User and Reference Manual

Altova MobileTogether Server
User & Reference Manual

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: 2017

Table of Contents

Chapter 1

Altova MobileTogether Server

1 Altova MobileTogether Server

MobileTogether Server serves MobileTogether solutions to client mobile devices. It runs on MS Windows, Linux, and macOS machines.

- MobileTogether solutions are created in Altova's MobileTogether Designer application and are deployed from MobileTogether Designer to MobileTogether Server.
- The MobileTogether Client app that is installed on client mobile devices then accesses MobileTogether solutions that are deployed on a MobileTogether Server.

MobileTogether Server has an easy-to-use Web UI that provides management of server processes and logs. This user manual describes how to set up MobileTogether Server and manage its processes.

Copyright ©2017 Altova GmbH. All rights reserved. Use of this software is governed by an Altova license agreement. XMLSpy, MapForce, StyleVision, SchemaAgent, UModel, DatabaseSpy, DiffDog, Authentic, MissionKit, FlowForce, RaptorXML, MobileTogether, and Altova as well as their respective logos are either registered trademarks or trademarks of Altova GmbH. Protected by U.S. Patents 7,739,292, 7,200,816, and other pending patents. This software contains third party software or material that is protected by copyright and subject to other terms and conditions as detailed on the Altova website at http://www.altova.com/legal_3rdparty.html.

This documentation

This documentation is organized into the following sections:

- Introduction
- Setting Up MobileTogether Server
- Server Procedures
- Web UI Reference
- Command Line Usage
- Altova LicenseServer

Current version: 3.2

Last updated: 04 May 2017

Chapter 2

Introduction

2 Introduction

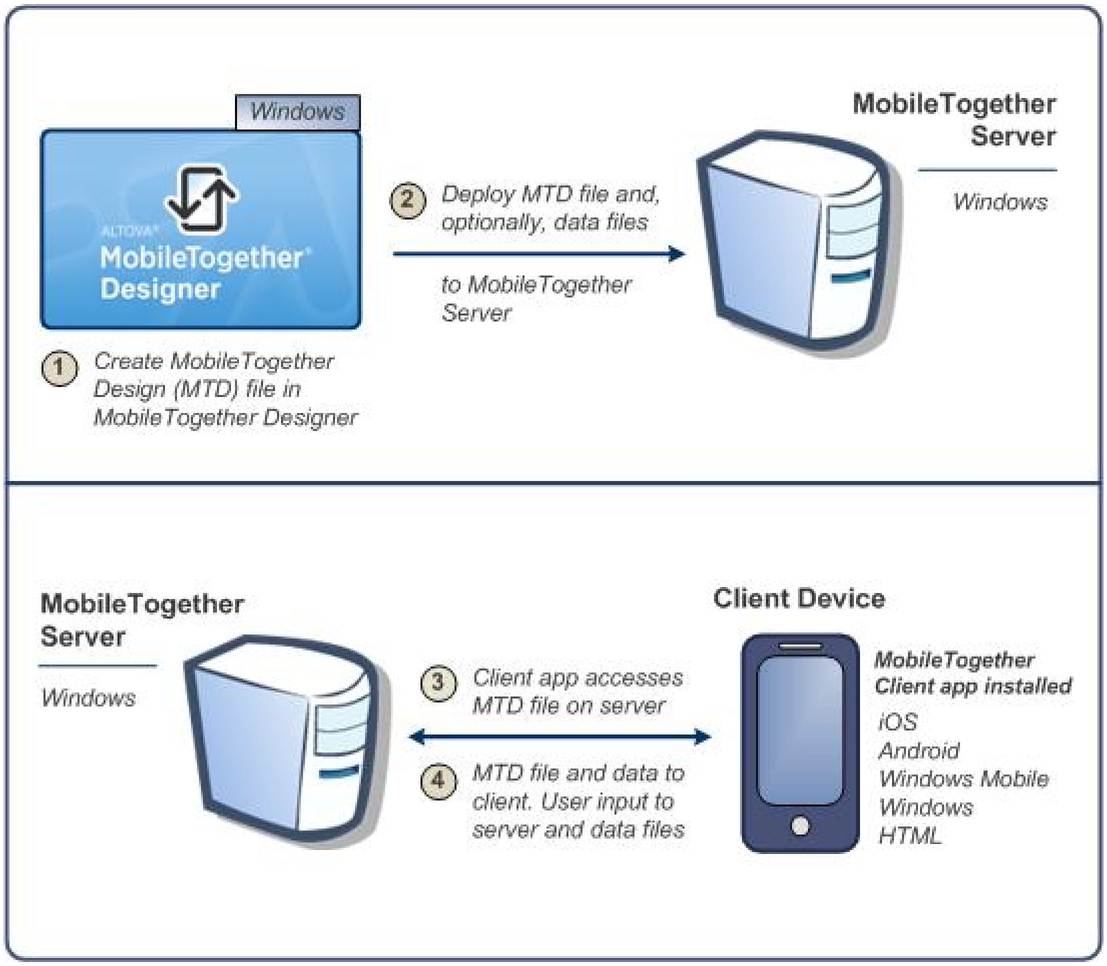
This introduction:

- MobileTogether Overview, which describes the MobileTogether system and the place of MobileTogether Server within that system
- Using MobileTogether Server section, which lists the broad steps required to set up MobileTogether Server for use with MobileTogether Client apps

2.1 MobileTogether Overview

MobileTogether consists of the following modules:

- MobileTogether Designer: MobileTogether solutions for mobile clients are created and deployed to MobileTogether Server. *See the MobileTogether Designer user manual.*
- MobileTogether Server: Serves MobileTogether solutions to MobileTogether Client apps installed on mobile devices. *See the section, Server Procedures, for descriptions of server administration tasks.*
- MobileTogether Client app (for mobile devices): Connects to a MobileTogether Server and accesses the MobileTogether solutions deployed on that server. *See the MobileTogether Client app user manual.*


Windows
ALTOVA®
MobileTogether®
Designer
1 Create MobileTogether Design (MTD) file in MobileTogether Designer
2 Deploy MTD file and, optionally, data files
to MobileTogether Server
MobileTogether Server
Windows
MobileTogether Server
Windows
3 Client app accesses MTD file on server
4 MTD file and data to client. User input to server and data files
Client Device
MobileTogether Client app installed
iOS
Android
Windows Mobile
Windows
HTML

System requirements

▼ **MobileTogether Designer**

Windows	Windows Vista, Windows 7/8/10
Windows Server	Windows Server 2008 R2 or newer

MobileTogether Server

Windows	Windows Vista, Windows 7/8/10
Windows Server	Windows Server 2008 R2 or newer
Linux	• CentOS 6 or newer • RedHat 6 or newer • Debian 7 or newer • Ubuntu 12.04 or newer
(Mac) OS X, macOS	OS X10.10, 10.11, macOS 10.12 or newer

MobileTogether Client

iOS	9 and higher for Apple mobile devices
Android	4.0 and higher for Android mobile devices
Windows Mobile	Windows Phone 8.1 for Windows phones
Windows RT, Metro	Windows 8.1, 10; Windows RT for Windows touch-enabled PCs and tablet computers
HTML	HTML browsers for any other mobile devices

2.2 Using MobileTogether Server

To set up MobileTogether Server for use with MobileTogether clients:

- Install and configure MobileTogether Server
- Deploy MobileTogether solutions from MobileTogether Designer to MobileTogether Server
- Configure MobileTogether Client apps (on mobile devices) to access solutions on a MobileTogether Server

---

The steps in more detail:

1. Install MobileTogether Server

   MobileTogether Server runs on Windows, Linux, and macOS systems. Before installing a new version of MobileTogether Server, de-install any previous version. See Installation on Windows, Installation on Linux, and Installation on macOS.

2. License MobileTogether Server

   In order to license MobileTogether Server, it must be able to connect to a LicenseServer on your network. Start MobileTogether Server, register MobileTogether Server with LicenseServer, and assign a license to MobileTogether Server from LicenseServer. See Licensing on Windows, Licensing on Linux, and Licensing on macOS.

3. Set up SSL encryption

   If you wish to encrypt server-client communication, you can set up SSL encryption for MobileTogether Server (*see Setting Up SSL Encryption*). Additionally, you will need to configure MobileTogether Client apps to communicate via SSL. *See the MobileTogether Client app user manual*.

4. Define basic settings

   Basic settings include administrator and client ports, and other communication settings and security settings.

5. Set up user accounts

   MobileTogether Server is always accessed via a user account, so user accounts have to be set up appropriately. There are two types of access:

   - *Administrator access:* Administrator access is via the Web UI and is used to carry out administrative tasks. Administrative actions include defining communication settings, security settings, and managing user accounts.
   - *End-user access:* End user access is via a mobile device and is used to download MobileTogether solutions to the client. Access to solutions on the

server is determined by the user account the client logs in with.

6. Deploy MobileTogether solutions to MobileTogether Server

   MobileTogether solutions are deployed from within the MobileTogether Designer application. *See the MobileTogether Designer user manual*.

7. Configure MobileTogether Client apps to access MobileTogether Server

   MobileTogether Client apps on mobile devices must be configured to connect to MobileTogether Server. The MobileTogether Server information that is required for configuring MobileTogether Client apps is listed in the section, Information for Clients. *Also see the MobileTogether Client app user manual*.

Server IP address and network firewall settings

Your server can have a public IP address (accessible over the Internet) and/or a private IP address (accessible within a private network; for example, via WiFi within a company network). If a mobile client device tries to connect via the Internet using the server's private IP address, then the connection will not work. This is because the private IP address is not known on the Internet and cannot be resolved. If a client device uses a private IP address, then the client device must already have access to the private network.

To ensure that the server can be accessed, do one of the following:

- Provide the server with a public IP address so that it can be reached via the Internet. On the client device, use this public IP address to access the server.
- If you use a firewall and install MobileTogether Server on a server with a private IP address (inside the private network), then use the network firewall to forward requests sent to a public IP-address/port-combination to your MobileTogether Server server. On the client device, use the public IP address.

You must also ensure that the firewall is configured to allow access to the server port used for MobileTogether Client communication. The ports used by MobileTogether Server are specified in the Settings page of the the Web UI of MobileTogether Server (*see the MobileTogether Server user manual*). On the client device, this is the port that must be specified as the server port to access.

Tip: Port 80 is usually open on most firewalls by default. So, if you are having difficulties with firewall settings and if port 80 is not already bound to some other service, you could specify port 80 as the MobileTogether Server port for client communication.

Chapter 3

Setting Up MobileTogether Server

3 Setting Up MobileTogether Server

This section describes installation, licensing and other setup procedures. It is organized into the following sections:

- Setup on Windows
- Setup on Linux
- Setup on macOS
- Setting Up SSL Encryption

3.1 Setup on Windows

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

Installation on Windows

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

Licensing on Windows

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

Installation on Windows

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

▼ System requirements

▼ *Windows*

Windows Vista, Windows 7/8/10

▼ *Windows Server*

Windows Server 2008 R2 or newer

▼ Installing MobileTogether Server

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

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

```
<ProgramFilesFolder>\Altova\MobileTogetherServer3.2\bin
\MobileTogetherServer.exe
```

▼ Altova LicenseServer

- In order for MobileTogether Server to work, it must be licensed via an Altova LicenseServer on your network.
- When you install MobileTogether Server on Windows systems, an option is available that allows you to download and install Altova LicenseServer together with MobileTogether 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 MobileTogether 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 MobileTogether Server with Altova LicenseServer.

▼ LicenseServer versions

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

Current version: 2.3

▼ Trial license

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

▼ Application folder location

The application will be installed in the following folder:

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

Licensing on Windows

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

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

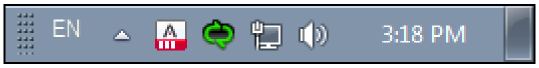
The steps you need to carry out are given below.

▾ Start ServiceController

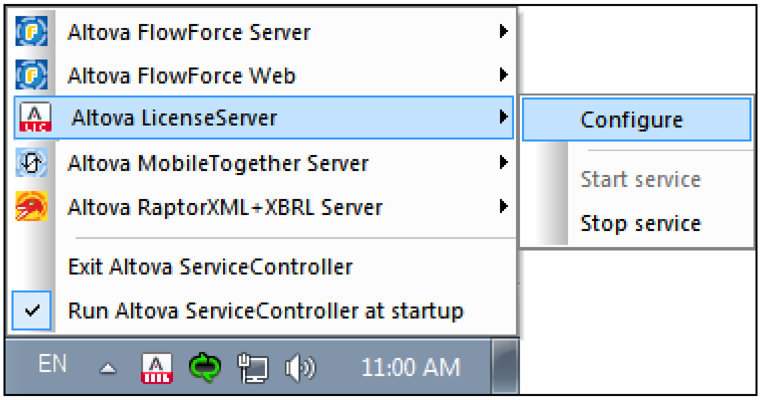
Altova ServiceController is started in order to start Altova LicenseServer and Altova MobileTogether Server.

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*).

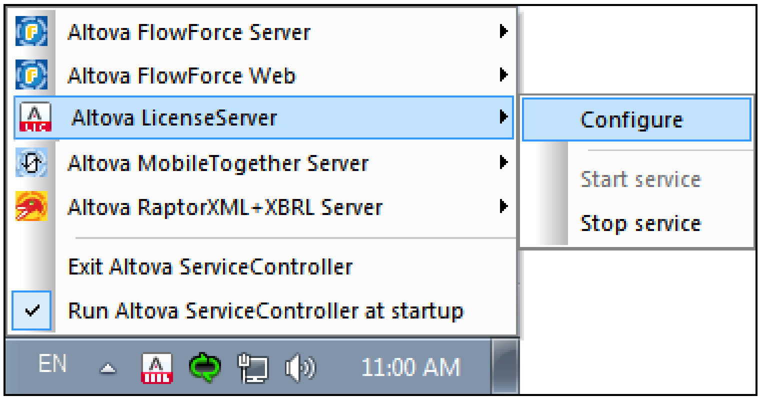


To specify that ServiceController starts automatically on logging in to the system, click the **ServiceController** icon in the system tray to display the **ServiceController** menu (*screenshot below*), and then toggle on the command **Run Altova ServiceController at Startup**. (This command is toggled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


Altova FlowForce Server
Altova FlowForce Web
Altova LicenseServer
Altova MobileTogether Server
Altova RaptorXML+XBRL Server
Exit Altova ServiceController
✓ Run Altova ServiceController at startup
Configure
Start service
Stop service
EN
11:00 AM

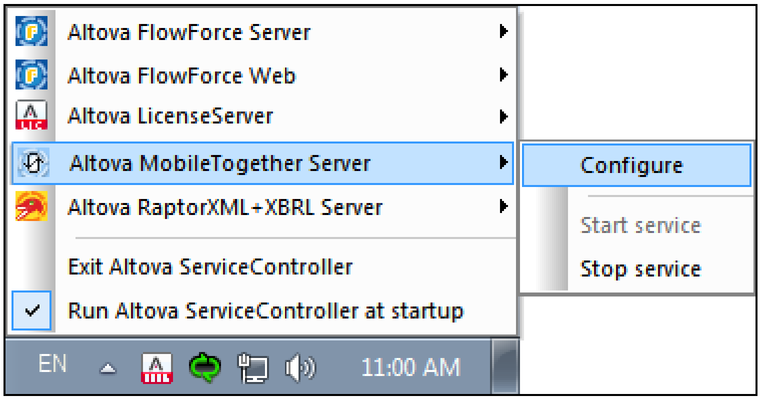
- Start LicenseServer

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.


Altova FlowForce Server
Altova FlowForce Web
Altova LicenseServer
Altova MobileTogether Server
Altova RaptorXML+XBRL Server
Exit Altova ServiceController
✓ Run Altova ServiceController at startup
Configure
Start service
Stop service
EN
11:00 AM

- Start 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.


