user count

- The *machine count* indicates the number of machines that are currently licensed to run the software with a given license. For example, 7/10 *machines* means that the license can be used for software instances on 10 machines, and is currently being used for software instances on 7 machines. Click the [Show Assigned Client](#) button to go to the [Client Management](#) tab and see the details of the client machine's license/s.
- The *user count* indicates the number of users that are currently using the license out of the total number of allowed users. Only licensed software installations that are currently running are counted. For example, 3/10 *users* means that the license is currently being used by 3 users out of a total of 10 allowed users. If a licensed software installation is currently running, you can click the [Show Running Client](#) button to open the [Client Monitoring tab](#) and see details of Altova products that are running on client machines on the network.
- The *user count* and *machine count* together tell you the current licensing capacity and usage of a given license. For example, if the machine count of an [installed user license](#) is 7/10 and if the user count is 3/10, then we know the following: (i) The product software is allowed to be licensed on 10 machines; (ii) The software has been licensed on 7 machines; (iii) 3 of the 7 licensed software installations are currently running.

Unassigning a license

To unassign a license from a software installation on a machine, go to the [Client Management](#) tab. Select the machine and then the software to be unassigned. Then click the **Edit Assigned Licenses** button, deselect the license, and click **Apply Changes**.

Note on desktop product licenses

There are three types of desktop user licenses:

- **Installed User:** Licenses are purchased for an exact number of computers on which you would like to install the software. For example, if you purchase a *10-Installed-User* license, you may install and use the software on up to 10 computers. On each licensed machine you may start as many simultaneous software instances as you wish. The license, for each "installed user" allows the software product to be used on that machine.
- **Concurrent User:** This license allows you to install the software on up to that number of computers that is equal to 10 times the allowed number of concurrent users. All the installations must be on the same physical network. At any given time, the software can be used by only the allowed number of concurrent users. For example, if you purchase a *20-Concurrent-User* license, then the software may be installed on up to 200 computers on the same physical network and used on up to 20 computers at any given time. If you would like to use *Concurrent User* licenses on different physical networks, you must purchase a separate license for each network. Note that you cannot use one *Concurrent User* license to span multiple networks.

- *Named User:* For each *Named-User* license you may install the software on up to 5 computers, but only the user named in the license may use the software. With this license, a user can work on different computers provided that **only one instance** of the software is used at any given time.

Note about Altova MissionKit licenses

[Altova MissionKit](#) is a suite of Altova desktop products. An Altova MissionKit license comprises individual licenses for each of the desktop products in the MissionKit suite. Each of these individual product licenses has a different and unique key code, but the same MissionKit Bundle ID. If you upload an Altova MissionKit license to the license pool, then the individual licenses of each product comprising MissionKit is listed in the License Pool (with the [Altova MissionKit logo](#) next to it). If you assign one of these product licenses to a particular user, then all the other products of that MissionKit bundle are also assigned to that user. As a result, no other product in that particular MissionKit bundle can be assigned to another user.

License check-outs

You can check out a license from the license pool for a period of up to 30 days so that the license is stored on the product machine. (The exact number of days is specified in the License Checkout setting of the [Settings](#) tab.) This enables you to work offline, which is useful, for example, if you wish to work in an environment where there is no access to your Altova LicenseServer (such as when your Altova product is installed on a laptop and you are traveling). While the license is checked out, LicenseServer displays the license as being in use, and the license cannot be used by any other machine. The license automatically reverts to the checked-in state when the check-out period ends. Alternatively, a checked-out license can be checked in at any time via the **Check in** button of the Software Activation dialog. To check out a license from the license pool to your product machine, go to the **Help** menu of your Altova desktop product and select **Software Activation**. Refer to the user manual of your Altova product for more more information.

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.

Note: FlowForce Server Advanced Edition and MapForce Server Advanced Edition will run only

on machines with two or more cores.

When assessing the number of cores you should license, this decision should take into account the data volumes you need to process and the processing time your business environment is expected to allow for. In most scenarios, a larger number of cores means more volumes of data processed in less time. Below are just a few application-specific tips:

- FlowForce Server runs as a multi-threaded application. If the number of concurrent requests to the server is big, an insufficient number of cores will lead to latency (waiting times). For example, if you are exposing jobs as Web services, there may be hundreds of concurrent requests from clients. In this case, FlowForce Server will significantly benefit from a larger number of cores.
- MapForce Server will utilize a single core at a time, per mapping. Therefore, if you need to run multiple mappings simultaneously, a larger number of cores is highly recommended. For example, when MapForce Server runs under FlowForce Server management, several mapping jobs may overlap and run concurrently, depending also on the setup. Note, however, that if the volumes processed by your mappings are extremely big, latency could still occur.

In addition to the above, note that there are various external factors that typically influence the processing volumes and times that your server is capable to handle (for example, the hardware, the current load on the CPU, memory allocation of other applications running on the server). In order to get the most accurate performance measurements, it is generally advisable to first run the tools in your environment and expose them to actual factors and data specific to your business.

MobileTogether Server licenses

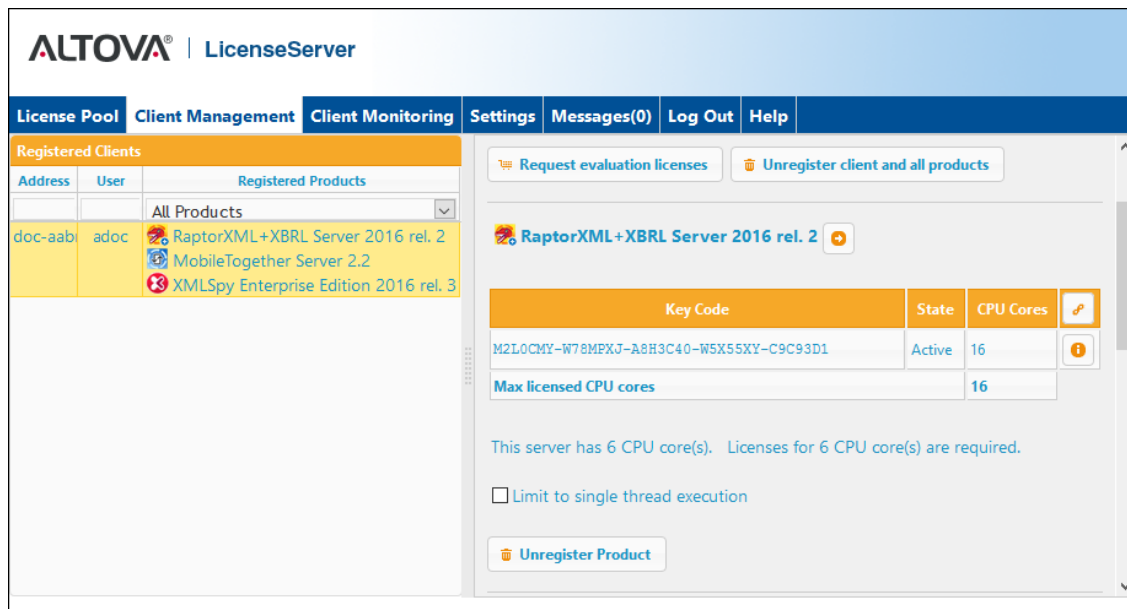
MobileTogether Server licenses are based on the number of CPU cores on the MobileTogether Server machine. Core licenses allow an unlimited number of MobileTogether Client devices to connect to the server. However, if you check the *Limit to single thread execution* check box, then only one mobile device 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 Server, 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.

7.8.2 Client Management

This section:

- [Icons in the Client Management tab](#)
- [Managing licenses in the Product Licensing pane](#)
- [Assigning licenses](#)
- [Single thread execution](#)
- [One client machine under different names](#)
- [Requesting an evaluation license](#)
- [Unregistering products](#)

The **Client Management** tab (screenshot below), is divided into two panes:



- *Registered clients:* The left-side pane displays a tabular listing of machines on the network that have at least one Altova product which is [registered with LicenseServer](#). Such machines are called *registered clients*. Each registered client is listed in the left pane with all its registered products. How to register products with LicenseServer is described in [Register Product/s](#). The display in this pane can be filtered by selecting or entering a filter at the top of one of the pane's columns.
- *Product licensing:* This is the right-side pane. When a registered client is selected in the left (*Registered Clients*) pane, then the licensing details of that client's registered products are displayed in the right pane. Here you can manage the licensing of each registered product (*described below*).

Icons in the Client Management tab



Edit Assigned Licenses. Available with each product listing. Pops up the [Edit Assigned](#)

[Licenses dialog](#), in which new licenses can be assigned and already assigned licenses can be edited.



Show Licenses. Appears with each license listing. Switches to the [License Pool tab](#) and highlights the selected license. Details of the license can be read here.



Unregister Product. The selected product (on the selected client machine) will be unregistered from LicenseServer. See [Unregistering products](#) below. To unregister a client and all its products, click **Unregister client and all products** at the top of the pane.