Altova FlowForce Server
Altova FlowForce Web
Altova LicenseServer
Altova MobileTogether Server
Altova RaptorXML+XBRL Server
Exit Altova ServiceController
Run Altova ServiceController at startup
Configure
Start service
Stop service
EN
11:00 AM

▼ Register MobileTogether Server

To register MobileTogether Server from the command line interface, use 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
```

Alternatively, you can register MobileTogether Server from the Settings tab of MobileTogether Server's Web UI. Essentially: (i) Start MobileTogether Server 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**.

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

▼ Assign a license

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

Note on cores and licenses

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

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

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

MobileTogether Server licenses

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

3.2 Setup on Linux

This section describes the installation and licensing of MobileTogether 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 MobileTogether Server
- Altova LicenseServer
- LicenseServer versions
- Trial license

Licensing on Linux

- Start LicenseServer
- Start MobileTogether Server
- Register MobileTogether Server
- Assign a license

Notes about Environment

Installation on Linux

MobileTogether 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
MobileTogether Server	qt4, krb5-libs, qt-x11	libgssapi-krb5-2	libgssapi-krb5-2

▼ Uninstall old versions of Altova server products

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

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

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

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

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

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

MobileTogether 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 MobileTogether Server, you may want to download LicenseServer from the Altova website at the same time as you download MobileTogether Server, rather than download it at a later time.

▼ Install MobileTogether 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 MobileTogether Server with the following command:

```
  [Debian]:   sudo dpkg --install mobiletogetherserver-3.2-debian.deb
  [Ubuntu]:   sudo dpkg --install mobiletogetherserver-3.2-ubuntu.deb
  [CentOS]:   sudo rpm -ivh mobiletogetherserver-3.2-1.x86_64.rpm
  [RedHat]:   sudo rpm -ivh mobiletogetherserver-3.2-1.x86_64.rpm
```

The MobileTogether Server package will be installed in the folder:

```
  /opt/Altova/MobileTogetherServer3.2
```

▼ Altova LicenseServer

In order for any Altova Server product—including MobileTogether 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 MobileTogether Server (*see previous step*).

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

The LicenseServer package will be installed in:

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

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

▼ LicenseServer versions

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

Current version: 2.3

▼ Trial license

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

Licensing on Linux

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

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

The steps you need to carry out are given below.

▼ Start LicenseServer

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

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

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

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

▼ Start MobileTogether Server

Start MobileTogether Server as a daemon with the following command:

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

To set up and configure MobileTogether Server, open its Web UI (Setup) page by entering the URL of the Web UI page in the address bar of an Internet browser: **`http://<serverIPAddressOrName>:8085`**.

Firewall Note
Make sure that the port address is not blocked by your firewall.

▼ Register MobileTogether Server

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

```
sudo /opt/Altova/MobileTogetherServer3.2/bin/mobiletogetherserver
licenseserver [options] ServerName-Or-IP-Address
```

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

```
sudo /opt/Altova/MobileTogetherServer3.2/bin/mobiletogetherserver
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 MobileTogether Server executable is:

```
/opt/Altova/MobileTogetherServer3.2/bin/
```

You can also register MobileTogether Server from the Settings tab of MobileTogether Server's Web UI. Essentially: (i) Start MobileTogether Server 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**.

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

▼ Assign a license

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

Note on cores and licenses

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

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

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

MobileTogether Server licenses

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

Notes about Environment

Folders

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

⊟ *Installation root*

`/opt/Altova/MobileTogetherServer3.2/`

⊟ *Database, License, and Solution Files*

`/var/opt/Altova/MobileTogetherServer`

⊟ *Environment settings*

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

File-based databases

File-based databases (such as SQLite databases) must reside in the folder defined in the Settings tab of MobileTogether Server as the Server Side's Solutions Working Directory. By default, this folder is:

`/var/opt/Altova/MobileTogetherServer/SolutionFiles`

JDBC-Connections

Note the following points:

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

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

Listing of important files

The following shell script is copied to the folder **`/opt/Altova/MobileTogetherServer3.2/etc`** so as not to overwrite already existing configuration files. Make the necessary changes as required. The parts highlighted in blue are environment-specific and will need to be adjusted to match your environment:

Shell script

```
#- jdbc - environment -
export PATH=/usr/local/jdk1.7.0_17/bin:/usr/lib64/qt-3.3/bin:/usr/local/
bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/home/qa/bin
export JAVA_HOME=/usr/local/jdk1.7.0_17
export CLASSPATH=/usr/local/jdbc/oracle/ojdbc6.jar:/usr/local/jdbc/oracle/
xdb.jar:/usr/local/jdbc/oracle/xmlparserv2.jar:/usr/local/jdbc/postgre/
postgresql-9.0-801.jdbc4.jar:/usr/local/jdbc/mssql/sqljdbc4.jar:/usr/local/
jdbc/iseries/lib/jt400.jar:/usr/local/jdbc/mysql/mysql-connector-java-
5.1.16-bin.jar:/usr/local/jdbc/sqlite/sqlitejdbc-v056.jar:/usr/local/jdbc/
Informix_JDBC_Driver/lib/ifxjdbc.jar:/usr/local/jdbc/sybase/jconn7/
jconn4.jar:/usr/local/jdbc/db2/db2jcc.jar:/usr/local/jdbc/db2/
db2jcc_license_cu.jar:./:
```

3.3 Setup on macOS

This section describes the installation and licensing of MobileTogether Server on macOS systems.

Installation on macOS

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

Licensing on macOS

- Start LicenseServer
- Start MobileTogether Server
- Register MobileTogether Server
- Assign a license

Notes about Environment

Installation on macOS

MobileTogether 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, 10.11, macOS 10.12 or newer
Java for OS X, macOS (latest version)

▼ Install Java for OS X

You must install Java for OS X in order for MobileTogether Server to run. The latest version at the time of writing (Java for OS X 2014-001) is available at: http://support.apple.com/kb/DL1572. For the latest Java for OS X version provided by Apple when you install, do a search for Java for OS X at the Apple website. (Note that this Java version for OS X might not be the latest Java version provided by Sun Microsystems, but this is the version you need to install.)

▼ Uninstall old versions of Altova server products

Before uninstalling MobileTogether Server, stop the service with the following command:

```
sudo launchctl unload /Library/LaunchDaemons/
com.altova.MobileTogetherServer3.2.plist
```

To check whether the service has been stopped, open the Activity Monitor in Finder and make sure that MobileTogether Server is not in the list.In the Applications folder in Finder, right-click the MobileTogether 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/MobileTogetherServer3.2/
```

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 MobileTogether Server.

▼ Download the disk image file

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

▼ Install MobileTogether Server

Click to open the downloaded disk image (.dmg). This causes the MobileTogether 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 MobileTogether Server package will be installed in the folder:
`/usr/local/Altova/MobileTogetherServer3.2` (application binaries)
`/var/Altova/MobileTogetherServer` (data files: database and logs)

MobileTogether Server server daemon starts automatically after installation and a re-boot of the machine. You can start MobileTogether Server as a daemon with the following command:

```
sudo launchctl load /Library/LaunchDaemons/
com.altova.MobileTogetherServer3.2.plist
```

After starting MobileTogether Server server daemon, you can open the Web UI page of MobileTogether Server in order to configure MobileTogether Server. Open the Web UI page by going to the Applications folder in Finder and double-clicking the MobileTogether Server icon there.

▼ Altova LicenseServer

In order for any Altova Server product—including MobileTogether 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 MobileTogether 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 MobileTogether Server version, or with a later version of LicenseServer.
- The LicenseServer version that is appropriate for a particular version of MobileTogether Server is displayed during the installation of MobileTogether Server. You can install this version of LicenseServer along with MobileTogether 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 MobileTogether Server.
- If you install a new version of MobileTogether Server and if your installed LicenseServer version is older than the appropriate LicenseServer, install the latest version available on the Altova website.
- At the time of LicenseServer de-installation, all registration and licensing information held in the older version of LicenseServer will be saved to a database on your server machine. This data will be imported automatically into the newer version when the newer version is installed.
- The version number of the currently installed LicenseServer is given at the bottom of the LicenseServer configuration page (all tabs).

Current version: 2.3

▼ Trial license

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

Licensing on macOS

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

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

The steps you need to carry out are given below.

▼ Start LicenseServer

To correctly register and license MobileTogether 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
```

▼ Start MobileTogether Server

MobileTogether Server server daemon starts automatically after installation and a re-boot of the machine. You can start MobileTogether Server as a daemon with the following command:

```
sudo launchctl load /Library/LaunchDaemons/
com.altova.MobileTogetherServer3.2.plist
```

If at any time you need to stop MobileTogether Server, use:

```
sudo launchctl unload /Library/LaunchDaemons/
com.altova.MobileTogetherServer3.2.plist
```

To set up and configure MobileTogether Server, open its Web UI (Setup) page in one of the following ways:

- Double-click the MobileTogether Server 3.2 icon in the Applications folder of the Finder
- Enter the URL of the Web UI page in the address bar of an Internet browser: **http://*<serverIPAddressOrName>*:8085**

Firewall Note
Make sure that the port address is not blocked by your firewall.

▼ Register MobileTogether Server

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

```
sudo /usr/local/Altova/MobileTogetherServer3.2/bin/MobileTogetherServer
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/MobileTogetherServer3.2/bin/MobileTogetherServer
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 MobileTogether Server executable is:

```
/usr/local/Altova/MobileTogetherServer3.2/bin/
```

You can also register MobileTogether Server from the Settings tab of MobileTogether Server's Web UI. Essentially: (i) Start MobileTogether Server 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**.

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

▾ Assign a license

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

Note on cores and licenses

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

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

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

MobileTogether Server licenses

MobileTogether Server licenses are based on the number of CPU cores on the

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

Notes about Environment

Folders

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

⊟ *Installation root*

`/usr/local/Altova/MobileTogetherServer3.2/`

⊟ *Database, License, and Solution Files*

`/var/Altova/MobileTogetherServer`

⊟ *Environment settings*

`/Library/LaunchDaemons/com.altova.MobileTogetherServer.plist`

The environment settings file must be defined according to your specific environment. The example path above serves only as a general guide.

Note: These environment variables are only set for the MobileTogether Server process and do not have an impact on other users.

File-based databases

File-based databases (such as SQLite databases) must reside in the folder defined in the Settings tab of MobileTogether Server as the Server Side's Solutions Working Directory. By default, this folder is:

`/var/Altova/MobileTogetherServer/SolutionFiles`

JDBC-Connections

Note the following points:

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

Note

On macOS servers, the only database connections supported are JDBC.

Listing of important files

The `Plist` file is installed in the `/Library/LaunchDaemons` folder. The parts highlighted in blue are environment-specific and will need to be adjusted to match your environment:

⊟ **Plist file**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/
DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
   <dict>
      <key>Label</key>
      <string>com.altova.MobileTogetherServer</string>
      <key>ProgramArguments</key>
      <array>
         <string>/usr/local/Altova/MobileTogetherServer3.2/bin/
MobileTogetherServer</string>
         <string>debug</string>
      </array>
      <key>KeepAlive</key>
      <true/>
      <key>UserName</key>
      <string>_altovamobiletogetherserver</string>
      <key>EnvironmentVariables</key>
      <dict>
         <key>CLASSPATH</key>
         <string>/usr/local/jdbc/oracle/ojdbc6.jar:/usr/local/jdbc/oracle/
xdb.jar:/usr/local/jdbc/oracle/xmlparserv2.jar:/usr/local/jdbc/postgre/
postgresql-9.0-801.jdbc4.jar:/usr/local/jdbc/mssql/sqljdbc4.jar:/usr/local/
jdbc/iseries/lib/jt400.jar:/usr/local/jdbc/mysql/mysql-connector-java-
5.1.16-bin.jar:/usr/local/jdbc/sqlite/sqlitejdbc-v056.jar:/usr/local/jdbc/
Informix_JDBC_Driver/lib/ifxjdbc.jar:/usr/local/jdbc/sybase/jconn7/
jconn4.jar:/usr/local/jdbc/db2/db2jcc.jar:/usr/local/jdbc/db2/
db2jcc_license_cu.jar:./</string>
      </dict>
   </dict>
</plist>
```

3.4 Setting Up SSL Encryption

If you require that communications between your MobileTogether Server and MobileTogether Client devices are encrypted using the SSL protocol, you will need to:

- Generate an SSL private key and create an SSL public key certificate file
- Set up MobileTogether Server for SSL communication.

The steps to do this are listed below.

MobileTogether uses the open-source OpenSSL toolkit to manage SSL encryption. The steps listed below, therefore, need to be carried out on a computer on which OpenSSL is available. OpenSSL typically comes pre-installed on most Linux distributions and on macOS machines. It can also be installed on Windows computers. For download links to installer binaries, see the OpenSSL Wiki.

---

1. Generate a private key

   SSL requires that a **private key** is installed on MobileTogether Server. This private key will be used to encrypt all data sent to MobileTogether Client apps. To create the private key, use the following OpenSSL command:

   ```
   openssl genrsa -out private.key 2048
   ```

   This creates a file called **private.key**, which contains your private key. Note where you save the file. You will need the private key to (i) generate the Certificate Signing Request (CSR), and (ii) to be installed on MobileTogether Server (*see Step 8 below*).

2. Certificate Signing Requests (CSRs)

   A Certificate Signing Request (CSR) is sent to a certificate authority (CA), such as VeriSign or Thawte, to request a public key certificate. The CSR is based on your private key and contains information about your organization. Create a CSR with the following OpenSSL command (which provides the private-key file, `private.key`, that was created in Step 1, as one of its parameters):

   ```
   openssl req -new -nodes -key private.key -out my.csr
   ```

   During generation of the CSR you will need to give information about your organization, such as that listed below. This information will be used by the certificate authority to verify your company's identity.

   - *Country*
   - *Locality* (the city where your business is located)
   - *Organization* (your company name). Do not use special characters; these will invalidate your certificate
   - *Common Name* (the DNS name of your server). This must exactly match your server's official name, that is, the DNS name client apps will use to connect to the server
   - *A challenge password.* Keep this entry blank!

3. Buy an SSL certificate

   Purchase an SSL certificate from a recognized certificate authority (CA), such as VeriSign or Thawte. For the rest of these instructions, we follow the VeriSign procedure. The procedure with other CAs is similar.

   - Go to the VeriSign website.
   - Click **Buy SSL Certificates**.
   - Different types of SSL certificates are available. For MobileTogether Server, Secure Site or Secure Site Pro certificates are sufficient. EV (extended verification) is not necessary, since there is no "green address bar" for users to see.
   - Proceed through the sign-up process, and fill in the information required to place your order.
   - When prompted for the CSR (*created in Step 2*), copy and paste the content of the `my.csr` file into the order form.
   - Pay for the certificate with your credit card.

   Allow time for obtaining a certificate
   Obtaining public key certificates from an SSL certificate authority (CA) typically takes **two to three business days**. Please take this into account when setting up your MobileTogether Server.

4. Receive public key from CA

   Your certificate authority will complete the enrollment process over the next two to three business days. During this time you might get emails or phone calls to check whether you are authorized to request an SSL certificate for your DNS domain. Please work with the authority to complete the process.

   After the authorization and enrollment process has been completed, you will get an email containing the **public key** of your SSL certificate. The public key will be in plain text form or attached as a `.cer` file.

5. Save public key to file

   For use with MobileTogether Server, the public key must be saved in a `.cer` file. If the public key was supplied as text, copy-paste all the lines from

   ```
   --BEGIN CERTIFICATE--
     ...
   --END CERTIFICATE--
   ```

   into a text file that we will call `mycertificate.cer`.

6. Save CA's intermediate certificates to file

   To complete your SSL certificate, you will need two additional certificates: the **primary** and **secondary intermediate certificates**. Your certificate authority (CA) will list content of intermediate certificates on its website.

- Verisign's intermediate certificates: https://knowledge.verisign.com/support/ssl-certificates-support/index?page=content&id=AR657&actp=LIST&viewlocale=en_US
- Verisign's intermediate certificates for its Secure Site product: https://knowledge.verisign.com/support/ssl-certificates-support/index?page=content&id=AR1735

Copy-paste both intermediate certificates (primary and secondary) into separate text files and save them on your computer.

7. Combine certificates in one public key certificate file

You now have three certificate files:

- Public key (`mycertificate.cer`)
- Secondary intermediate certificate
- Primary intermediate certificate

Each contains text blocks bracketed by lines that look like this:

```
--BEGIN CERTIFICATE--
  ...
--END CERTIFICATE--
```

Now copy-paste all three certificates into one file so that they are in sequence. The order of the sequence is important: (i) public key, (ii) secondary intermediate certificate, (iii) primary intermediate certificate. Ensure that there are no lines between certificates.

```
--BEGIN CERTIFICATE--
  public key from mycertificate.cer (see Step 5)
--END CERTIFICATE--
--BEGIN CERTIFICATE--
  secondary intermediate certificate (see Step 6)
--END CERTIFICATE--
--BEGIN CERTIFICATE--
  primary intermediate certificate (see Step 6)
--END CERTIFICATE--
```

Save the resulting combined certificate text to a file named **`publickey.cer`** . This is the *public key certificate file* of your SSL certificate. It includes your public key certificate as well as the complete chain of trust in the form of the intermediate certificates that were used by the CA to sign your certificate. The public key certificate file will be installed on MobileTogether Server together with the private key (*see Step 8*).

8. Install SSL certificate on MobileTogether Server

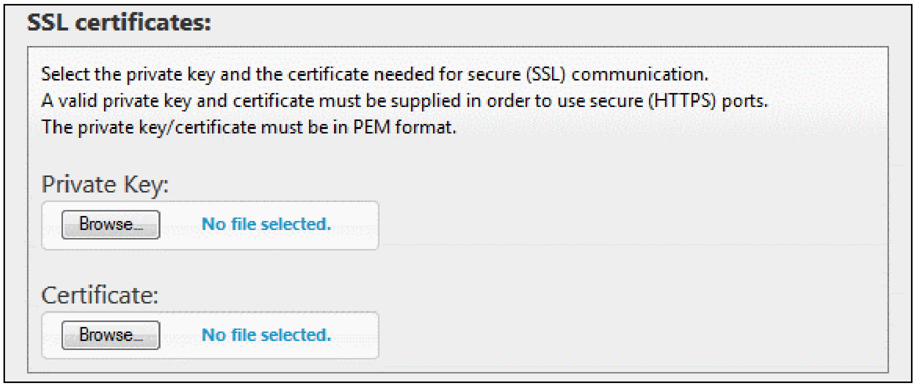
The SSL certificate is a set of certificates that are saved in the following files:

- `private.key`: Contains the private key certificate
- `publickey.cer`: Contains the public key certificate and the CA's intermediate

certificates (primary and secondary)

To install the SSL certificates on MobileTogether Server, do the following:

- Log in to the MobileTogether Server UI (by default on port `8085` of your server).
- Go to the Settings tab.
- Under SSL Certificates (*see screenshot below*), upload the two certificate files.



SSL certificates:

Select the private key and the certificate needed for secure (SSL) communication.
A valid private key and certificate must be supplied in order to use secure (HTTPS) ports.
The private key/certificate must be in PEM format.

Private Key:
No file selected.

Certificate:
No file selected.

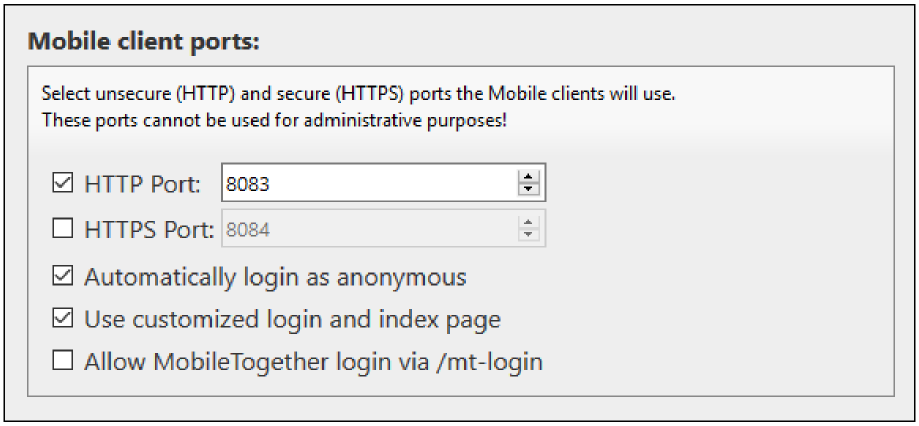
o For the private key, select **`private.key`** (*created in Step 1*)
o For the certificate, select **`publickey.cer`** (*created in Step 7*)

- Click **Save** at the bottom of the General Settings section to save your changes.

9. Set the server's HTTPS port

After installing the SSL certificate, you can specify a server port for SSL client communication. Do this as follows:

- Log in to the MobileTogether Server UI (by default on port `8085` of your server).
- Go to the Settings tab.
- Under Mobile Client Ports (*see screenshot below*), enable and specify the `HTTPS` port.



Mobile client ports:

Select unsecure (HTTP) and secure (HTTPS) ports the Mobile clients will use.
These ports cannot be used for administrative purposes!

☑ HTTP Port: 8083
☐ HTTPS Port: 8084
☑ Automatically login as anonymous
☑ Use customized login and index page
☐ Allow MobileTogether login via /mt-login

Make sure that any firewall is set up to allow access to MobileTogether Server through the `HTTPS` port.

10. Test SSL communication

    You can now use any SSL testing tool to check whether secure communication with your server via `HTTPS` is working properly. For example, this can be done with a browser, using the following SSL test site: https://ssltools.websecurity.symantec.com/checker/views/certCheck.jsp

    This will tell you: (i) whether the public key certificate file was properly constructed with the intermediate trust chain in Step 7, and (ii) whether your server can be reached properly through the firewall.

11. Enable MobileTogether Clients to use SSL

    In MobileTogether Client apps that communicate with an SSL-enabled MobileTogether Server, enable SSL communication by checking the *SSL Encryption* checkbox. See the MobileTogether Client documentation for information about how to find this check box.

Chapter 4

Server Procedures

4 Server Procedures

This section describes important server procedures. It assumes that MobileTogether Server has already been licensed. Note, however, that in order for MobileTogether Server to be accessed, both LicenseServer and MobileTogether Server must be started and running as services.

- Start Altova LicenseServer
- Start MobileTogether Server
- Set Up SSL Encryption
- Set Administrator and Mobile Client Ports
- Users and Roles
- Available Privileges
- Configure the Firewall
- Information for Clients
- How to Back Up and Restore MobileTogether Server

4.1 Start Altova LicenseServer

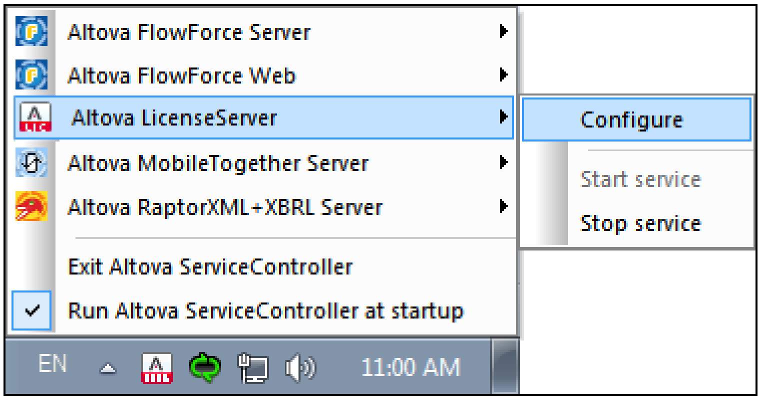
In order to run an installation of an Altova server product(i) FlowForce Server; (ii) RaptorXML(+XBRL) Server; (iii) MobileTogether Server; (iv) MapForce Server; (v) StyleVision Server, that installation must be licensed with an Altova LicenseServer on your network. LicenseServer must be running continuously as a service in order for all connected MobileTogether Server installations to run. Stopping LicenseServer will also stop all connected MobileTogether Server installations. If this happens, you will need to first restart LicenseServer, and then restart the stopped MobileTogether Server installation that you want to work with.

Start and stop LicenseServer as follows:

On Windows

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.


Altova FlowForce Server
Altova FlowForce Web
Altova LicenseServer
Altova MobileTogether Server
Altova RaptorXML+XBRL Server
Exit Altova ServiceController
✓ Run Altova ServiceController at startup
Configure
Start service
Stop service
EN
11:00 AM

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

To stop LicenseServer, select **Stop Service** from the LicenseServer submenu (*see screenshot above*).

On Linux

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

```
[Debian]:          sudo /etc/init.d/licenseserver start
[Ubuntu]:          sudo initctl 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 commands.)

On macOS

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
```

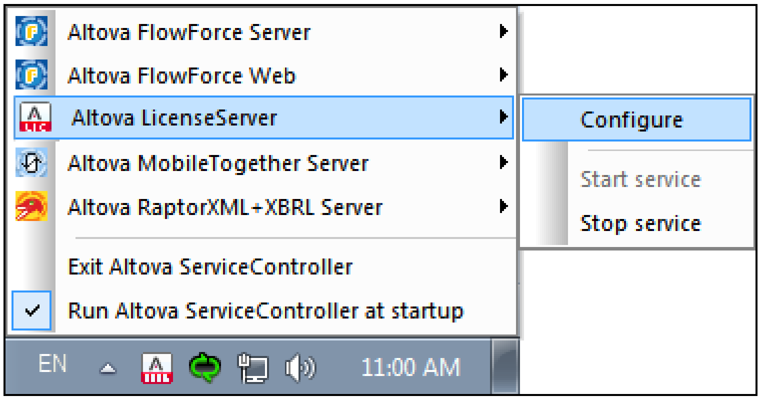
4.2 Start MobileTogether Server

In order to run MobileTogether Server, MobileTogether Server must be started as a service. Additionally, in order to use the Web UI of MobileTogether Server, it too must be started as a service. How to do this explained below.

On Windows

You can start MobileTogether Server 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.


Altova FlowForce Server
Altova FlowForce Web
Altova LicenseServer
Altova MobileTogether Server
Altova RaptorXML+XBRL Server
Exit Altova ServiceController
✓ Run Altova ServiceController at startup
Configure
Start service
Stop service
EN
11:00 AM

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

To stop MobileTogether Server, select Stop Service from the MobileTogether Server submenu (*see screenshot above*).

On Linux

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

```
[Debian]:          sudo /etc/init.d/mobiletogetherserver start
[Ubuntu]:          sudo initctl start mobiletogetherserver
[CentOS 6]:        sudo initctl start mobiletogetherserver
[CentOS 7]:        sudo systemctl start mobiletogetherserver
[RedHat]:          sudo initctl start mobiletogetherserver
```

(If you need to stop MobileTogether Server, replace **`start`** with **`stop`** in the above commands.)

On macOS

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

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

If at any time you need to stop MobileTogether Server, use:

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

4.3 Set Up SSL Encryption

If you require that communications between your MobileTogether Server and MobileTogether Client devices are encrypted using the SSL protocol, you will need to:

- Generate an SSL private key and create an SSL public key certificate file
- Set up MobileTogether Server for SSL communication.

The steps to do this are listed below.

MobileTogether uses the open-source OpenSSL toolkit to manage SSL encryption. The steps listed below, therefore, need to be carried out on a computer on which OpenSSL is available. OpenSSL typically comes pre-installed on most Linux distributions and on macOS machines. It can also be installed on Windows computers. For download links to installer binaries, see the OpenSSL Wiki.

---

1. Generate a private key

   SSL requires that a **private key** is installed on MobileTogether Server. This private key will be used to encrypt all data sent to MobileTogether Client apps. To create the private key, use the following OpenSSL command:

   ```
   openssl genrsa -out private.key 2048
   ```

   This creates a file called **`private.key`**, which contains your private key. Note where you save the file. You will need the private key to (i) generate the Certificate Signing Request (CSR), and (ii) to be installed on MobileTogether Server (*see Step 8 below*).

2. Certificate Signing Requests (CSRs)

   A Certificate Signing Request (CSR) is sent to a certificate authority (CA), such as VeriSign or Thawte, to request a public key certificate. The CSR is based on your private key and contains information about your organization. Create a CSR with the following OpenSSL command (which provides the private-key file, `private.key`, that was created in Step 1, as one of its parameters):

   ```
   openssl req -new -nodes -key private.key -out my.csr
   ```

   During generation of the CSR you will need to give information about your organization, such as that listed below. This information will be used by the certificate authority to verify your company's identity.

   - *Country*
   - *Locality* (the city where your business is located)
   - *Organization* (your company name). Do not use special characters; these will invalidate your certificate
   - *Common Name* (the DNS name of your server). This must exactly match your server's official name, that is, the DNS name client apps will use to connect to the server
   - *A challenge password.* Keep this entry blank!

3. Buy an SSL certificate

   Purchase an SSL certificate from a recognized certificate authority (CA), such as VeriSign or Thawte. For the rest of these instructions, we follow the VeriSign procedure. The procedure with other CAs is similar.

   - Go to the VeriSign website.
   - Click **Buy SSL Certificates**.
   - Different types of SSL certificates are available. For MobileTogether Server, Secure Site or Secure Site Pro certificates are sufficient. EV (extended verification) is not necessary, since there is no "green address bar" for users to see.
   - Proceed through the sign-up process, and fill in the information required to place your order.
   - When prompted for the CSR (*created in Step 2*), copy and paste the content of the **`my.csr`** file into the order form.
   - Pay for the certificate with your credit card.

   > **Allow time for obtaining a certificate**
   > Obtaining public key certificates from an SSL certificate authority (CA) typically takes **two to three business days**. Please take this into account when setting up your MobileTogether Server.

4. Receive public key from CA

   Your certificate authority will complete the enrollment process over the next two to three business days. During this time you might get emails or phone calls to check whether you are authorized to request an SSL certificate for your DNS domain. Please work with the authority to complete the process.

   After the authorization and enrollment process has been completed, you will get an email containing the **public key** of your SSL certificate. The public key will be in plain text form or attached as a **`.cer`** file.

5. Save public key to file

   For use with MobileTogether Server, the public key must be saved in a **`.cer`** file. If the public key was supplied as text, copy-paste all the lines from

   ```
   --BEGIN CERTIFICATE--
     ...
   --END CERTIFICATE--
   ```

   into a text file that we will call **`mycertificate.cer`**.

6. Save CA's intermediate certificates to file

   To complete your SSL certificate, you will need two additional certificates: the **primary** and **secondary intermediate certificates**. Your certificate authority (CA) will list content of intermediate certificates on its website.

- Verisign's intermediate certificates: https://knowledge.verisign.com/support/ssl-certificates-support/index?page=content&id=AR657&actp=LIST&viewlocale=en_US
- Verisign's intermediate certificates for its Secure Site product: https://knowledge.verisign.com/support/ssl-certificates-support/index?page=content&id=AR1735

Copy-paste both intermediate certificates (primary and secondary) into separate text files and save them on your computer.

7. Combine certificates in one public key certificate file

You now have three certificate files:

- Public key (`mycertificate.cer`)
- Secondary intermediate certificate
- Primary intermediate certificate

Each contains text blocks bracketed by lines that look like this:

```
--BEGIN CERTIFICATE--
  ...
--END CERTIFICATE--
```

Now copy-paste all three certificates into one file so that they are in sequence. The order of the sequence is important: (i) public key, (ii) secondary intermediate certificate, (iii) primary intermediate certificate. Ensure that there are no lines between certificates.

```
--BEGIN CERTIFICATE--
  public key from mycertificate.cer (see Step 5)
--END CERTIFICATE--
--BEGIN CERTIFICATE--
  secondary intermediate certificate (see Step 6)
--END CERTIFICATE--
--BEGIN CERTIFICATE--
  primary intermediate certificate (see Step 6)
--END CERTIFICATE--
```

Save the resulting combined certificate text to a file named **`publickey.cer`** . This is the *public key certificate file* of your SSL certificate. It includes your public key certificate as well as the complete chain of trust in the form of the intermediate certificates that were used by the CA to sign your certificate. The public key certificate file will be installed on MobileTogether Server together with the private key (*see Step 8*).