Managing licenses in the Product Licensing pane

In the right-side *Product licensing* pane, you can do the following:

- *Assign, unassign, change a product's license:* Click the product's **Edit Assigned Licenses** button to do this. See [Assigning licenses](#) below. Note that each server product has a line stating how many CPU cores need to be licensed to run that product on that client. If the number of licensed cores is less than the number required, then the information is marked in red. (The number of CPU cores that need to be licensed is the number of CPU cores on that client and is obtained from the client machine by LicenseServer.)
- *Set up a single-core, server-product license to use only one core of a client:* See [Single thread execution](#) below.
- *Unregister a product from LicenseServer:* Use a product's **Unregister Product** button for this. See [Unregistering products](#) below.

Assigning licenses

To assign a license to a registered product, click the **Edit Assigned Licenses** button of that product. This displays the Edit Assigned Licenses dialog (*screenshot below*).

Edit Assigned Licenses

Product:

RaptorXML+XBRL Server 2016 rel. 2

Registered at:

doc-aab

Requires licenses for 6 CPU cores Max licensed CPU cores: 16

Available licenses

	#	License Type	State	Key Code	Expires in days	SMP days left	
<input checked="" type="checkbox"/>	16	CPU Cores	Active	M2L0CMY-W78MP		336	

Apply Changes

Go to License Pool

Select the license/s you wish to assign, then click **Apply Changes**. The license/s will be assigned to that product and displayed in the *Product licensing* tab of the Client Management tab (see screenshot below).

RaptorXML+XBRL Server 2016 rel. 2

Key Code	State	CPU Cores	
M2L0CMY-W78MPXJ-A8H3C40-W5X55XY-C9C93D1	Active	16	
Max licensed CPU cores		16	

This server has 6 CPU core(s). Licenses for 6 CPU core(s) are required.

☐ Limit to single thread execution

Unregister Product

Single thread execution

If an Altova server-product license for only one core is available in the license pool, a machine with multiple cores can be assigned this one-core license. In such a case, the machine will run that product on a single core. Processing will therefore be slower, because multi-threading (which is possible on multiple cores) will not be available. The product will be

executed in single thread mode on that machine.

To assign a single-core license to a multiple-core machine, select the *Limit to single thread execution* check box for that product.

In the case of **MobileTogether Server (MTS)**, if single-thread execution is selected for an MTS core license, then only one mobile device will be able to connect to the MobileTogether Server at any time. Note that, if, in this case, a second device connects to MobileTogether Server, 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.

One client machine under different names

If a client machine is registered more than once with LicenseServer, it might appear in the Client Management tab under multiple names, that is, with multiple entries. This could happen, for example, if a machine is re-registered with the host name given in a different form.

To ensure that additional licenses are not redundantly assigned to the same machine under its different names, you should unregister redundant client machine entries by clicking the **Unregister client and all products** button at the top of the right-hand (*Product licensing*) pane. Also, if the same license is assigned multiple times to the same machine under its different names, licensing conflicts could arise. So, to avoid these two situations (redundant licensing and multiple assignments of a single license), it is recommended that redundant entries of a single client machine be unregistered.

Given below are forms a machine name might take in the Client Management tab:

- *Host name with domain name (the fully qualified domain name, FQDN), such as: "win80-x64_1.my.domain.com" or "Doc3.my.domain.com".* This happens when the host name of the machine (with or without the domain information) is passed as the argument of the `licenseserver` CLI command that is used to register the server product with LicenseServer. For example: `<AltovaServerProduct> licenseserver Doc3`. This produces an FQDN such as: `Doc3.my.domain.com`.

An FQDN is also produced when `localhost` is supplied on Windows 7 and 10 systems as the host name.

- *Host name without domain name. For example: "win80-x64_1" or "Doc3".* This happens on Windows 8 systems when `localhost` is given as the machine name.
- *localhost.* In some cases, `localhost` is also displayed as a machine name.

Note: If, during installation of an Altova server product on Windows machines, the machine is automatically registered with LicenseServer, `localhost` is used by the installer as the machine name.

Connecting to LicenseServer over VPN

If a client machine connects to your network over a Virtual Private Network (VPN) service, the

client machine might be assigned an IP address dynamically, leading to it being identified as a different machine each time it connects. How to resolve the issue that results is described in the topic [Network Information](#).

Requesting an evaluation license

You can obtain a 30-day free evaluation license for each of a client's installed Altova products that have been registered with LicenseServer. Click the **Request Evaluation Licenses** button at the top of the right-hand (*Product licensing*) pane. A dialog appears that contains a list of the Altova products (on that client machine) which have been registered with LicenseServer. Make sure that the products for which you want an evaluation license are checked, then fill in the registration fields, and send the request. You will receive an e-mail from Altova containing the 30-day evaluation license/s. For server products, the number of cores for which the license is valid will be exactly the number required by the product at the time the request is sent. Save the license/s to disk and [upload them to the license pool](#).

Unregistering products

Each Altova product registered with LicenseServer is listed in the right-hand (*Product licensing*) pane under its client machine name and has an **Unregister Product** button at the bottom of its entry. Click this button to unregister the product from LicenseServer. If a license was assigned to the product, the assignment will be terminated when the product is unregistered. To unregister all products, click the **Unregister client and all products** button at the top of the right-hand (*Product licensing*) pane (*see first screenshot in this section*).

To re-register a product with LicenseServer, do the following:

- *Server products:* Go to the Settings page in the server product's Web UI. If the server product has no Web UI, open a command prompt window and use the product's CLI to register it. The steps for each server product. are described here: [Register FlowForce Server](#), [Register MapForce Server](#), [Register MobileTogether Server](#), [Register StyleVision Server](#), and [Register RaptorXML\(+XBRL\) Server](#).
- *Desktop products:* Open the product's [Software Activation dialog \(Help | Software Activation\)](#), switch to Activation via LicenseServer mode, and, in the *Altova LicenseServer* field, select the LicenseServer you want to register the product with. The product will be registered, and it will appear in LicenseServer's Client Management tab, in that client's registered-product list.

For more information, see the section, [Assigning Licenses to Registered Products](#).

7.8.3 Client Monitoring

The **Client Monitoring** tab provides an overview of the selected client machine. The tab displays the following:







Checked-out Clients

End users of an Altova desktop product (not server product), such as XMLSpy or MapForce, can check out a license that is registered with LicenseServer. This would typically be done in situations where it is foreseen that the end user machine will be offline for a certain period of time. The license can be checked out from LicenseServer for the period during which the machine will be offline. For this period, the end user can continue using the Altova desktop product without making contact with LicenseServer. The licenses and users that are currently checked out, together with the check-out period, are listed under this heading.

Note: License check-out is carried out by the end user via the Software Activation dialog (**Help | Software Activation**) of the Altova desktop product.

Running clients

A list of the Altova products that are currently running on that client. If multiple instances of a product are running then each of these instances is listed.

Running Clients								
Product	Edition	Version	User	Address	State	Failover	Last seen (seconds ago)	
 RaptorXML+XBRL Serv		2016 rel. 2	DOBRA	doc-aab	Running		8	 
 XMLSpy	Enterprise Editio	2016 rel. 3	adoc	doc-aab	Running		11	 

Note: [Failover LicenseServers](#) work with client applications that are v2015rel3 or later (in the case of Altova MobileTogether Server, version 1.5 or later); older clients are flagged.

Note: If a license for a desktop product is checked out, then it will be displayed in a separate *Checked-out Clients* section of the Client Monitoring tab. The license will be indicated as running, and will not be able for use to any other machine on the network.

Icons in the Client Monitoring tab



Show License(s). Available for each product instance. Goes to the [License Pool](#) tab, and highlights the license of the selected product instance.



Manage Client. Available for each product instance. Goes to the [Client Management](#) tab, and highlights the client of the selected product instance.

7.8.4 Settings

This section:

- [Failover LicenseServer settings](#)
- [Network settings](#)
- [Alert Mail settings](#)
- [Miscellaneous settings](#)

The **Settings** tab is described below. You can set the following:

- A **time period to wait till LicenseServer shuts down**. A shutdown would be implemented typically for server maintenance. The time to shutdown can be used to save work on clients running Altova desktop products. The shutdown time you select is the maximum time to shutdown. LicenseServer will shut down earlier, as soon as LicenseServer is no longer connected to any client running a desktop product. The countdown to shutdown starts when you click **Shutdown**. To cancel shutdown, click **Abort Shutdown**. To enable clients to run during a LicenseServer shutdown, configure a [Failover LicenseServer](#).
- A second LicenseServer can be configured to take over from the Primary LicenseServer if the Primary LicenseServer becomes unavailable. This second LicenseServer is called the [Failover LicenseServer](#). How to specify its settings are described [here](#).
- The password for logging in to LicenseServer. Enter the desired password and click **Change Password**.
- Test connectivity to Altova by clicking **Test Connection to Altova**. Note that you must save new settings (by clicking the **Save** button at the bottom of the pane) before testing the connection. The **Test Connection to Altova** button is disabled while the test is in progress, and becomes enabled again when the test has been completed.
- Network settings for the web-based configuration page (Web UI), for the proxy server (if any) that is used to connect to the Internet, and for License Service availability. These settings are described in [Network settings](#) below.
- Email server settings and the alert mail recipient to contact in the event of a significant LicenseServer occurrence. These settings are described in [Alert Mail settings](#) below.
- After you change a setting, click **Save** at the bottom of the pane. Note that a changed setting will not take effect till it is saved.

Failover LicenseServer settings

A second LicenseServer can be configured to take over from the Primary LicenseServer if the Primary LicenseServer becomes unavailable. This second LicenseServer is called the **Failover LicenseServer**.

Failover LicenseServer Settings

To reduce the risk of an unavailable LicenseServer you can configure a second LicenseServer as a backup or "Failover LicenseServer".

In the event that the Primary LicenseServer becomes unavailable a Failover LicenseServer can take over.

LicenseServer Mode

- ☐ Primary LicenseServer
- ☒ Failover LicenseServer

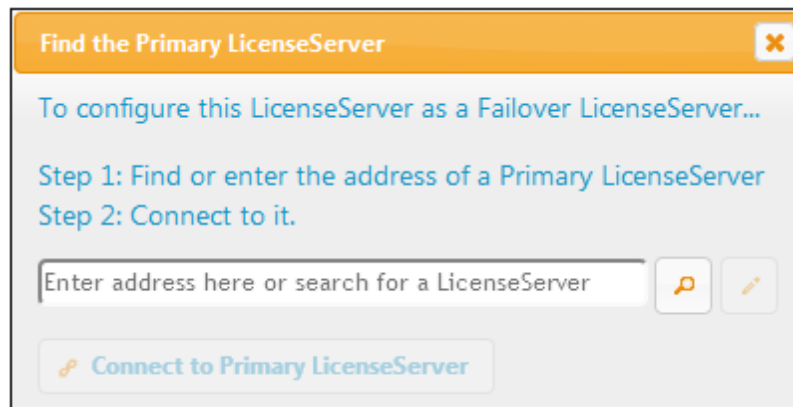
Please note: The Failover LicenseServer periodically synchronizes all licenses, registered clients and license assignments from the Primary LicenseServer. Whenever a Failover LicenseServer takes over from a Primary LicenseServer any changes to these items made on the Failover LicenseServer during this period will be lost as soon as the Primary LicenseServer regains control. Other settings such as Proxy Server and Mail settings are independently set in each server and are not synchronized.

This is a Failover LicenseServer for the LicenseServer at **kubu6.altova.com**

Last seen 2/5/2015, 11:56:04 AM

To set up a LicenseServer as the Failover LicenseServer, do the following:

1. Install LicenseServer as described in the Installation section.
2. Set the LicenseServer's mode to *Failover LicenseServer* by selecting the corresponding radio button (see *screenshot above*). (By default the LicenseServer Mode is set to *Primary LicenseServer*.)
3. In the Find Primary LicenseServer dialog that appears (*screenshot below*), enter the Primary LicenseServer you want to back up with this Failover LicenseServer. You can do this in one of two ways: (i) Click **Search for LicenseServers** and then select, in the combo box, the LicenseServer you want to backup from the list of found LicenseServers; (ii) Click **Manually Enter Address**, and enter the address of the LicenseServer you want to backup. After entering the Primary LicenseServer, click **Connect to Primary LicenseServer**.



4. A confirmation dialog appears, asking you to confirm that you wish to set the current LicenseServer as the Failover LicenseServer (of the Primary LicenseServer you have just selected). Click **Yes** if you wish to confirm. Note that going ahead with the confirmation will remove any installed licenses and registered clients on the current LicenseServer.

Once a Failover LicenseServer has been configured, both the Primary LicenseServer and Failover LicenseServer will have notifications about their respective modes at the top of the Configuration Page. In the two screenshots below, the Failover LicenseServer is shown first, then the Primary LicenseServer.



Note the following points:

- After a Failover LicenseServer has been configured, it periodically synchronizes all licenses, registered clients, and license agreements from the Primary. If the Primary becomes unavailable, then the Failover takes over the role of LicenseServer. Once the Primary becomes available again, the Primary retakes control from the Failover. Any license-related modifications made on the Failover in this period will be lost when the Primary regains control.
- The Failover LicenseServer will provide licenses only to clients that are of version 2015 rel 3 or later (in the case of Altova MobileTogether Server, version 1.5 or later). Older clients are flagged as such in the [Client Monitoring tab](#) of the Primary LicenseServer (*screenshot below*). We recommend that you upgrade your client applications to version 2015 rel 3 or later if you wish to use the Failover LicenseServer feature (in the case of Altova MobileTogether Server, version 1.5 or later).

License checkout

You can select whether to allow desktop licenses to be checked out from the license pool to the machine on which the Altova desktop product is installed. If you allow this, then a client that has

acquired a license from your LicenseServer can check out this license and remain unmonitored for any period up to the maximum period you specify in this setting. The maximum allowed period for check outs is 30 days. (This enables a client to continue using the desktop product even when access to LicenseServer is not possible (say, when traveling) or not wanted.) The client can do the check out via the **Help | Software Activation** command of the Altova desktop application.

After a license has been checked out, it is moved to the *Checked-out Clients* section of the [Client Monitoring](#) tab. It will be considered to be in use, and it will not be available for use by any other client on the network.

Network settings

Administrators can specify network access points to the LicenseServer configuration page and to LicenseServer.

The screenshot displays the LicenseServer configuration interface, organized into three main sections: Web UI, Proxy Server, and License Service. Each section contains configuration options and a warning about restarts.

- Web UI:** A warning states that changing settings will cause a restart. It allows configuring host addresses (radio buttons for 'All interfaces' or 'Only the following hostname or IP address' with a text field '0.0.0.0') and the port (radio buttons for 'Dynamically chosen' or 'Fixed port' with a text field '8088').
- Proxy Server:** A warning states that changing settings will cause a restart. It includes fields for Hostname ('myproxy'), Port Number ('1285'), User Name ('myusername'), and Password ('•••••').
- License Service:** A warning states that changing settings will cause a restart. It allows configuring host addresses (radio buttons for 'All interfaces', 'Local only (localhost)', or 'Only the following hostnames or IP addresses' with a text field).

- *Web UI:* Allowed IP addresses can vary from: all interfaces and IP addresses on that

machine to a fixed address. Ports can be either dynamically calculated or fixed. This allows a wide range of allowed `IP-Address:Port` settings. The default port setting is **8088**.

- *Proxy Server (available from v1.3 onwards)*: If a proxy server is being used to connect to the Internet, the details of the proxy server must be entered in the Proxy Server pane (see *screenshot above*). Otherwise, the proxy server settings can be left blank. To configure LicenseServer to use a proxy server, enter the proxy server's host name, and, if required, a port number. If the proxy server does not need authentication, then the *User Name* and *Password* fields can be left blank.
- *License Service*: The machine on which License Server is installed can be connected to multiple networks via one or more network interfaces. On each network, the License Server machine is known by a host name and an IP address. The *License Service* setting lets you configure on which networks the license service is available. The `localhost` option allows the service on the local machine only. If you list hostnames and/or IP addresses, use a comma-separated list without any spaces (for example: `hostname1,IPAddress1,hostname2`). The port number of the service is fixed at **35355**.

The default settings allow unrestricted access to LicenseServer and its configuration page from within the networks to which LicenseServer is connected. If you wish to restrict access to either LicenseServer or its configuration page, enter the appropriate settings and click **Save**.

Run a connectivity test (see *above*) to check that the settings are correct.

Alert Mail settings

Alert mails can be sent to designated email addresses when significant LicenseServer events occur. One example of a significant event is the following. Altova LicenseServer needs to be connected to the `altova.com` server. If the connection is broken for more than 24*5 hours (5 days), LicenseServer will not allow licenses. As a result, work sessions with Altova products licensed by LicenseServer could be disrupted. In order to alert the administrator that a connection is broken, an alert mail can be sent to an email address.

The Alert Mail pane (see *screenshot below*) is where you enter settings for sending alert mails to an administrator's email address.

Alert Mail

Configure email settings for communication with administrator.

SMTP Host

SMTP Port

User authentication

User password

From

To

Miscellaneous

☐ Show hint how to receive evaluation licenses for a server product

☒ Send a warning email if contact with a running product is lost.

SMTP Host and *SMTP Port* are the access details of the email server from which the email alert will be sent. *User Authentication* and *User Password* are the user's credentials for accessing the email server. The *From* field takes the address of the email account from which the email will be sent. The *To* field takes the recipient's email address.

Click **Save** when done. After saving the Alert Mail settings, email alerts will be sent to the address specified whenever a significant event occurs, such as when connection to `altova.com` is lost. Note that such events are also recorded in the [Messages tab](#), and can be looked up there.

Miscellaneous settings

Show hints for receiving and deploying evaluation licenses

Checking this box (see screenshot above) displays, at the top of the configuration page, brief instructions about how to evaluate and deploy evaluation licenses.

Send a warning email if contact with a running product is lost

A warning message is sent from the *From* address to the *To* address if a connection with a product that is licensed and running is lost.

7.8.5 Messages, Log Out

The **Messages** tab displays all messages relevant to licenses in the license pool of the LicenseServer. Each message has a **Delete** button that allows you to delete that particular message.

The **Log Out** tab serves as the Log Out button. Clicking the tab logs you out immediately and then displays the Login mask.

7.9 Password Reset

If you forget your LicenseServer password, you can use the `passwordreset` command from the CLI to reset the password to `default`.