8. Install SSL certificate on MobileTogether Server

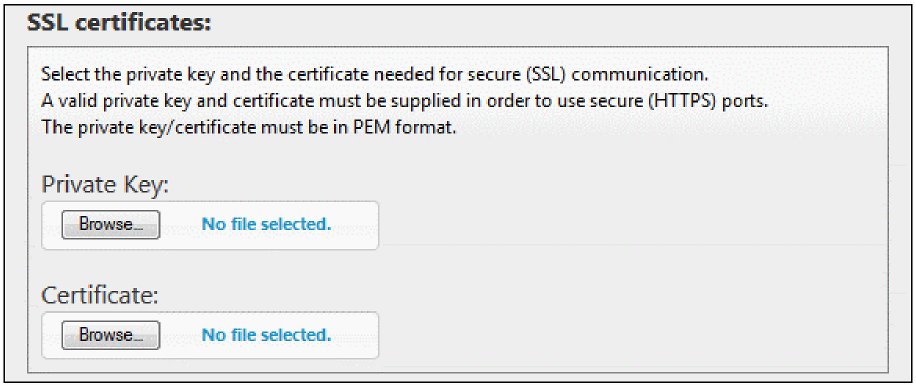
The SSL certificate is a set of certificates that are saved in the following files:

- `private.key`: Contains the private key certificate
- `publickey.cer`: Contains the public key certificate and the CA's intermediate

certificates (primary and secondary)

To install the SSL certificates on MobileTogether Server, do the following:

- Log in to the MobileTogether Server UI (by default on port `8085` of your server).
- Go to the Settings tab.
- Under SSL Certificates (*see screenshot below*), upload the two certificate files.



SSL certificates:

Select the private key and the certificate needed for secure (SSL) communication.
A valid private key and certificate must be supplied in order to use secure (HTTPS) ports.
The private key/certificate must be in PEM format.

Private Key: Browse... No file selected.

Certificate: Browse... No file selected.

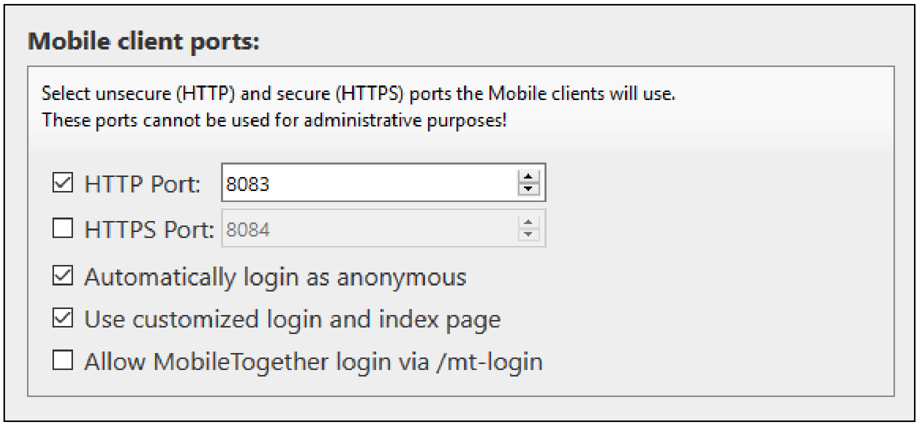
o For the private key, select **`private.key`** (*created in Step 1*)
o For the certificate, select **`publickey.cer`** (*created in Step 7*)

- Click **Save** at the bottom of the General Settings section to save your changes.

9. Set the server's HTTPS port

After installing the SSL certificate, you can specify a server port for SSL client communication. Do this as follows:

- Log in to the MobileTogether Server UI (by default on port `8085` of your server).
- Go to the Settings tab.
- Under Mobile Client Ports (*see screenshot below*), enable and specify the `HTTPS` port.



Mobile client ports:

Select unsecure (HTTP) and secure (HTTPS) ports the Mobile clients will use.
These ports cannot be used for administrative purposes!

☑ HTTP Port: 8083
☐ HTTPS Port: 8084
☑ Automatically login as anonymous
☑ Use customized login and index page
☐ Allow MobileTogether login via /mt-login

Make sure that any firewall is set up to allow access to MobileTogether Server through the `HTTPS` port.

10. Test SSL communication

    You can now use any SSL testing tool to check whether secure communication with your server via `HTTPS` is working properly. For example, this can be done with a browser, using the following SSL test site: https://ssltools.websecurity.symantec.com/checker/views/certCheck.jsp

    This will tell you: (i) whether the public key certificate file was properly constructed with the intermediate trust chain in Step 7, and (ii) whether your server can be reached properly through the firewall.

11. Enable MobileTogether Clients to use SSL

    In MobileTogether Client apps that communicate with an SSL-enabled MobileTogether Server, enable SSL communication by checking the *SSL Encryption* checkbox. See the MobileTogether Client documentation for information about how to find this check box.

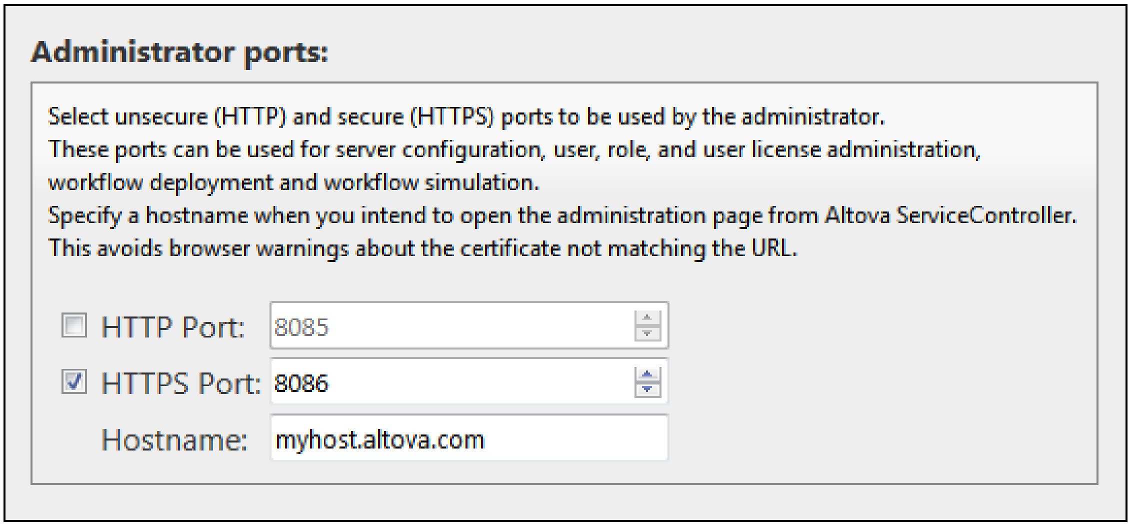
4.4 Set Administrator and Mobile Client Ports

The administrator ports are used to connect to the Web UI of MobileTogether Server, while the mobile client ports are those the mobile client device uses to connect to the services of MobileTogether Server.

Set the administrator ports

The administrator ports provide access for the following purposes:

- To connect to the server's Web UI and perform administrative functions, such as setting up Users and Roles.
- To deploy MobileTogether designs (as MobileTogether solutions) to the server. MobileTogether Designer has a setting that specifies the address and port of the MobileTogether Server to which to deploy designs.



Administrator ports:

Select unsecure (HTTP) and secure (HTTPS) ports to be used by the administrator.
These ports can be used for server configuration, user, role, and user license administration, workflow deployment and workflow simulation.
Specify a hostname when you intend to open the administration page from Altova ServiceController.
This avoids browser warnings about the certificate not matching the URL.

☐ HTTP Port: 8085

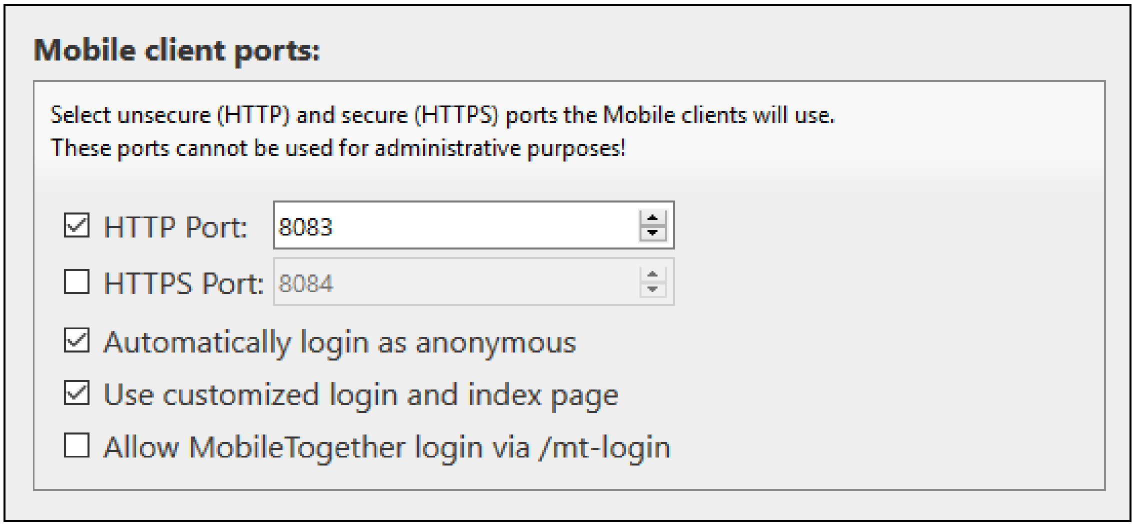
☑ HTTPS Port: 8086

Hostname: myhost.altova.com

The `HTTP` port is the unsecure port; `HTTPS` is the secure port. To use `HTTPS`, you need to set up SSL Encryption. If you set up the HTTPS port and wish to avoid browser warnings about the SSL certificate not matching the URL, then specify the hostname of the computer on which the MobileTogether Server configuration page will be opened.

Set the mobile client ports

The ports that mobile devices will use to connect to the server. The `HTTP` port is the unsecure port; `HTTPS` is the secure port. To use `HTTPS`, you need to set up SSL Encryption.



Mobile client ports:

Select unsecure (HTTP) and secure (HTTPS) ports the Mobile clients will use.
These ports cannot be used for administrative purposes!

☑ HTTP Port: 8083

☐ HTTPS Port: 8084

☑ Automatically login as anonymous

☑ Use customized login and index page

☐ Allow MobileTogether login via /mt-login

Automatically login as anonymous
If selected, clients will be logged in automatically as anonymous. The login page is skipped, and the server's first page is shown directly. The first page is either the standard page that displays the root folder, or it is a custom page that you have defined (*see next point*). If this option is **not** selected, the client will need to login with the appropriate credentials via the default login page. If anonymous login is selected, then remember to set the relevant privileges for anonymous.

Use customized login and index pages
Select this option if a customized login page and first page should be used. This enables you to design your own entry point for clients. Set up the customized pages as follows:

1. Create the two pages as HTML pages, and name them **`login.html`** and **`index.html`**, respectively.
2. Save the two files in the **`index`** folder that is located in the MobileTogether Server ***application data folder*** (*see table below*). Additional files, such as image files and CSS files, are best saved in a subfolder of the **`index`** folder (for instance in one that is called, say, **`static`**).

Linux	`/var/opt/Altova/MobileTogetherServer`
Mac	`/var/Altova/MobileTogetherServer`
Windows Vista, Windows 7, 8, 10	`C:\ProgramData\Altova\MobileTogetherServer`

The code listings of a sample login page and sample first (index) page are given below. These listings are basic, but you can modify the code as you like.

⊟ *login.html*

```
<html>
  <header>
    <title>Customized Login</title>
  </header>
  <head>
    <meta http-equiv="Cache-Control" content="no-store" />
  </head>
  <body>
    <div>
      <h1>Sign in</h1>
      <p>A bare-basics custom page for client logins to MobileTogether
Server. Modify this page as required, and use the Static sub-folder to
save CSS stylesheets, images, etc.</p>
      <form method="post" action="/do_login" name="loginform">
        <table>
          <tbody>
            <!-- The user to login -->
            <tr><td>Username:</td></tr>
            <tr><td><input type="text" name="username" size="30"></td></
tr>
            <!-- The password of the user -->
            <tr><td>Password:</td></tr>
            <tr><td><input type="password" name="password" size="30"></
td></tr>
            <!-- The Active Directory domian details -->
            <tr><td>&nbsp;</td></tr>
            <tr><td>Active Directory Login:</td></tr>
            <tr>Domain suffix: <td><input type="providernamesuffix"
name="providernamesuffix" value=""></td></tr>
            <tr>Domain prefix: <td><input type="providernameprefix"
name="providernameprefix" value=""></td></tr>
            <!-- The Sign-In button -->
            <tr><td><input type="submit" value="Sign in"></td></tr>
          </tbody>
        </table>
        <!-- The page to redirect to after a successful login. -->
        <input type="hidden" name="from_page" value="/index"></
input><br>
      </form>
    </div>
  </body>
</html>
```

⊟ *index.html*

```
<html>
  <header>
    <title>Custom Index</title>
  </header>
  <head>
    <meta http-equiv="Cache-Control" content="no-store" />
    <title>Custom Index</title>
  </head>
  <body>
    <img src="/index/static/logo.png"></img><hr/>
    <a href="/do_logout">Logout</a>
    <p>MobileTogether Custom Login</p>
    <div><a href='/run?d=/public/About'>Start the About app</a></div>
    <div><a href='/run?d=/public/DateCalc'>Start the Date Calculator
app</a></div>
    <div><a href='/run?d=/public/WorldPopulation'>Start the World
Population Statics app</a></div>
  </body>
</html>
```

Allow MobileTogether login via /mt-login

This option specifies that the login will be via the default login page and first page—and not via the customized login and index pages. This allows you to store the **`login.html`** and **`index.html`** files at the designated location, but still use the default pages. Note that the client's browser or browser settings might require that the browser cache is emptied in order for this setting to take effect.

4.5 Users and Roles

A user account is defined by a log-in name and password, and has a set of access rights associated with it. Users access MobileTogether Server for administrative purposes or as client end users.

Access rights are determined by the privileges a user is granted. A user receives privileges in the following ways: (i) privileges inherited from roles the user is a member of, (ii) privileges assigned directly to the user. A role is defined by a set of privileges. A role is either assigned privileges directly and/or inherits the privileges of another role that it is a member of. Privileges themselves are access rights to the various administrative functions and services of MobileTogether Server. Examples of privileges are: the right to manage server settings, to set a user's own password, to run simulations on the server.

Through the use of roles, user privileges can be defined in a hierarchical way. For example, the role of `SimpleAdmin` role could allow the privilege, *Manage server settings*. If `AdvancedAdmin` is a member of `SimpleAdmin`, it inherits the management of server settings, and could additionally be assigned the privilege, *Maintain users, roles and privileges*. The hierarchical chain can the be further extended. For a list of privileges, see Available Privileges.

▼ About Users

A user is defined by a name-and-password combination. Users access MobileTogether Server in two ways:

- *Web UI access:* The Web UI is the administrative interface of MobileTogether Server. Logging in to the Web UI requires a name-and-password combination; it is therefore done as a user.
- *Service interface:* The HTTP service interface exposes MobileTogether Server services, typically to the MobileTogether Client app on a mobile device. A user accesses the service interface, using a name-and-password combination. The services exposed relate typically to access to MobileTogether solutions and their related data.

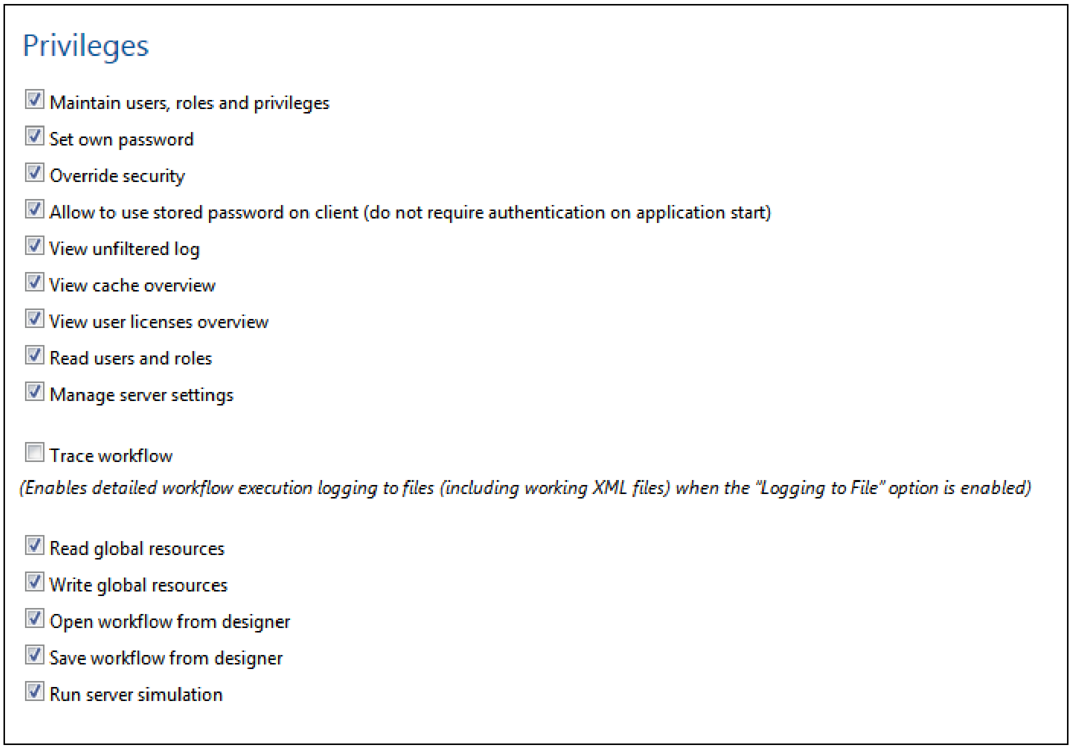
Two special users are predefined:

`root`	**`root`** is the initial administrator user. It is initially the most powerful user, having all privileges and having the ability to add other users and to set roles. Its initial name-password combination is: **`root-root`**. The password can be changed at any time.
`anonymous`	**`anonymous`** is an account for anonymous users that access services exposed via the HTTP service interface. It cannot be used for logging in top the Web UI, and it has no initial password.

▼ About Privileges

A privilege is an activity that a user is allowed to carry out. There is a fixed number of MobileTogether Server privileges, and a user can be assigned zero to all of the available privileges. It is, however, good practice to assign privileges via roles (*see next section*), rather than to assign privileges directly to the user. The assigning of privileges and roles to a user is done by a user that has been assigned this privilege. Initially, it is **`root`** user that has this privilege.

The screenshot below shows all the available privileges.



Privileges

☑ Maintain users, roles and privileges

☑ Set own password

☑ Override security

☑ Allow to use stored password on client (do not require authentication on application start)

☑ View unfiltered log

☑ View cache overview

☑ View user licenses overview

☑ Read users and roles

☑ Manage server settings

□ Trace workflow

(Enables detailed workflow execution logging to files (including working XML files) when the "Logging to File" option is enabled)

☑ Read global resources

☑ Write global resources

☑ Open workflow from designer

☑ Save workflow from designer

☑ Run server simulation

The tab *Users and Roles | Reports | Privileges Report* provides a list of all privileges, with each privilege being listed together with all the users/roles that have that privilege.

▼ About Roles

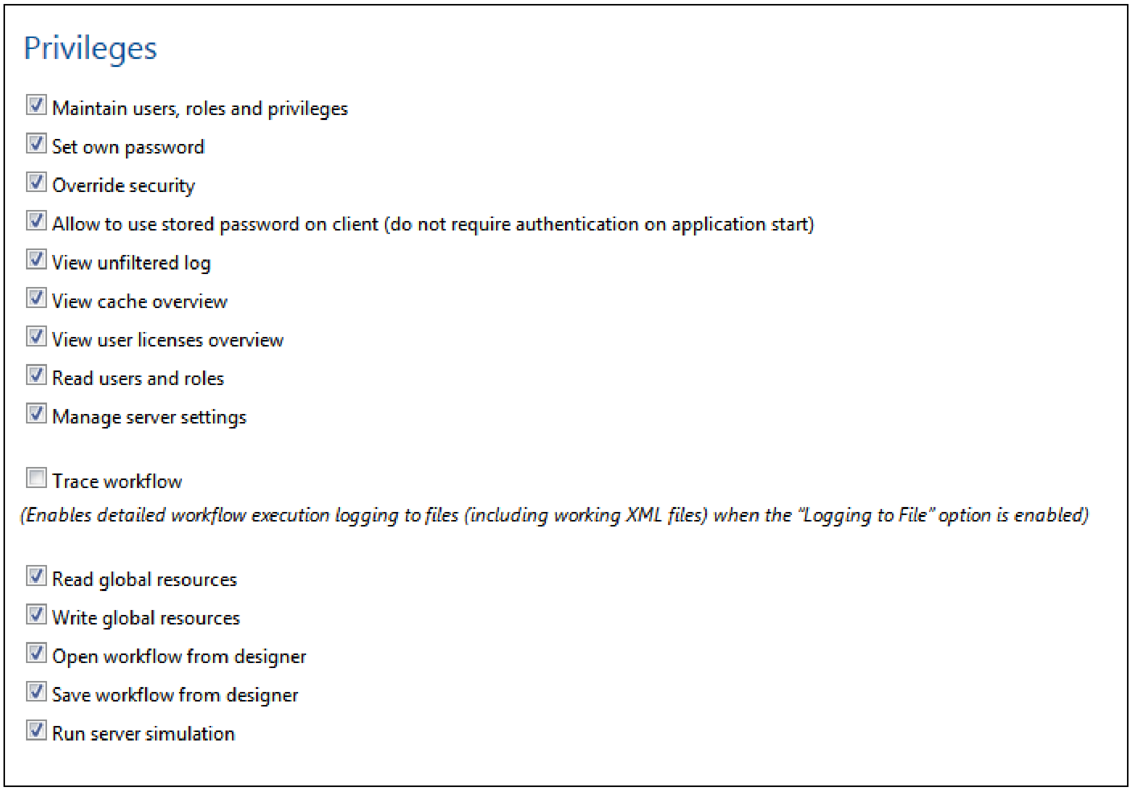
A role defines a set of privileges. It can be assigned to another role or to a user. A role's privileges automatically become the privileges of any other role or any user that the role is assigned to. A user can be assigned any number of roles. As a result, a user will have all the privileges defined in the multiple assigned roles.

The following roles are predefined:

- **`authenticated`** is automatically assigned to every user **except `anonymous`**. So user with a name-and-password is assigned the `authenticated` role.
- **`all`** is automatically assigned to every user **including `anonymous`**.
- **`workflow-designer`** is assigned to users that design workflows in MobileTogether Designer. This role allows a user to open and save workflows, as well as to run a simulation on the server.
- **`workflow-user`** is assigned to users running the workflow on a mobile device. This role allows the user to access the service interface without needing to log in to the server and start the solution on the client.

4.6 Available Privileges

Privileges themselves are access rights to the various administrative functions and services of MobileTogether Server. When a user logs in to MobileTogether Server (either via its Web UI or services interface), the user's access rights are determined by the user's privileges. Privileges are assigned to the user either directly or via roles, in the *Users and Roles* tab.



Privileges

☑ Maintain users, roles and privileges

☑ Set own password

☑ Override security

☑ Allow to use stored password on client (do not require authentication on application start)

☑ View unfiltered log

☑ View cache overview

☑ View user licenses overview

☑ Read users and roles

☑ Manage server settings

☐ Trace workflow

(Enables detailed workflow execution logging to files (including working XML files) when the "Logging to File" option is enabled)

☑ Read global resources

☑ Write global resources

☑ Open workflow from designer

☑ Save workflow from designer

☑ Run server simulation

The available privileges are described below.

⊟ *Maintain users, roles and privileges*

Any user having this privilege can create, delete and edit users and roles, their privilege assignments and passwords. This is an administrative privilege and should only be assigned to MobileTogether administrators. By default, only the user "root" possesses this privilege.

⊟ *Set own password*

Any user having this privilege can change his own password. Users who do not have this privilege need to have their password set by a MobileTogether administrator. By default the "authenticated" role, and hence every user account except "anonymous", possesses this privilege.

⊟ *Override security*

Any user having this privilege can change permissions in the container hierarchy without needing "write" security permission. This allows MobileTogether administrators to regain access to resources accidentally rendered inaccessible. This is an administrative privilege and should only be assigned to MobileTogether administrators. By default, only "root" possesses this privilege.

Allow use of stored password on client

Allows the user to use the password stored on the client. User does not need authentication.

View unfiltered log

By default users can only see log entries related to Configurations they have "read" access to. By granting this privilege a user can read all log entries, including those not associated with a specific configuration. By default, only "root" possesses this privilege.

View cache overview

Allows the user to view the cache overview on the server.

View user licenses overview

Allows the user to see the licenses overview on the server.

Read users and roles

By default users will only see their own user account and any roles they are member of. By granting this privilege a user can read all defined users and roles. By default, only "root" possesses this privilege.

Manage server settings

Allows the user to edit server settings.

Trace workflow

Allows detailed workflow execution logging to files, if the "logging to file directory" option is enabled in the Logging group of the Settings dialog box.

Read global resources

Allows the user with this privilege to read the global resource alias/configuration from the server.

Write global resources

Allows the user with this privilege to write/save the global resource alias/configuration to the server.

Open workflow from designer

Allows a user to open a deployed MobileTogether design file from the server. The host login details are supplied by selecting the menu option File | Open from MobileTogether server.

Save workflow from designer

Allows a user to save/deploy a MobileTogether design file to the server. The host login details are supplied by selecting the menu option File | Save to MobileTogether server.

⊟ *Run server simulation*

Allows the user having this privilege to run a simulation from the browser (and preview the result). Note the Back browser button returns you to the container view.

4.7 Configure the Firewall

Server IP address and network firewall settings

Your server can have a public IP address (accessible over the Internet) and/or a private IP address (accessible within a private network; for example, via WiFi within a company network). If a mobile client device tries to connect via the Internet using the server's private IP address, then the connection will not work. This is because the private IP address is not known on the Internet and cannot be resolved. If a client device uses a private IP address, then the client device must already have access to the private network.

To ensure that the server can be accessed, do one of the following:

- Provide the server with a public IP address so that it can be reached via the Internet. On the client device, use this public IP address to access the server.
- If you use a firewall and install MobileTogether Server on a server with a private IP address (inside the private network), then use the network firewall to forward requests sent to a public IP-address/port-combination to your MobileTogether Server server. On the client device, use the public IP address.

You must also ensure that the firewall is configured to allow access to the server port used for MobileTogether Client communication. The ports used by MobileTogether Server are specified in the Settings page of the the Web UI of MobileTogether Server (*see the MobileTogether Server user manual*). On the client device, this is the port that must be specified as the server port to access.

Tip: Port 80 is usually open on most firewalls by default. So, if you are having difficulties with firewall settings and if port 80 is not already bound to some other service, you could specify port 80 as the MobileTogether Server port for client communication.

4.8 Information for Clients

The MobileTogether Client app on mobile devices will need to connect to MobileTogether Server. The following server information will be required by the MobileTogether Client app.

IP Address	The IP address of MobileTogether Server
Port	The `HTTP` or `HTTPS` port, which is specified in the *Mobile Client Ports* setting
SSL	Whether communication should be SSL encrypted or not
User name	As what user to log in. This will determine access rights. See Users and Roles
Password	The password of the user account

Note: Data that is saved on the web client is saved in the local storage (aka web storage) of your browser. HTML 5.0 local storage is supported in the following browsers:

IE 8.0 +	Firefox 3.5+	Safari 4.0+	Chrome 4.0+	Opera 10.5+	iPhone 2.0+	Android 2.0+

Updating server settings on client devices

In order for a client device to run a solution, the server's access settings must be configured on that device. If the server settings change—for example, if the MobileTogether Server is moved to another machine that has a different IP address—then the server settings on client devices must be modified accordingly. In MobileTogether Designer, you can use the MobileTogether function **`mt-server-config-url`** to generate a URL that contains the new server settings and looks something like this: **`mobiletogether://mt/change-settings?settings=<json encoded settings>`**. This URL can be sent as an email link to client users. When the link is tapped, server settings on the client are automatically updated. See the MobileTogether Designer user manual for more information about generating this URL.

4.9 How to Back Up and Restore MobileTogether Server

This section describes how to back up and restore MobileTogether Server.

- Backing up MobileTogether Server consists of copying essential application data files and data files to a safe location.
- Restoring MobileTogether Server consists of copying the backed-up files into place on the new installation.
- Updating client connections to the server.

These procedures are described below.

Backing up MobileTogether Server

Before starting a backup, MobileTogether Server must be stopped. (This is necessary in order to avoid inconsistency between the DB status of live files and backup files.) The MobileTogether Server files that need to be backed up are located by default in the application data folder (*see below*). The `.cfg` configuration file can be edited with a text editor, as an alternative to changing settings via the Web Administration Interface or the Command Line Interface.

The location of the ***application data folder*** depends on the operating system and platform, and, by default, is as follows.

Linux	`/var/opt/Altova/MobileTogetherServer`
Mac	`/var/Altova/MobileTogetherServer`
Windows Vista, Windows 7, 8, 10	`C:\ProgramData\Altova\MobileTogetherServer`

The following table lists the main files and folders in the application data folder.

`cache`	Default directory for caches of solutions. If a cache is not available, it will be recreated automatically at runtime.
`logs`	Default directory for log files that are created when the *Logging to file* option is enabled and for general MobileTogether Server logs.
`SolutionFiles`	Default directory for XML or image files referenced from deployed solutions.
`cert.pem`	PEM file with certificate needed for secure (SSL) communication.
`key.pem`	PEM file with private key needed for secure (SSL) communication.
`mobiletogether.db`	This is the main database file (SQLite) that stores the MobileTogether Server object system, user data, deployed solutions, files, and others.
`mobiletogetherlog.db`	This is the database file (SQLite) that stores the MobileTogether Server logs.

`mobiletogetherserver.cfg`	Stores global configuration settings of MobileTogether Server (port number, solutions directory, etc)
`mobiletogetherserver.licsid`	File with ID of registered LicenceServer client.
`mobiletogetherserver.licsvr`	Contains the address of the LicenseServer and failover if one is specified.

Note: Before the installation of every new version of MobileTogether Server, the files and folders listed above are, by default, copied to a backup folder located in the application data folder (*see table above*). The name of each backup folder contains the backup date and time. If you wish to disable automatic backup before the next installation, do this in the Upgrade Settings section of the Settings tab.

Restoring MobileTogether Server

To restore a previous configuration of MobileTogether Server from backup files (*see above*), do the following:

1. Install the same version of MobileTogether Server as that from which you backed up (*see above*).
2. Stop MobileTogether Server.
3. Copy the backed-up files (*see above*) into the correct folders on the new installation.
4. Start MobileTogether Server.

Updating client connections to the server

If you have moved MobileTogether Server to another machine (with new settings, such as a new IP address), client device settings to connect to MobileTogether Server must be updated. See Information for Clients for details.

4.10 Frequently Asked Questions

▾ *There are several workflows on our server. A new solution has been added that uses an ADO connection to an IBM DB2 database. We have noticed that from the time a client tries to access this solution, MobileTogether Server crashes. Deleting the workflow does not help. The problem disappears only when the server is re-started. But it reappears every time a client accesses this solution. Do you know anything about this?*

Yes, this is a known problem and is related to the databases in question. Workflows that contain ADO connections to IBM DB2 or Informix databases trigger this crash when the workflow asks the server to access the database for the first time. The reason that the problem persists even after deleting the solution appears to be that some connection data is saved in the server's memory. This connection data is deleted only when the server is re-started.

Chapter 5

Web UI Reference

5 Web UI Reference

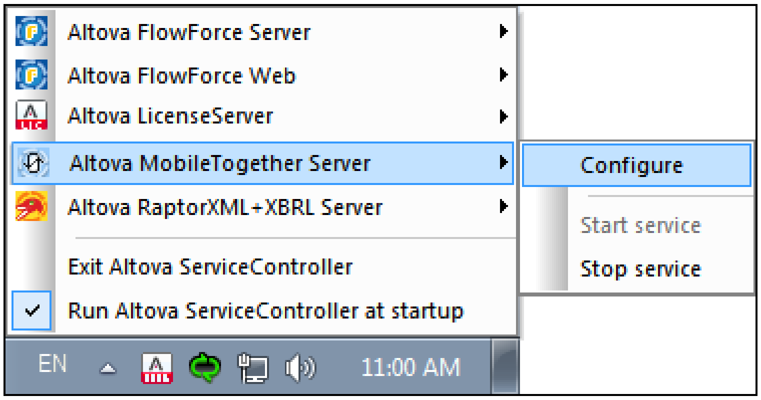
MobileTogether Server has a **Web User Interface** (Web UI) with which you can easily configure MobileTogether Server. The Web UI can be opened in any Internet browser on any of the supported operating systems.

Accessing the Web UI of MobileTogether Server

The MobileTogether Server Web UI is accessed as follows:

On Windows

To access the Web UI, click the **ServiceController** icon in the system tray (*see screenshot below*), hover over **Altova MobileTogether Server** in the menu that pops up, and then select **Configure** from the MobileTogether Server submenu. If MobileTogether Server is not already running, use the *Start Service* option to start MobileTogether Server.