1. Open a command line window.
2. Change to the directory where the LicenseServer application package or executable is installed.
3. Enter the command: `licenseserver passwordreset`
This resets the LicenseServer administrator password to `default`.
4. You can now log in to the Administrator Web UI with the password `default`.

Chapter 8

Digital Certificate Management

8 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 OS X / macOS 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 OS X / macOS, you can do the equivalent operation in Safari (see [Trusting Server Certificates on OS X / macOS](#)). 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

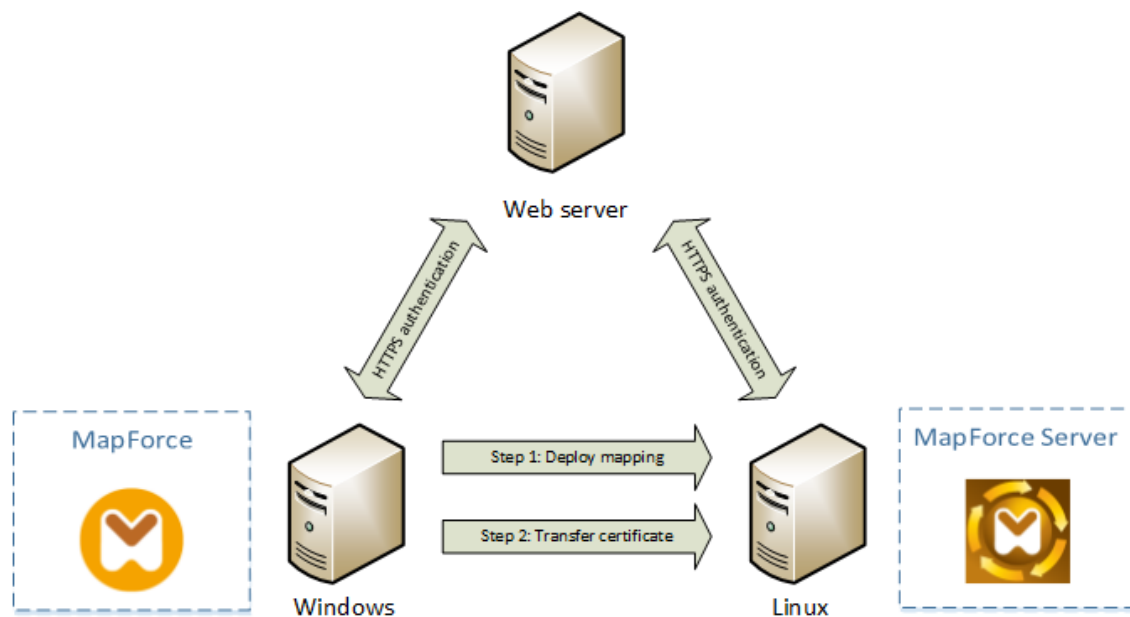
Web server may be configured to handle HTTPS and client certificates in one of the following ways:

- 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	<p>On Windows, you can manage certificates using the Certificate snap-in (see Accessing the Certificate Stores on Windows).</p> <p>TLS support is available through the <i>Secure Channel</i> (also known as <i>SChannel</i>) library.</p>
Linux	<p>On Linux, you can manage certificates using the OpenSSL (<code>openssl</code>) 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.</p> <p>TLS support is available through the OpenSSL library (https://www.openssl.org/).</p>
OS X / macOS	<p>On OS X / macOS, you can manage certificates using the <i>Keychain Access Manager</i>, located under Finder > Applications > Utilities.</p> <p>TLS support is provided by the <i>Secure Transport</i> library native to the operating system.</p>

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 OS X / macOS platforms, see [Client Certificates on Linux](#) and [Client Certificates on OS X / macOS](#), respectively.

8.1 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 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:

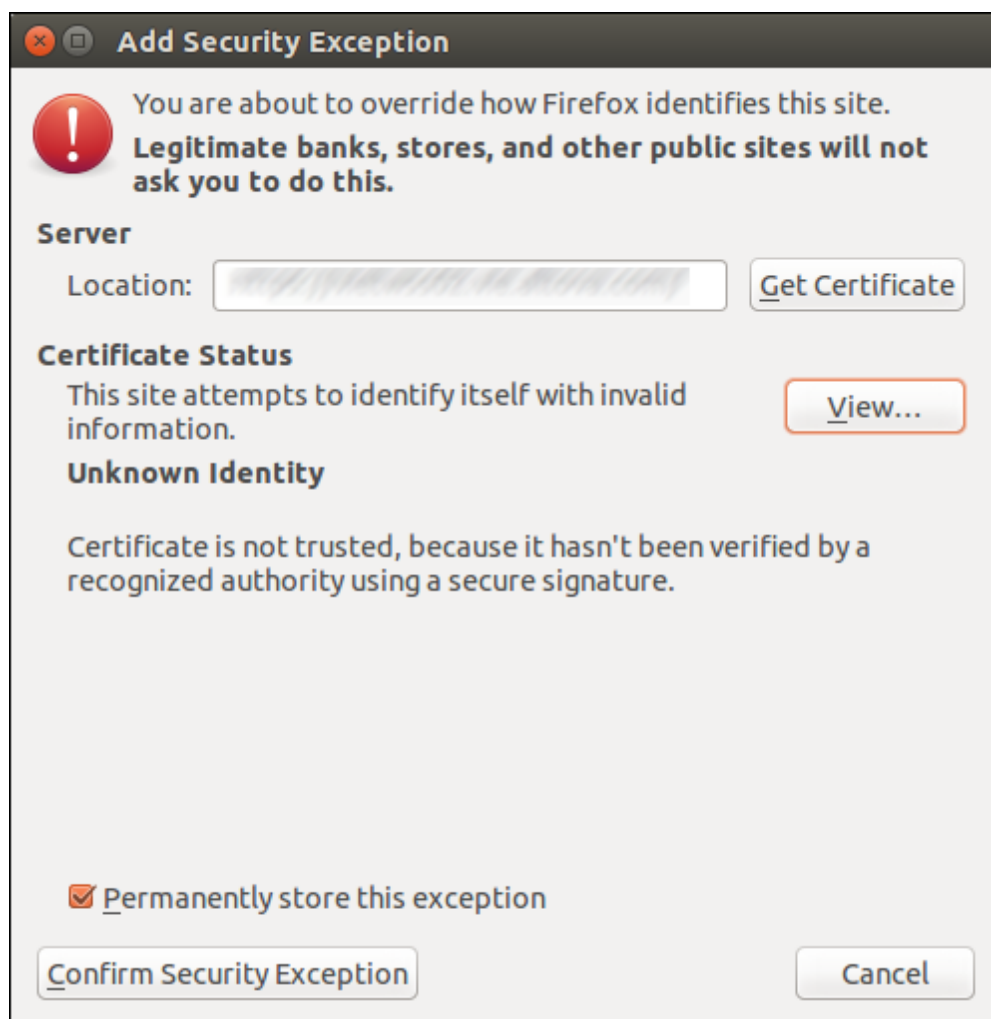
```
cp server_cert.crt /etc/pki/ca-trust/source/anchors/
```

4. Use the command:

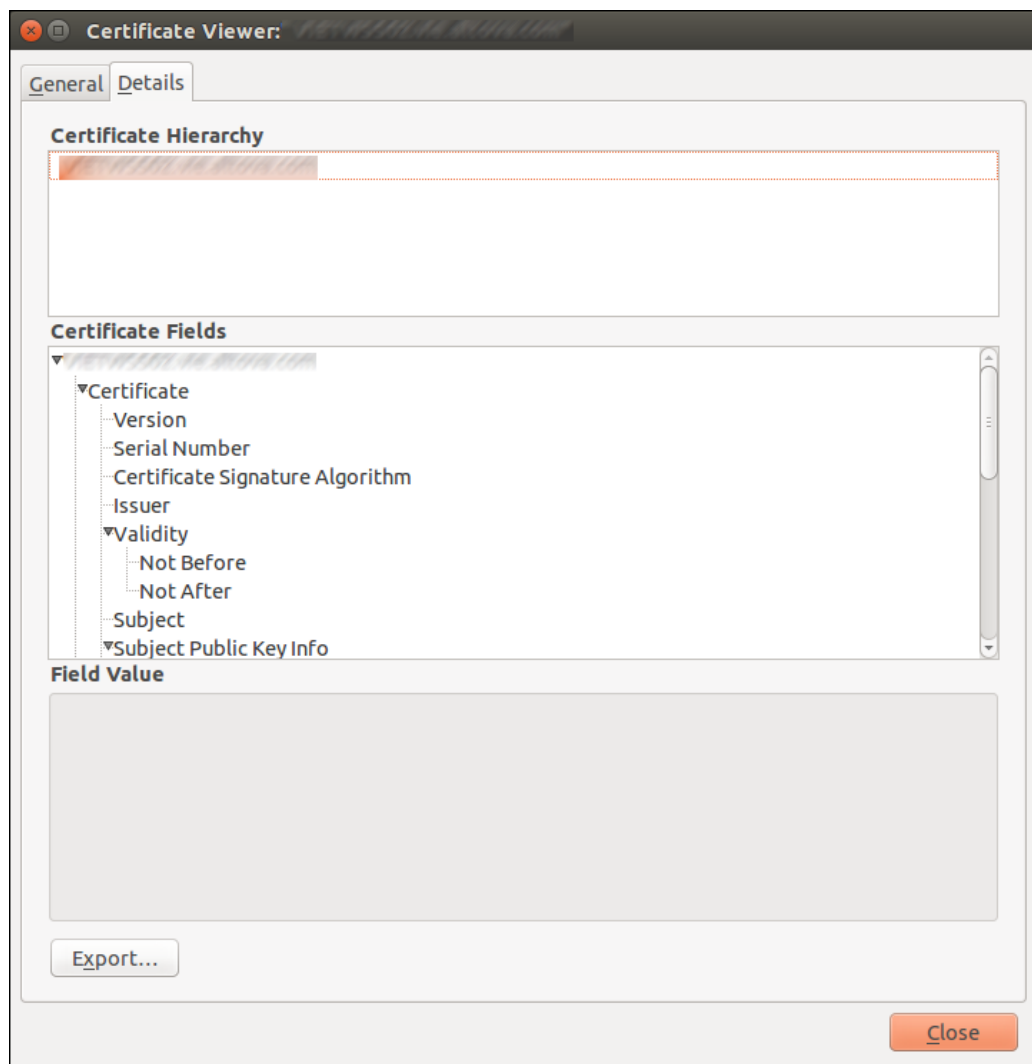
```
update-ca-trust extract
```

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.



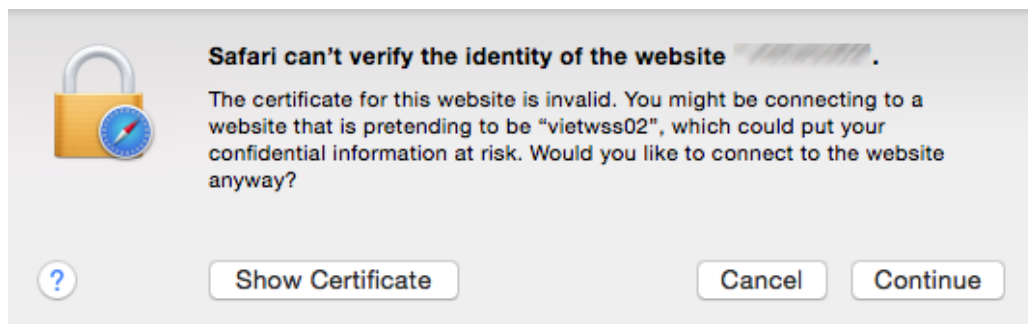
5. Click **Export** and save the certificate file to a local directory.

8.2 Trusting Server Certificates on OS X / macOS

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.

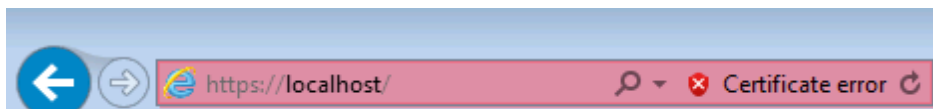
8.3 Trusting Server Certificates on Windows

To establish trust with a Web server, obtain the certificate file of the Web server, and import it into the system's certificate store, see also [https://technet.microsoft.com/en-us/library/cc754489\(v=ws.11\).aspx](https://technet.microsoft.com/en-us/library/cc754489(v=ws.11).aspx). One of the ways to obtain the server certificate is through the browser, as shown in the example below.

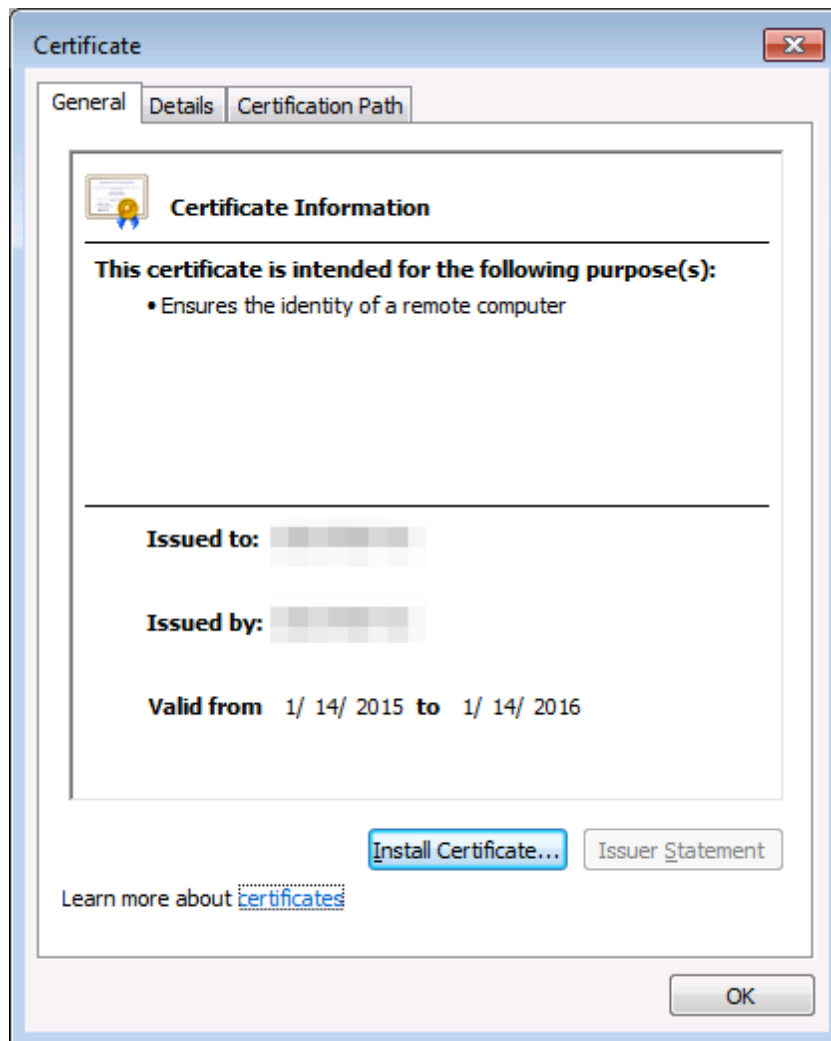
To establish trust with a Web server using Internet Explorer 11:

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**.



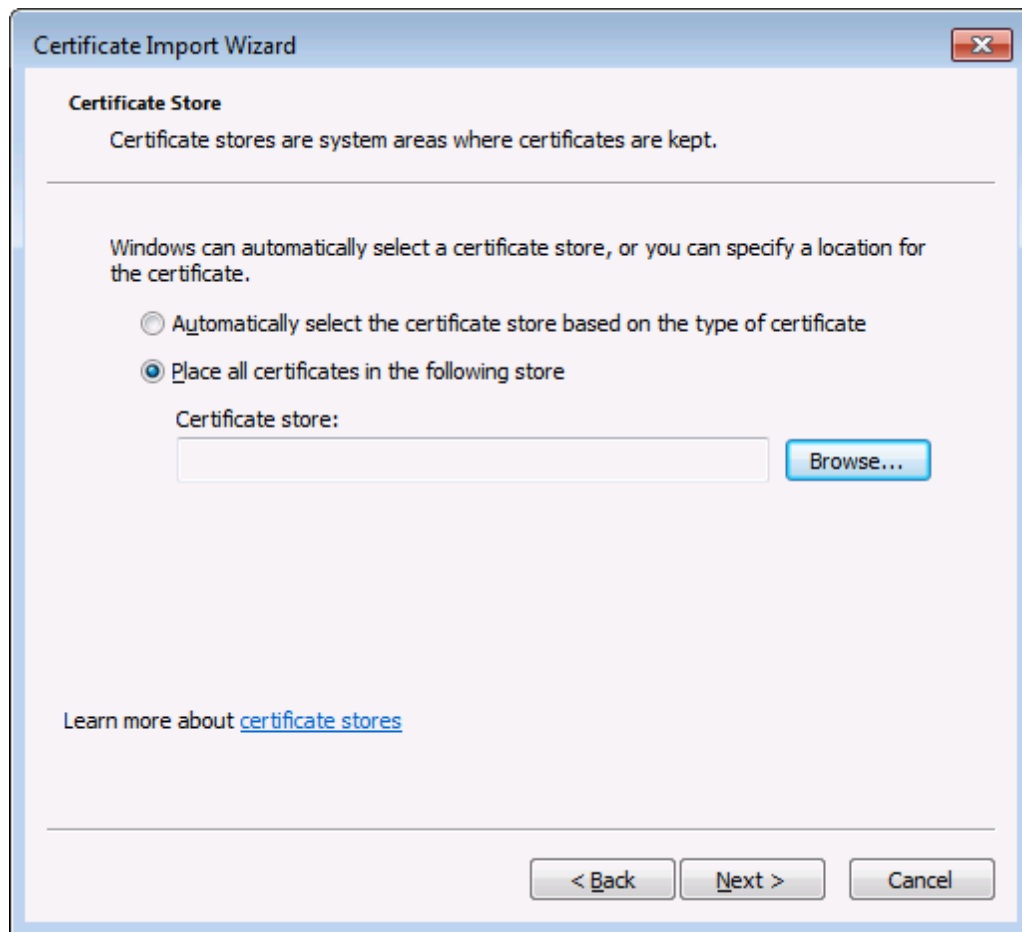
4. Click **Install Certificate**.