Altova FlowForce Server
Altova FlowForce Web
Altova LicenseServer
Altova MobileTogether Server
Altova RaptorXML+XBRL Server
Exit Altova ServiceController
Run Altova ServiceController at startup
Configure
Start service
Stop service
EN
11:00 AM

To sign in, enter the username and password. The default `username/password` is `root/root`. If Active Directory Login via one or more domains has been defined, then a *Login* combo box is available in which you can: (i) select from among the defined domains, or (ii) choose to login directly (not via a domain).

On Linux

To access the Web UI, enter its URL in the address bar of a browser and press **Enter**. By default, the URL of the Web UI page (for administrative access) is:

`http://<serverIPAddressOrName>:8085/`

To sign in, enter the username and password. The default `username/password` is `root/root`. If Active Directory Login via one or more domains has been defined, then a *Login* combo box is available in which you can: (i) select from among the defined domains, or (ii) choose to login directly (not via a domain).

On Mac OS X

To access the Web UI, enter its URL in the address bar of a browser and press **Enter**. By default, the URL of the Web UI page (for administrative access) is:

`http://<serverIPAddressOrName>:8085/`

To sign in, enter the username and password. The default `username/password` is `root/root`. If Active Directory Login via one or more domains has been defined, then a *Login* combo box is available in which you can: (i) select from among the defined domains, or (ii) choose to login directly (not via a domain).

Web UI tabs

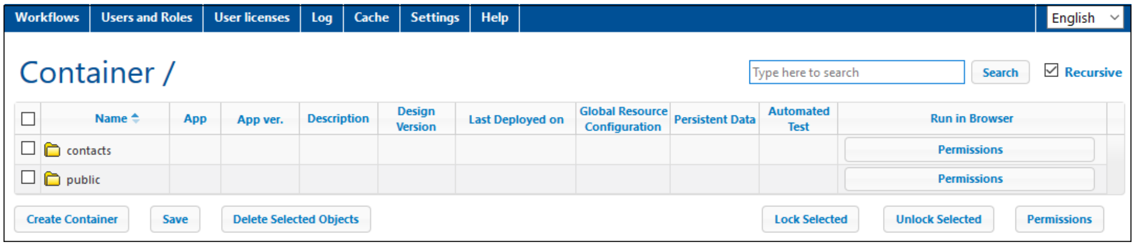
The Web UI is the administrator interface of MobileTogether Server. The various administrative functions are available in the following Web UI tabs:

- Workflows: An interface for managing the server's container structure and container permissions.
- Users and Roles: To set up user accounts and roles, and the privileges associated with users and roles. The access rights of users are defined in this tab.
- User licenses: Shows the mobile devices that are currently licensed, and their license details.
- Log: Displays the logged server actions according to the selected filters.
- Cache: This tab shows the details of caches that are currently available on the server. Caches can also be activated/deactivated and deleted.
- Settings: MobileTogether Server settings, such as access ports, log settings, and server session timeouts, are defined in this tab.

5.1 Workflows

The **Workflows** tab (*screenshot below*) provides an interface for managing the container structure of the `root` folder of MobileTogether Server and the access rights (permissions) for each container. Containers are folders that contain sub-containers and/or solutions (aka design files or `.mtd` files). MTD files cannot be added to a container via the server's Web UI, but are deployed to the server from MobileTogether Designer. At deployment, the exact path to a container must be specified; this is facilitated by being able to browse, in MobileTogether Designer, to the required container.

The *Workflows* tab initially displays the **`root`** container, which is denoted by the `"/"` character.



□	Name	App	App ver.	Description	Design Version	Last Deployed on	Global Resource Configuration	Persistent Data	Automated Test	Run in Browser
□	contacts									Permissions
□	public									Permissions

Functionality

The buttons of the tab provide the following functionality:

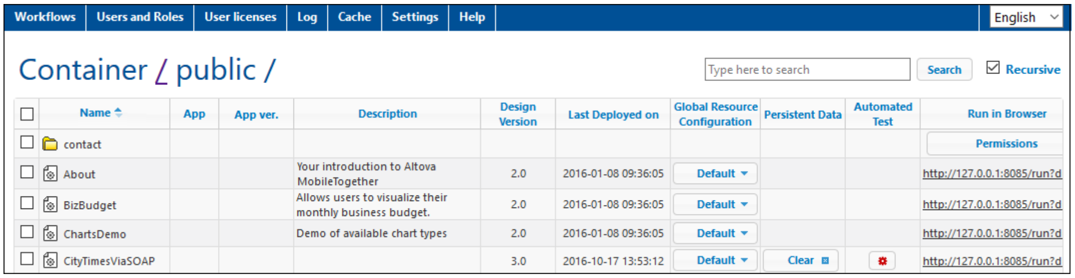
Create Container	Creates a container in the current folder. Click a container to open it
Save	Saves changes such as a modification of descriptions
Delete Selected	Deletes the selected container or file
Lock Selected	A locked solution cannot be overwritten by a new deployment. If attempted, an error message is displayed in MobileTogether Designer
Unlock Selected	Unlocks a locked solution
Permissions	Sets which users/roles can access individual containers, and their level of access
Search	Searches for the term submitted. Check *Recursive* to search in descendant containers

Other available actions:

- To navigate up the container hierarchy, click the required ancestor folder in the path at the top of the *Workflows* tab
- To navigate down the container hierarchy, click a container to open it
- Click a solution file's URL to run the solution

The container /public/

Clicking the **`public`** container opens the container and displays its contents. `public` is a predefined container containing sample design files (solutions) that are delivered with the program. Click a solution's URL to run it.



Container / public /

□	Name	App	App ver.	Description	Design Version	Last Deployed on	Global Resource Configuration	Persistent Data	Automated Test	Run in Browser
□	contact									Permissions
□	About			Your introduction to Altova MobileTogether	2.0	2016-01-08 09:36:05	Default			http://127.0.0.1:8085/run?d
□	BizBudget			Allows users to visualize their monthly business budget.	2.0	2016-01-08 09:36:05	Default			http://127.0.0.1:8085/run?d
□	ChartsDemo			Demo of available chart types	2.0	2016-01-08 09:36:05	Default			http://127.0.0.1:8085/run?d
□	CityTimesViaSOAP				3.0	2016-10-17 13:53:12	Default	Clear	✱	http://127.0.0.1:8085/run?d

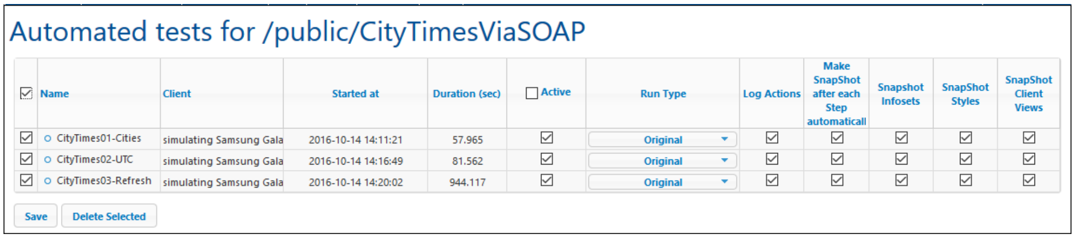
The display of a container's contents

A container contains sub-containers and/or solutions (aka design files or `.mtd` files). The contents of each container are displayed as a tabular list. The columns of the table display the properties of solutions:

- *Name:* Name of the solution file as saved in MobileTogether Designer.
- *App, App version:* The *App* and *App Ver* columns appear only if at least one AppStore App (*see the MobileTogether Designer user manual*) has been deployed to the server. They display, respectively, the name of the AppStore App and its version.
- *Description:* Short description of the solution, which can be entered by clicking in the field.
- *Design Version:* Version of MobileTogether Designer in which the solution was created.
- *Last Deployed On:* The date and time of the solution's last deployment
- *Global Resource Configuration*: The global resource that has been defined for that solution and deployed to the server. If no global resource is specified, `Default` is displayed
- *Persistent Data:* A **Clear Data** button appears in this column if data has been changed while running the solution. Click the button if you wish to undo the changes
- *Automated Test:* A blue wheel indicates that at least one test run for automated testing of that solution is available, but is not active. A red wheel indicates that at least one test run of the available test runs is active. To activate a design's test run or configure how the test run is played back on the client, click the solution's wheel icon (*shown in the screenshot above*). This displays a page showing the automated tests of that solution (*see next section below*). For information about Automated Testing, see the MobileTogether Designer documentation.
- *Run in Browser:* The server URL where the solution file is deployed. Click to run the solution

Automated testing

When you click the wheel icon in a solution's Automated Test column, a page is displayed that shows the automated tests of that solution (*screenshot below*).



Automated tests for /public/CityTimesViaSOAP

☑	Name	Client	Started at	Duration (sec)	□ Active	Run Type	Log Actions	Make SnapShot after each Step automaticall	Snapshot Infosets	SnapShot Styles	SnapShot Client Views
☑	○ CityTimes01-Cities	simulating Samsung Gala	2016-10-14 14:11:21	57.965	☑	Original ▾	☑	☑	☑	☑	☑
☑	○ CityTimes02-UTC	simulating Samsung Gala	2016-10-14 14:16:49	81.562	☑	Original ▾	☑	☑	☑	☑	☑
☑	○ CityTimes03-Refresh	simulating Samsung Gala	2016-10-14 14:20:02	944.117	☑	Original ▾	☑	☑	☑	☑	☑

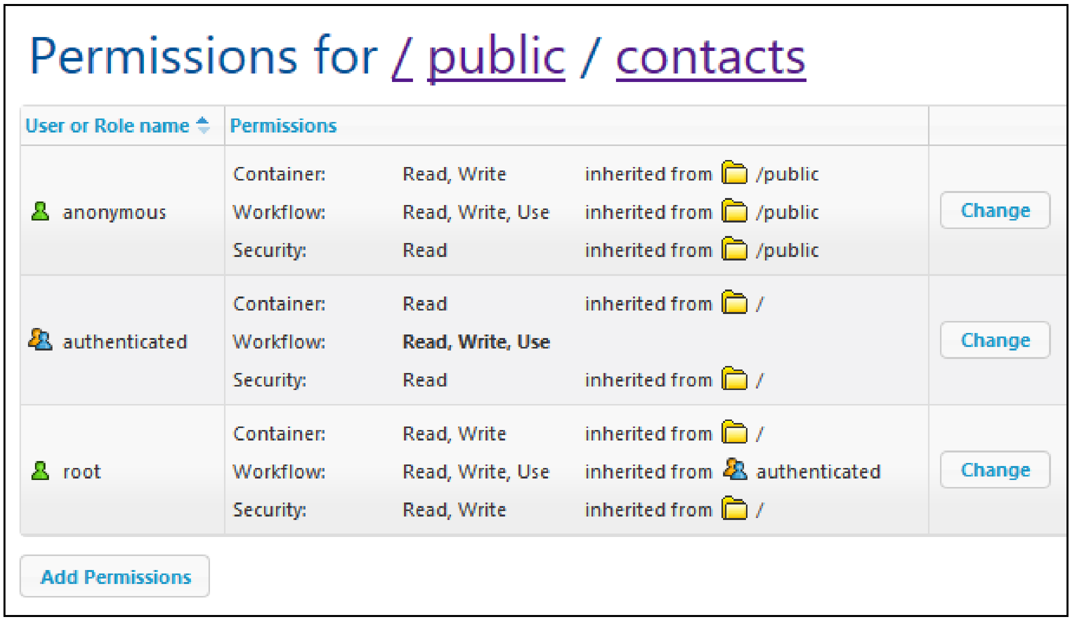
The Automated Tests page shows all the test runs that have been deployed for the selected solution. You can set up individual test runs for playback on client devices as follows:

1. In the *Active* column, check the test runs that you want to make active. These test runs will be played back on the client when the user starts a solution. If multiple test runs are selected, then all the selected test runs will be played back. If any one of a solution's test runs has been activated, then, on the Workflows page, the wheel in the design's *Automated Test* column is displayed in red.
2. Set the logging details you want during playback. Do this by checking the columns you want. See the Automated Testing section in the MobileTogether Designer documentation for information about these options.
3. Click **Save** to finish.

If you wish to delete a test run, select its check box in the leftmost column and click **Delete Selected**.

Permissions

Permissions are access rights, and they can be set for each container individually. Permissions determine which users or roles have access to that container, and what kind of access each user/role has (read, write, use). These access rights can be set for the container, its workflows (or solutions), and read/write security.



Permissions for / public / contacts

User or Role name	Permissions			
anonymous	Container:	Read, Write	inherited from /public	Change
	Workflow:	Read, Write, Use	inherited from /public	
	Security:	Read	inherited from /public	
authenticated	Container:	Read	inherited from /	Change
	Workflow:	**Read, Write, Use**		
	Security:	Read	inherited from /	
root	Container:	Read, Write	inherited from /	Change
	Workflow:	Read, Write, Use	inherited from authenticated	
	Security:	Read, Write	inherited from /	

⊟ *Rules for inheritance of permissions*

- For containers, permissions are inherited from parent containers.
- For users, permissions are inherited from all roles the user is a member of, as well as from permissions directly assigned to the user.
- Inheritance rules for users take precedence over container hierarchy rules.
- If a permission is redefined for any role the user is a member of, container hierarchy inheritance for this particular permission is overridden.

Permissions are checked for every user interaction. A user can only successfully access and/or edit when all required permissions are granted. Permissions are set for the following groups:

Container

- *Read:* The user can list the contents and find an object in the container.
- *Read-Write:* Additional to read, can create new (and delete existing) objects, depending on other permissions that may apply.

Workflow

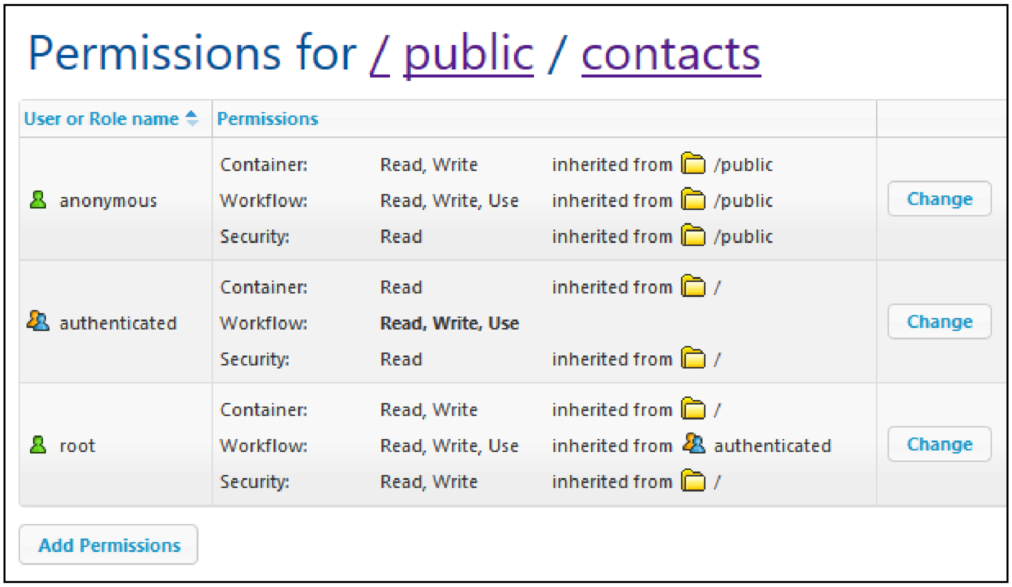
- *Read:* The user can run solutions.
- *Read-Write:* The user can additionally write to solution data. To modify file data, write access must also be granted for the container.

Security

- *Read:* The user is permitted to read the permission list of any child object of the container.
- *Read-Write:* The user can additionally change the permissions list of any child object of the container.
- By default a user is permitted to read only permissions assigned to it or a role it is a member of. If the *Read Users and Roles* privilege is granted (*see Users and Roles*), users can read all permission entries.