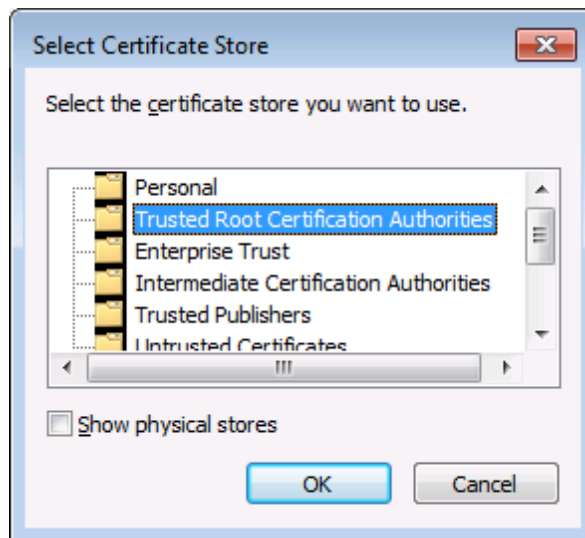
5. Click **Next**.



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.

8.4 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**.

8.5 Exporting Certificates from Windows

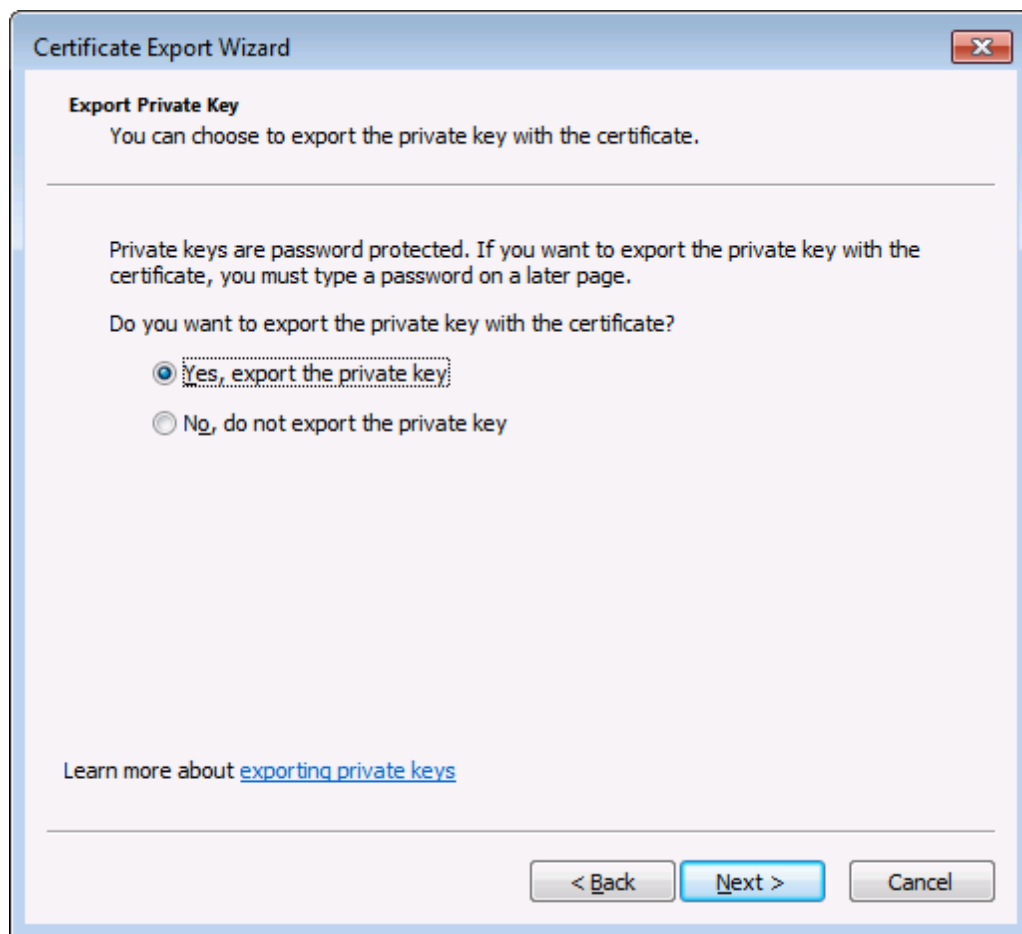
For mappings that call Web services through HTTPS and are deployed to an OS X/ macOS or Linux server running MapForce Server or FlowForce Server, the same client certificate must be available on the non-Windows operating system as the one used on Windows to design and test the mapping. To execute such 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:

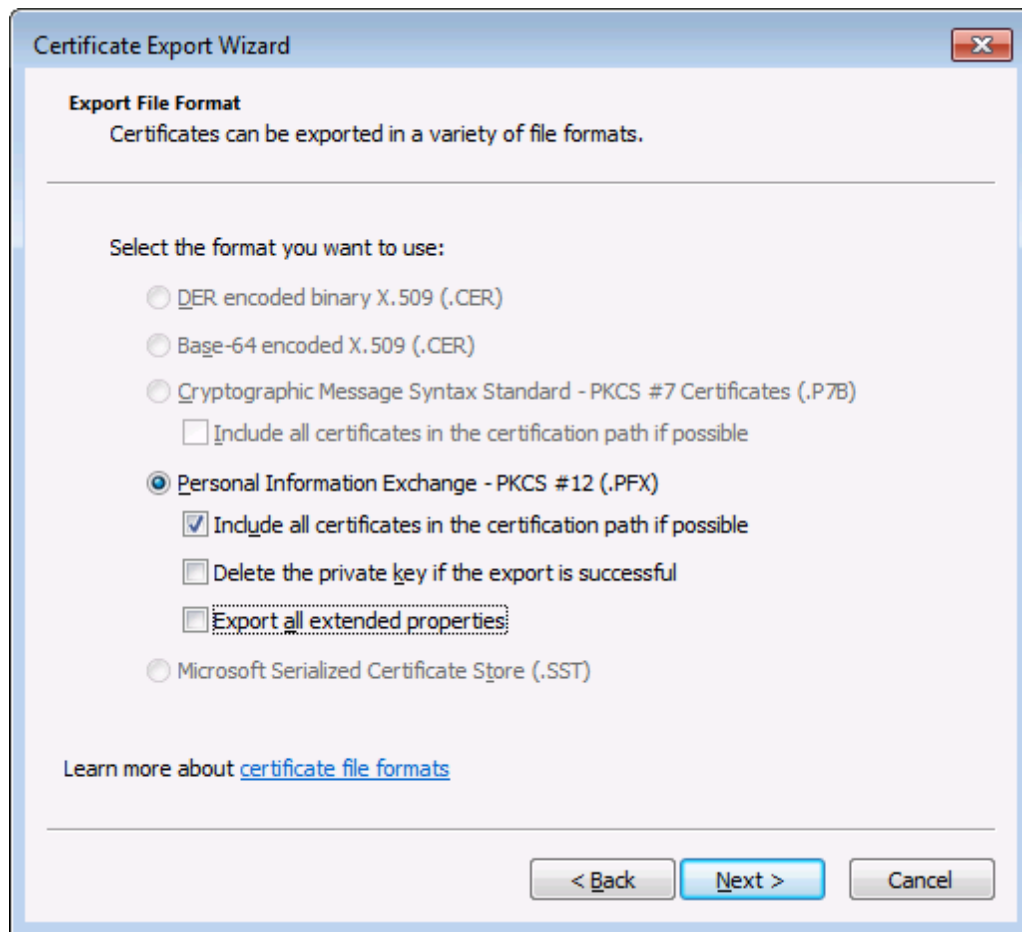
1. On Windows, open the Certificates snap-in (see [Accessing the Certificate Stores on Windows](#)).
2. Right-click the certificate that you want to export, point to **All Tasks**, and then click **Export**.
3. Click **Next**.



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

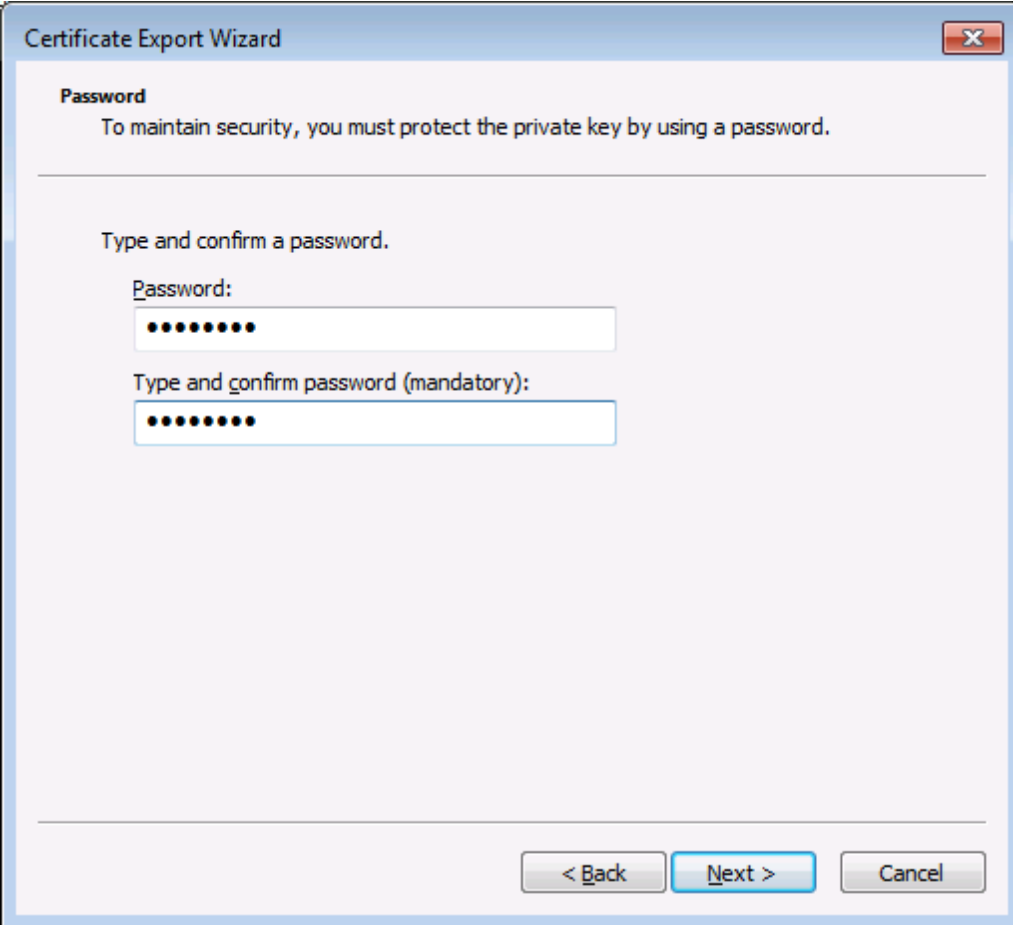


5. Choose the *Personal Information Exchange - PKCS #12 (.pfx)* file format, and then click **Next**.



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.

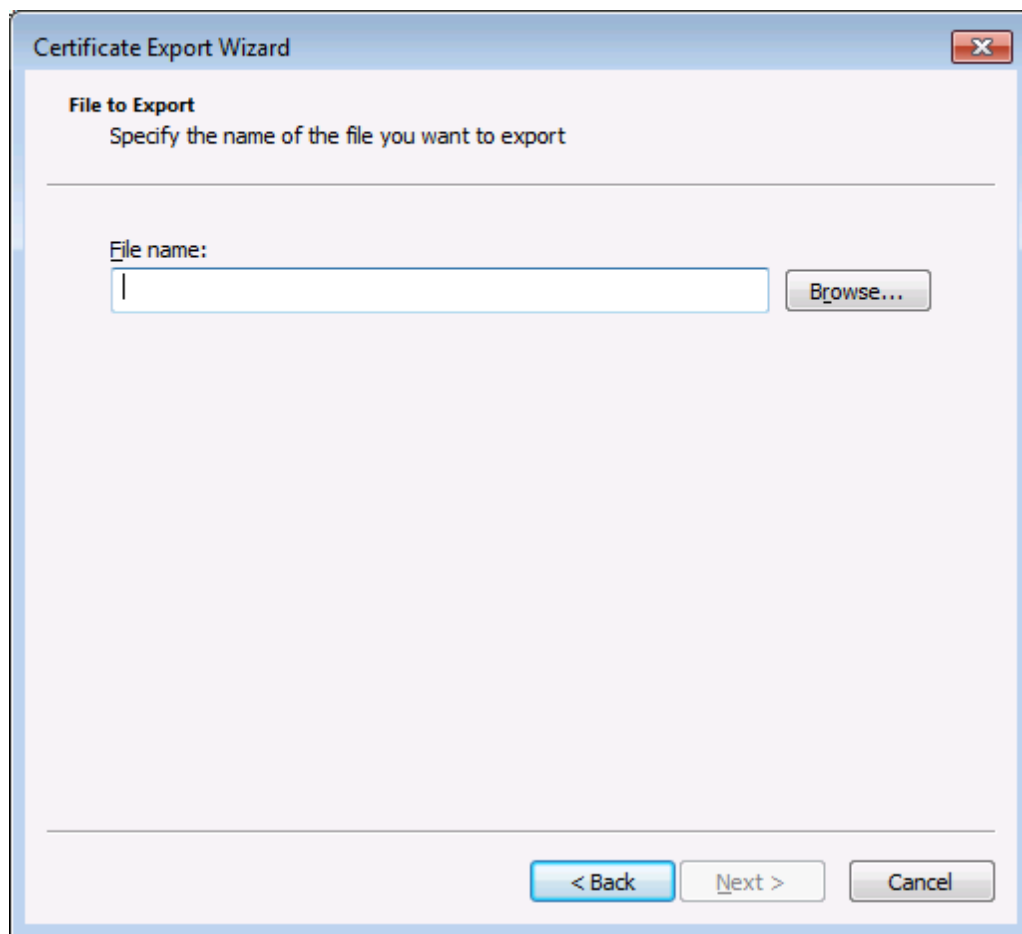
6. Enter a password, and then click **Next**. You will need this password after you copy the certificate to the target operating system.



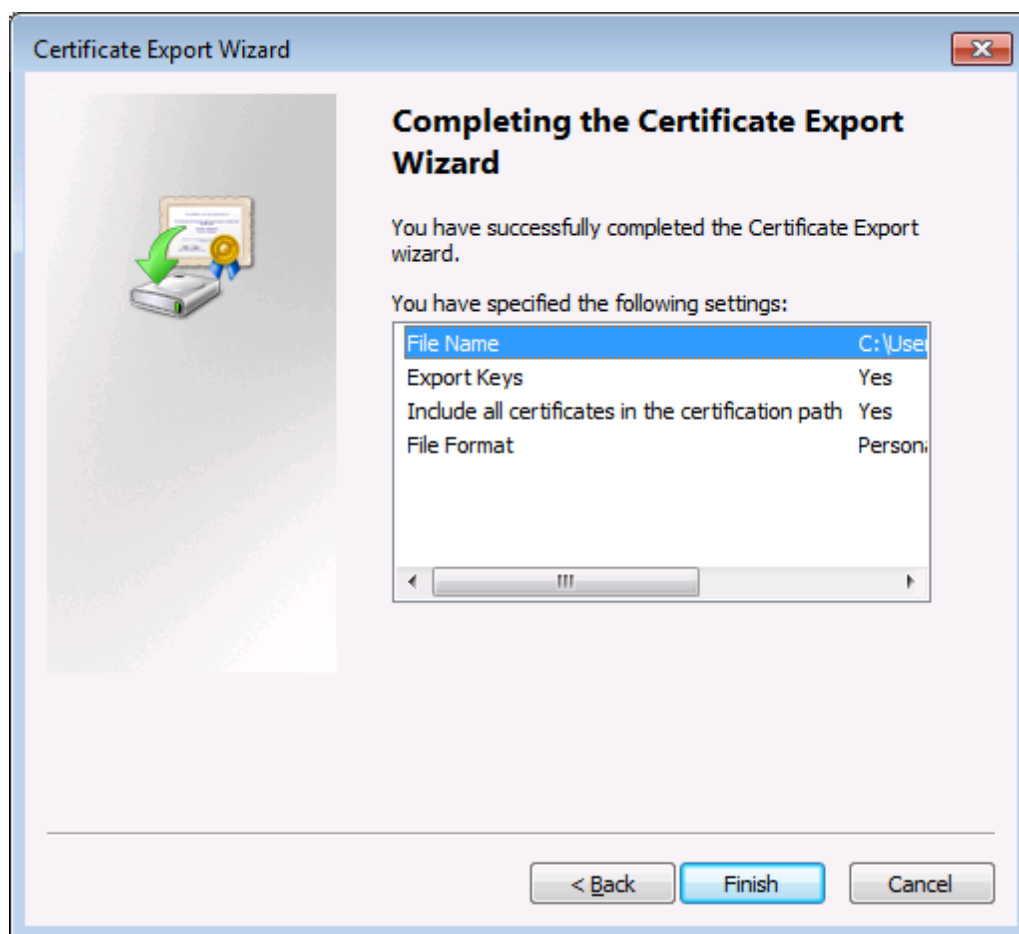
The image shows a Windows dialog box titled "Certificate Export Wizard". The window has a standard Windows title bar with a close button (X) in the top right corner. The main content area is light gray and contains the following text and controls:

- Password**: A section header.
- To maintain security, you must protect the private key by using a password.**: A message in a light gray box.
- Type and confirm a password.**: A general instruction.
- Password:**: A label for the first text input field.
- Type and confirm password (mandatory):**: A label for the second text input field.
- Two text input fields, both containing ten black dots to represent masked characters.
- At the bottom right, three buttons: "< Back" (disabled), "Next >" (active/highlighted), and "Cancel" (disabled).