⊟ *Editing the permissions of a container*

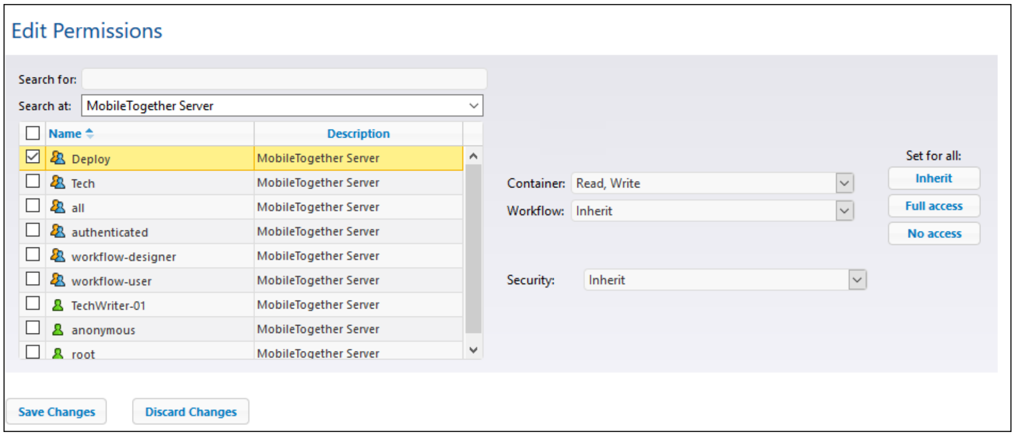
1. Click the **Permissions** button of the container. This takes you to the container's *Permissions* page (*screenshot below*).



Permissions for / public / contacts

User or Role name	Permissions			
anonymous	Container:	Read, Write	inherited from /public	Change
	Workflow:	Read, Write, Use	inherited from /public	
	Security:	Read	inherited from /public	
authenticated	Container:	Read	inherited from /	Change
	Workflow:	**Read, Write, Use**		
	Security:	Read	inherited from /	
root	Container:	Read, Write	inherited from /	Change
	Workflow:	Read, Write, Use	inherited from authenticated	
	Security:	Read, Write	inherited from /	

2. To edit the access rights of an already permitted user/role, click its **Change** button (*see screenshot above*). To add permissions for a new user/role, click **Add Permissions**. Both these buttons open the *Edit Permissions* pane.



Edit Permissions

Search for:

Search at: MobileTogether Server

☐	Name	Description
☑	Deploy	MobileTogether Server
☐	Tech	MobileTogether Server
☐	all	MobileTogether Server
☐	authenticated	MobileTogether Server
☐	workflow-designer	MobileTogether Server
☐	workflow-user	MobileTogether Server
☐	TechWriter-01	MobileTogether Server
☐	anonymous	MobileTogether Server
☐	root	MobileTogether Server

Container: Read, Write

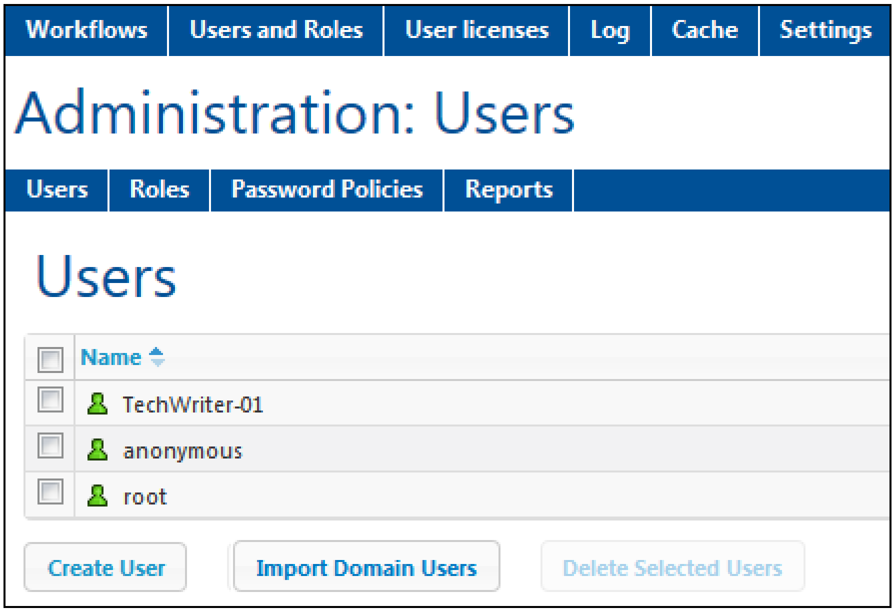
Workflow: Inherit

Security: Inherit

3. In the Edit Permissions pane, select a user/role by checking its check box. If you are editing existing permissions, permissions will be inherited from this user/role. If you are adding permissions, this user/role will be added to the permitted users/roles list of this container. In the *Search At* combo box, you can select users and roles that have been defined for MobileTogether Server or for all enabled domains (by selecting, respectively, *MobileTogether Server* or *Windows* in the combo box). A domain's users and roles are defined by the domain's administrator. They will be available in the pane only if the Active Directory Login setting has been enabled in the Settings tab.
4. Change the permissions as required. The *Inherit* option causes permissions to be inherited from the parent container.

5.2 Users and Roles

The **Users and Roles** tab (*screenshot below*) has four sub-tabs. These sub-tabs work together to enable user accounts to be administered. User accounts can be set up and configured for privileges, and summaries of accounts and privileges can be viewed in the *Reports* sub-tab. See the sub-sections for detailed descriptions.



About Users

A user is defined by a name-and-password combination. Users access MobileTogether Server in two ways:

- *Web UI access:* The Web UI is the administrative interface of MobileTogether Server. Logging in to the Web UI requires a name-and-password combination; it is therefore done as a user.
- *Service interface:* The HTTP service interface exposes MobileTogether Server services, typically to the MobileTogether Client app on a mobile device. A user accesses the service interface, using a name-and-password combination. The services exposed relate typically to access to MobileTogether solutions and their related data.

Two special users are predefined:

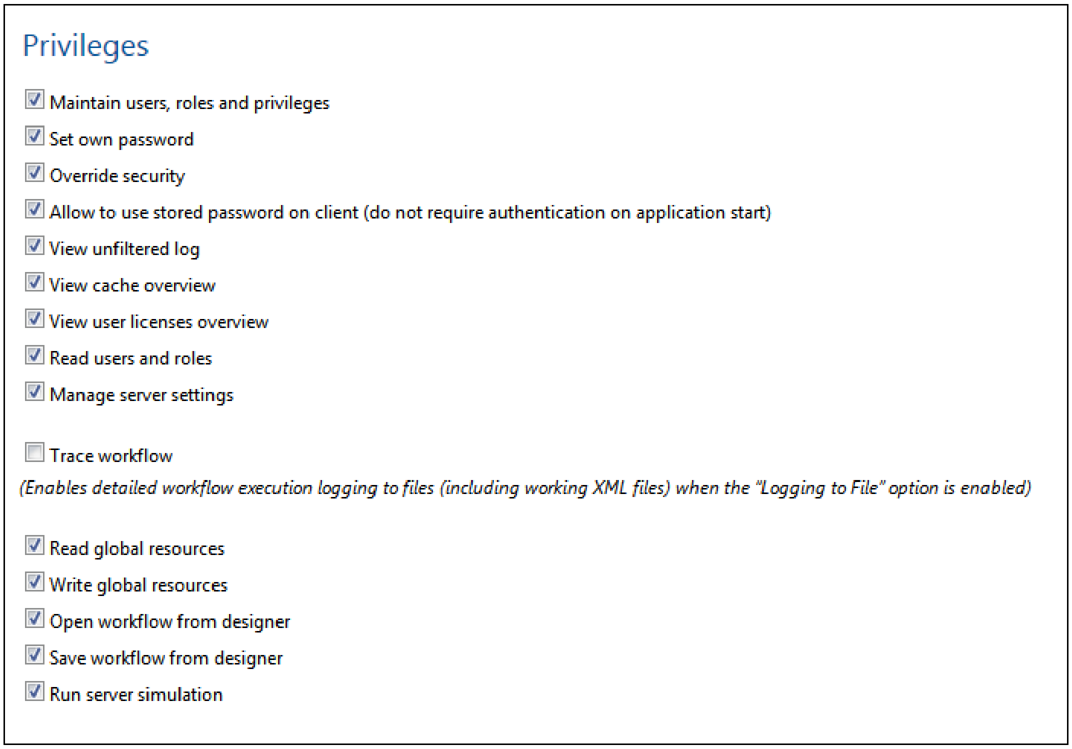
`root`	**`root`** is the initial administrator user. It is initially the most powerful user, having all privileges and having the ability to add other users and to set roles. Its initial name-password combination is: **`root-root`**. The password can be changed at any time.

anonymous	**anonymous** is an account for anonymous users that access services exposed via the HTTP service interface. It cannot be used for logging in top the Web UI, and it has no initial password.

About Privileges

A privilege is an activity that a user is allowed to carry out. There is a fixed number of MobileTogether Server privileges, and a user can be assigned zero to all of the available privileges. It is, however, good practice to assign privileges via roles (*see next section*), rather than to assign privileges directly to the user. The assigning of privileges and roles to a user is done by a user that has been assigned this privilege. Initially, it is **root** user that has this privilege.

The screenshot below shows all the available privileges.



Privileges

☑ Maintain users, roles and privileges
☑ Set own password
☑ Override security
☑ Allow to use stored password on client (do not require authentication on application start)
☑ View unfiltered log
☑ View cache overview
☑ View user licenses overview
☑ Read users and roles
☑ Manage server settings

☐ Trace workflow
(Enables detailed workflow execution logging to files (including working XML files) when the "Logging to File" option is enabled)

☑ Read global resources
☑ Write global resources
☑ Open workflow from designer
☑ Save workflow from designer
☑ Run server simulation

The tab *Users and Roles | Reports | Privileges Report* provides a list of all privileges, with each privilege being listed together with all the users/roles that have that privilege.

About Roles

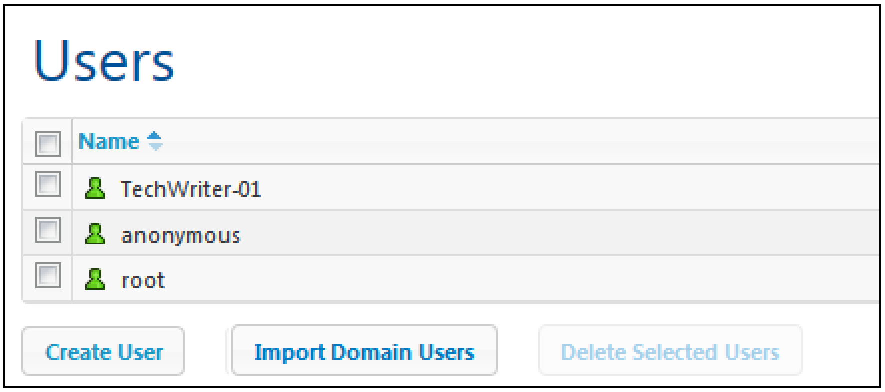
A role defines a set of privileges. It can be assigned to another role or to a user. A role's privileges automatically become the privileges of any other role or any user that the role is assigned to. A user can be assigned any number of roles. As a result, a user will have all the privileges defined in the multiple assigned roles.

The following roles are predefined:

- **`authenticated`** is automatically assigned to every user **`except` `anonymous`**. So user with a name-and-password is assigned the `authenticated` role.
- **`all`** is automatically assigned to every user **including `anonymous`**.
- **`workflow-designer`** is assigned to users that design workflows in MobileTogether Designer. This role allows a user to open and save workflows, as well as to run a simulation on the server.
- **`workflow-user`** is assigned to users running the workflow on a mobile device. This role allows the user to access the service interface without needing to log in to the server and start the solution on the client.

Users

The *Users and Roles* | *Users* tab (*screenshot below*) displays all users, and enables you to create new users, access a user's properties (by clicking a user name), and delete users.



Users

□	Name
□	TechWriter-01
□	anonymous
□	root

About Users

A user is defined by a name-and-password combination. Users access MobileTogether Server in two ways:

- *Web UI access:* The Web UI is the administrative interface of MobileTogether Server. Logging in to the Web UI requires a name-and-password combination; it is therefore done as a user.
- *Service interface:* The HTTP service interface exposes MobileTogether Server services, typically to the MobileTogether Client app on a mobile device. A user accesses the service interface, using a name-and-password combination. The services exposed relate typically to access to MobileTogether solutions and their related data.

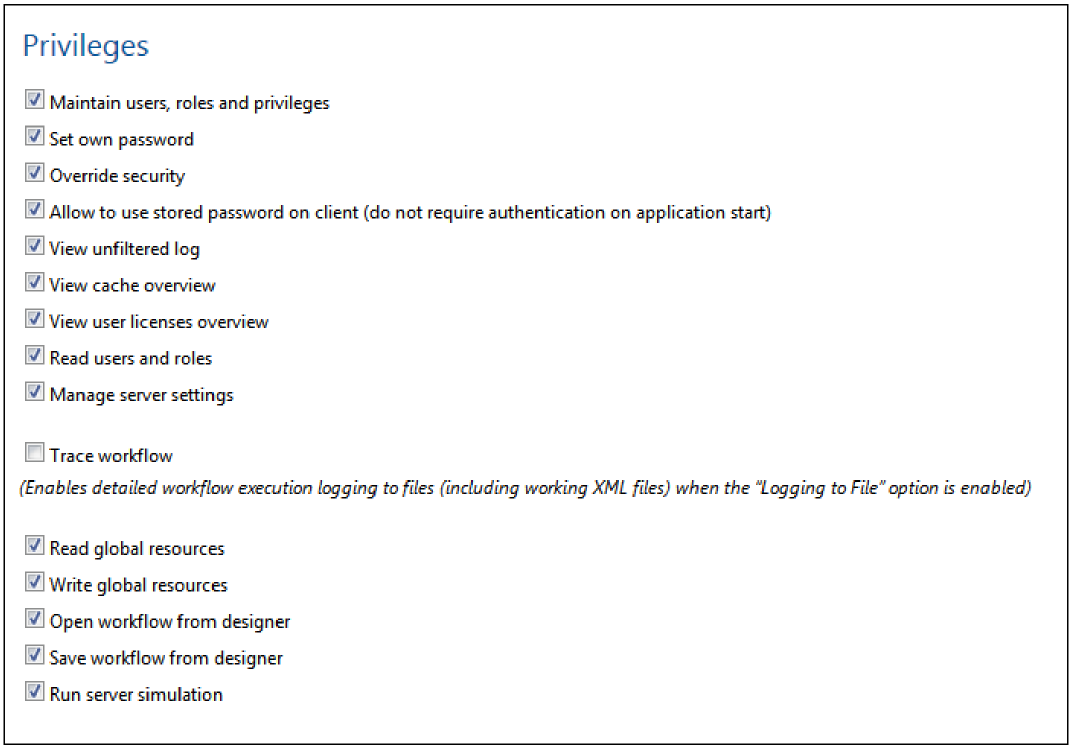
Two special users are predefined:

`root`	**`root`** is the initial administrator user. It is initially the most powerful user, having all privileges and having the ability to add other users and to set roles. Its initial name-password combination is: **`root-root`**. The password can be changed at any time.
`anonymous`	**`anonymous`** is an account for anonymous users that access services exposed via the HTTP service interface. It cannot be used for logging in top the Web UI, and it has no initial password.

About Privileges

A privilege is an activity that a user is allowed to carry out. There is a fixed number of MobileTogether Server privileges, and a user can be assigned zero to all of the available privileges. It is, however, good practice to assign privileges via roles (*see next section*), rather than to assign privileges directly to the user. The assigning of privileges and roles to a user is done by a user that has been assigned this privilege. Initially, it is **`root`** user that has this privilege.

The screenshot below shows all the available privileges.



Privileges

☑ Maintain users, roles and privileges

☑ Set own password

☑ Override security

☑ Allow to use stored password on client (do not require authentication on application start)

☑ View unfiltered log

☑ View cache overview

☑ View user licenses overview

☑ Read users and roles

☑ Manage server settings

□ Trace workflow

(Enables detailed workflow execution logging to files (including working XML files) when the "Logging to File" option is enabled)

☑ Read global resources

☑ Write global resources

☑ Open workflow from designer

☑ Save workflow from designer

☑ Run server simulation

The tab *Users and Roles | Reports | Privileges Report* provides a list of all privileges, with each privilege being listed together with all the users/roles that have that privilege.