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



8. Click **Finish**.



8.6 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. Open in MapForce the mapping which calls the Web service.
2. Double-click the header of the Web Service component. The Component Settings dialog box appears.
3. Click **Edit** next to HTTP Security Settings.
4. In the HTTP Security Settings dialog box, click **Client Certificate**, and then select the required certificate from the **Current User\Personal** store on Windows.
5. Save the mapping and compile it to a mapping execution file or deploy it to FlowForce Server.
6. 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.

8.7 Client Certificates on OS X / macOS

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. Open in MapForce the mapping which calls the Web service.
2. Double-click the header of the Web Service component. The Component Settings dialog box appears.
3. Click **Edit** next to HTTP Security Settings.
4. In the HTTP Security Settings dialog box, click **Client Certificate**, and then select the required certificate.
5. If the certificate name does not match exactly the host name of the server, select **Allow name mismatch between certificate and request**.
6. Save and deploy the mapping to the target operating system .
7. 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.

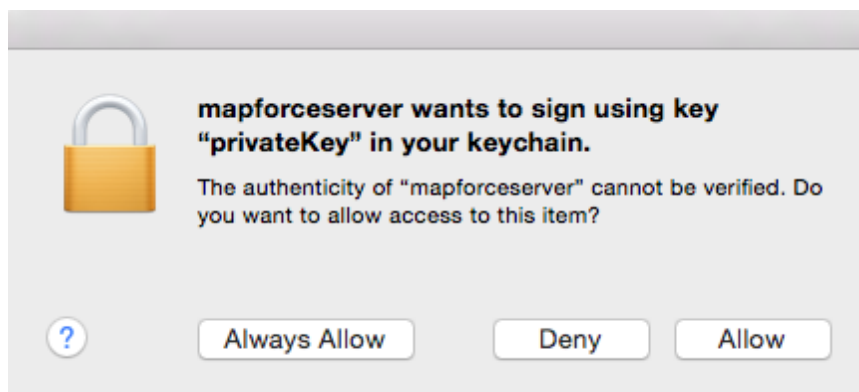
To transfer the client certificate to OS X / macOS:

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 OS X / macOS.
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:

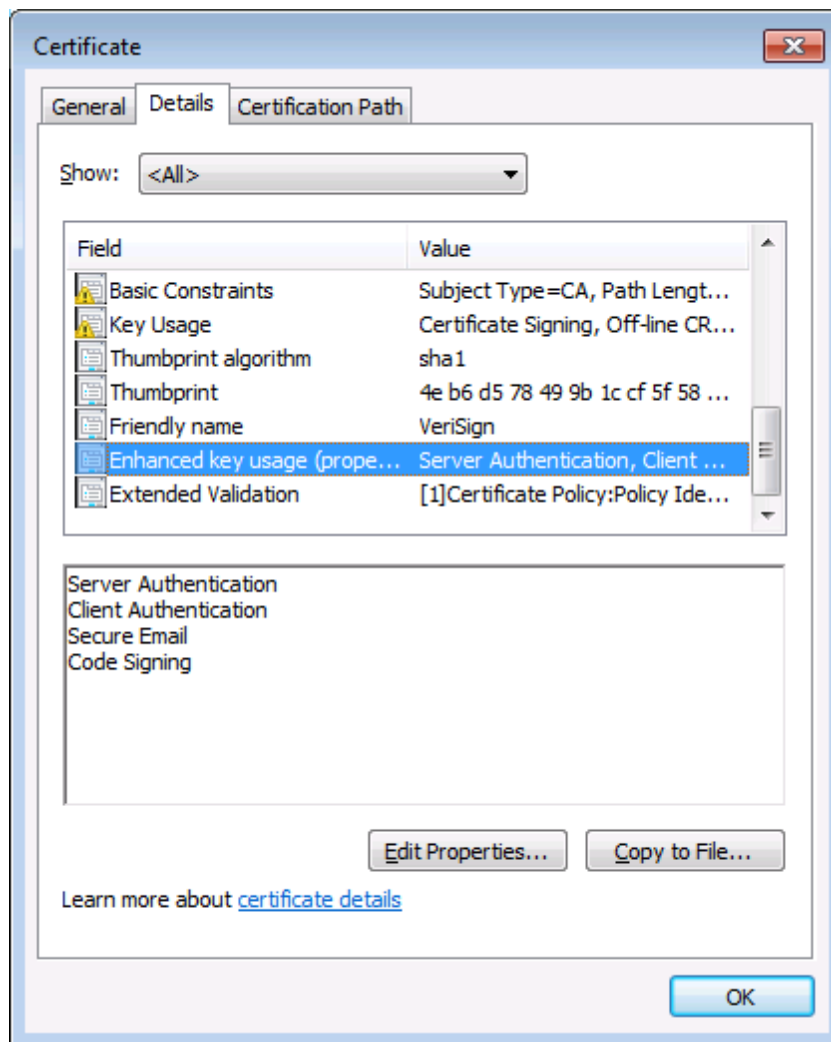
- If you execute the mapping remotely through SSH, first unlock the keychain with the `security unlock-keychain` command.
- If you execute the mapping through the OS X / macOS graphical user interface, when prompted to allow MapForce Server access to the keychain, click **Allow**.



8.8 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 [https://technet.microsoft.com/en-us/library/cc754489\(v=ws.11\).aspx](https://technet.microsoft.com/en-us/library/cc754489(v=ws.11).aspx).
- 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

A

Administrator interface, 142
Alert emails, 156
Altova LicenseServer,
 (see LicenseServer), 96
Altova ServiceController, 108
Assigning licenses, 136

C

Client machine monitoring, 155
Client management pane, 150
Command line usage, 42
Commands,
 exportresourcestrings, 46
 help, 48
 licenseserver, 50
 run, 52
 setdeflang, 56
Configuration page, 142
 opening on Linux, 116
 opening on Mac OS X, 118
 opening on Windows, 113
 URL of, 113
 URL of (Linux), 116
 URL of (Mac OS X), 118

D

Default password, 113
Desktop products,
 registering with LicenseServer, 124
Digital certificates,
 exporting from Windows, 178
 in MapForce mappings, 166
 managing on Windows, 177
 transferring to Linux, 184
 transferring to Mac, 186

 trusting on Linux, 169
 trusting on Mac, 172
 trusting on Windows, 173

E

Evaluation licenses, 150
exportresourcestrings command, 46

F

FlowForce Server,
 registering with LicenseServer, 125

H

help command, 48
Hostname of clients, 98
HTTPS,
 calling Web services through, 166

I

Installation on Linux, 27
Installation on Mac OS X, 35
Installation on Windows, 19
IP address of clients, 98

L

License Pool, 121, 143
Licenses,
 assigning, 136, 150
 managing, 150
 uploading, 121, 143
LicenseServer,
 Configuration page, 142
 installation on Linux, 103
 installation on Mac OS X, 106
 installation on Windows, 101

LicenseServer,

- interface with, 142
- registering desktop products with, 124
- registering FlowForce Server with, 125
- registering MapForce Server with, 130
- registering MobileTogether Server with, 132
- registering RaptorXML(+XBRL) Server with, 133
- registering StyleVision Server with, 134
- settings, 156
- starting, 111
- steps for assigning licenses, 109

licenseserver command, 50**LicenseServer configuration page,**

- (see Configuration page), 113, 116, 118

Licensing on Linux, 30**Licensing on Mac OS X, 38****Licensing on Windows, 21****Linux,**

- executing mappings with Web service calls through HTTPS, 184
- installation on, 27
- licensing on, 30
- transferring client certificates to, 184
- trusting server certificates on, 169

Localization, 46, 56**Logout, 163**

M

Mac,

- executing mappings with Web service calls through HTTPS, 186
- transferring client certificates to, 186
- trusting server certificates on, 172

Mac OS X,

- installation on, 35
- licensing on, 38

MapForce Server,

- registering with LicenseServer, 130

Messages, 163**MobileTogether Server,**

- registering with LicenseServer, 132

N

Network information,

- and IP address of client, 98
- connections via VPN, 98
- connection procedure, 98

Network settings, 156

P

Password,

- default at startup, 113

Password reset, 164

R

RaptorXML(+XBRL) Server,

- registering with LicenseServer, 133

Registering desktop products with LicenseServer, 124**Registering FlowForce Server with LicenseServer, 125****Registering MapForce Server with LicenseServer, 130****Registering MobileTogether Server with LicenseServer, 132****Registering RaptorXML(+XBRL) Server with LicenseServer, 133****Registering StyleVision Server with LicenseServer, 134****Resetting the password, 164****run command, 52**

S

Server Management tab, 136**ServiceController, 108****setdeflang command, 56****Settings, 156****Setup,**

- on Linux, 26
- on Mac OS X, 34
- on Windows, 18

StyleVision Server,

- registering with LicenseServer, 134

U

Unregistering products and clients, 150

Updates, 100

Uploading licenses, 121, 143

V

VPN connections to LicenseServer, 98

W

Web services,

calling through HTTPS, 166

Windows,

executing mappings with Web service calls through HTTPS,
188

installation on, 19

licensing on, 21

trusting server certificates on, 173