About Roles

A role defines a set of privileges. It can be assigned to another role or to a user. A role's privileges automatically become the privileges of any other role or any user that the role is assigned to. A user can be assigned any number of roles. As a result, a user will have all the privileges defined in the multiple assigned roles.

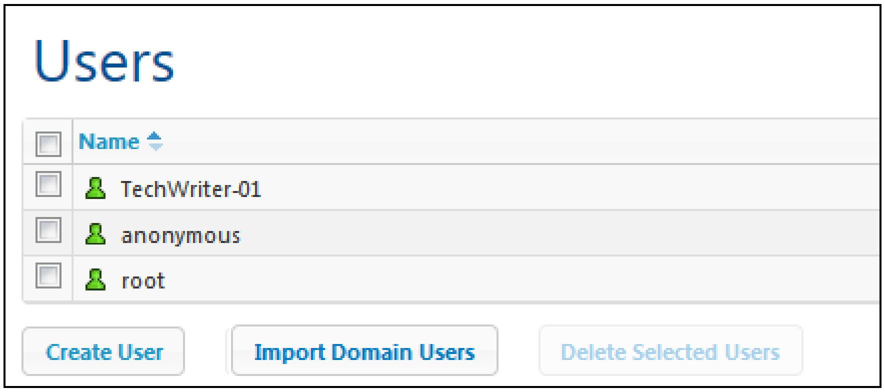
The following roles are predefined:

- **`authenticated`** is automatically assigned to every user **except `anonymous`**. So user with a name-and-password is assigned the `authenticated` role.
- **`all`** is automatically assigned to every user **including `anonymous`**.
- **`workflow-designer`** is assigned to users that design workflows in MobileTogether Designer. This role allows a user to open and save workflows, as well as to run a simulation on the server.
- **`workflow-user`** is assigned to users running the workflow on a mobile device. This role allows the user to access the service interface without needing to log in to the server and start the solution on the client.

Creating a user

A new user can be created by `root` user or any user that has the *Maintain users, roles, and privileges* privilege. Create a new user as follows:

1. In the *Users and Roles* | *Users* tab, click **Create User** (*see screenshot below*). This displays the Create User page.



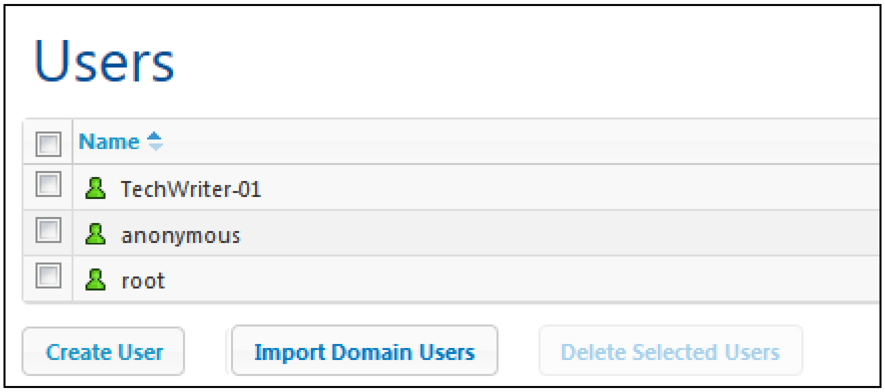
Users

□	Name
□	TechWriter-01
□	anonymous
□	root

2. On the Create User page, enter a user name and password.
3. To assign privileges to the user, you can either select the privileges directly (by checking their check boxes), and/or assign roles to the user (*see next section*). A user will have privileges that are directly assigned plus those inherited from all assigned roles. We recommend using roles to assign privileges to a user (*see next section*).
4. Select a password policy from the policies that you have defined.
5. Click **Save** to finish. The user now appears in the list of users (*see screenshot above*). You can edit a user's properties by clicking the user name in the list of users.

Importing a domain user

If Active Directory login has been enabled for a domain without automatically importing all users, you can import individual domain users of an enabled domain. Click **Import Domain Users** (*see screenshot below*). In the Import Domain Users dialog that is displayed, search for the user you want to import, select the user, and click **Import Selected**.

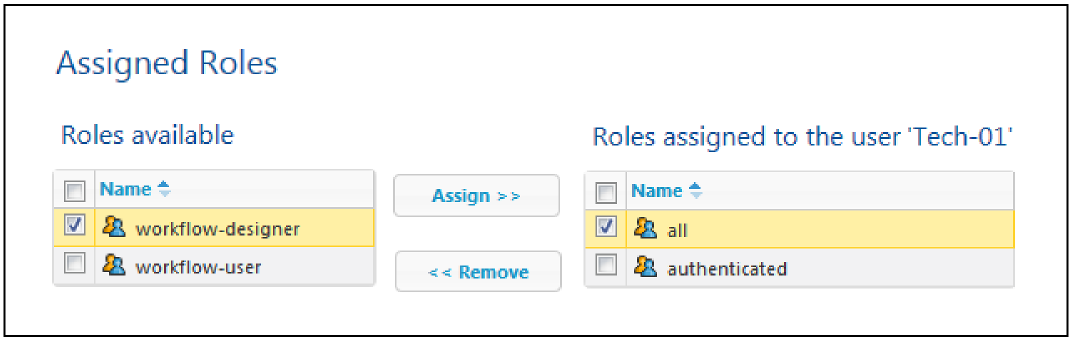


□	Name
□	TechWriter-01
□	anonymous
□	root

After the user is imported, you can assign roles to the user as for any other user. The new user can now log in to MobileTogether Server with the user's domain-specific user name and password.

▼ Assigning roles to a user

Roles can be assigned to a user on the user's Properties page. To access the user's Properties page, click the user name in the *Users and Roles* | *Users* tab. At the bottom of the user's Properties page is the Assigned Roles pane (*screenshot below*).



□	Name
☑	workflow-designer
□	workflow-user

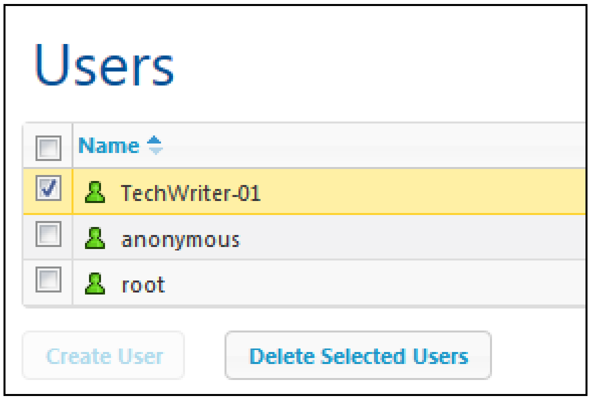
□	Name
☑	all
□	authenticated

All available roles are listed on the left. All roles assigned to the user are listed on the right. Select the available role (in the list on the left) that you want to assign, and click **Assign**. To remove an assigned role, select it in the list on the right, and click **Remove**.

To see a listing of all the privileges of a user, go to *Users and Roles | Reports | Privileges by User*.

▼ Deleting a user

A user can be deleted by `root` user or any user that has the *Maintain users, roles, and privileges* privilege. Delete a user as follows: In the *Users and Roles* | *Users* tab, select the user/s you want to delete (*see screenshot below*), click **Delete Selected Users**.



Users

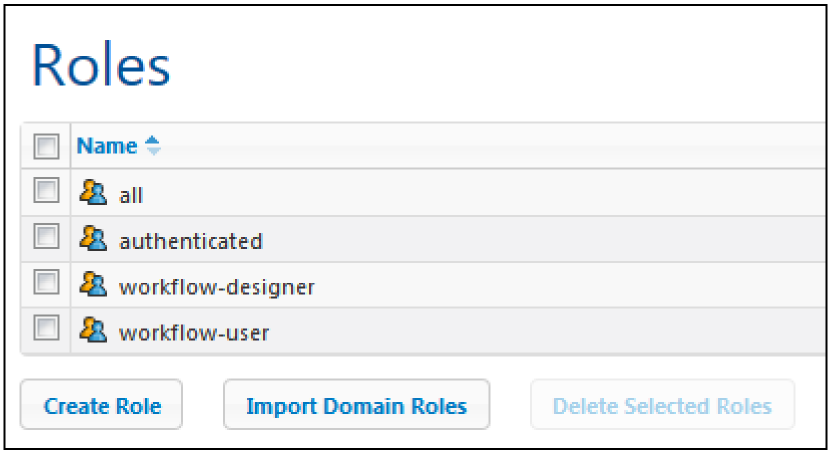
☐	Name
☑	TechWriter-01
☐	anonymous
☐	root

Roles

A role defines a set of privileges. It can be assigned to another role or to a user. A role's privileges automatically become the privileges of any other role or any user that the role is assigned to. A user can be assigned any number of roles. As a result, a user will have all the privileges defined in the multiple assigned roles.

The following roles are predefined:

- **`authenticated`** is automatically assigned to every user **except `anonymous`**. So user with a name-and-password is assigned the `authenticated` role.
- **`all`** is automatically assigned to every user **including `anonymous`**.
- **`workflow-designer`** is assigned to users that design workflows in MobileTogether Designer. This role allows a user to open and save workflows, as well as to run a simulation on the server.
- **`workflow-user`** is assigned to users running the workflow on a mobile device. This role allows the user to access the service interface without needing to log in to the server and start the solution on the client.



Roles

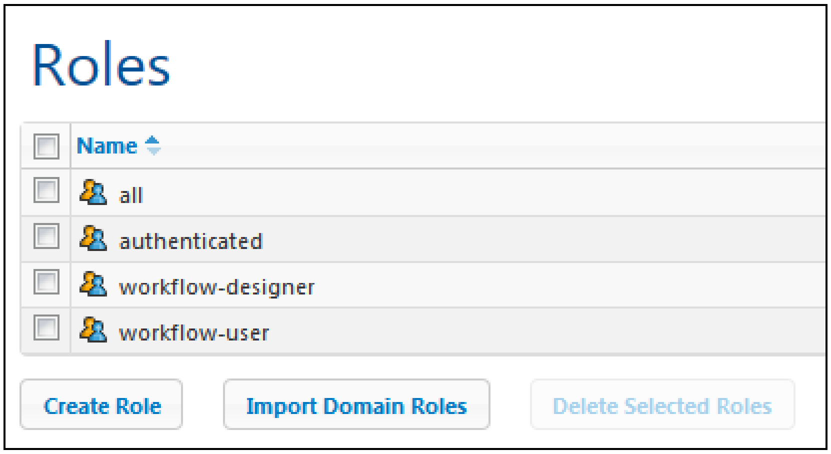
□	Name
□	all
□	authenticated
□	workflow-designer
□	workflow-user

Via the *Users and Roles* | *Roles* tab, you can create new roles, edit the properties of roles, and assign roles to users and/or other roles. Click the name of a role to access its Properties page, where you can select/deselect privileges and assign the role to a user and/or other roles.

▼ Creating a role and defining its privileges

A new role can be created by `root` user or any user that has the *Maintain users, roles, and privileges* privilege. Create a new role as follows:

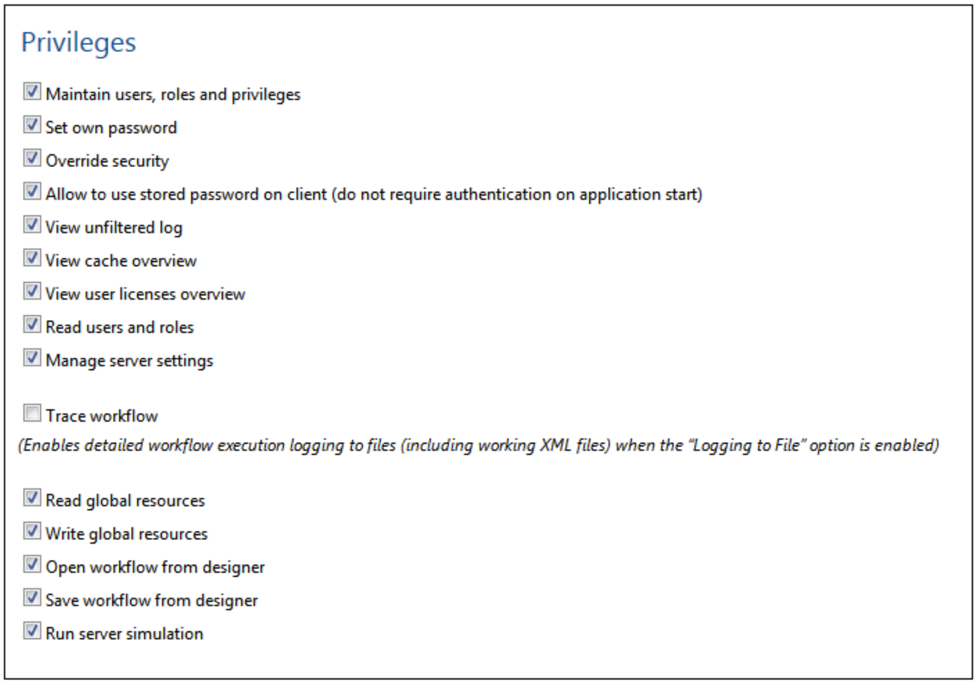
1. In the *Users and Roles* | *Roles* tab, click **Create Role** (*see screenshot below*). This displays the Create Role page.



Roles

□	Name
□	all
□	authenticated
□	workflow-designer
□	workflow-user

2. On the Create Role page, give the role a name.
3. To define privileges for the role, select the privileges by checking their check boxes.



Privileges

☑ Maintain users, roles and privileges
☑ Set own password
☑ Override security
☑ Allow to use stored password on client (do not require authentication on application start)
☑ View unfiltered log
☑ View cache overview
☑ View user licenses overview
☑ Read users and roles
☑ Manage server settings

□ Trace workflow

(Enables detailed workflow execution logging to files (including working XML files) when the "Logging to File" option is enabled)

☑ Read global resources
☑ Write global resources
☑ Open workflow from designer
☑ Save workflow from designer
☑ Run server simulation

4. Click **Save** to finish.

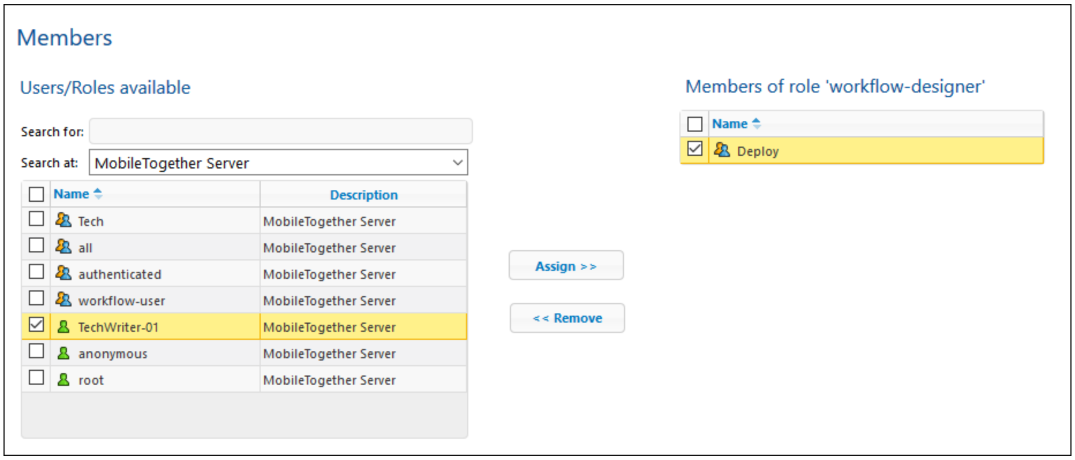
After you have saved the role, you can assign members to it in the Members pane at the bottom of the page (*see next section*). A member can be a user or another role. You can subsequently edit a role's properties by clicking the role's name in the list of roles in the *Users and Roles* | *Roles* tab.

To see a listing of all the privileges of a role, go to the tab, *Users and Roles | Reports | Privileges by User*.

▼ Assigning members (users or other roles) to a role

Roles can have members, which can be either users or other roles. Members inherit the privileges of its parent role.

To assign a member to a role, go to the Members pane at the bottom of the role's Properties page (*screenshot below*).



Members

Users/Roles available

Search for:

Search at: MobileTogether Server

□	Name	Description
□	Tech	MobileTogether Server
□	all	MobileTogether Server
□	authenticated	MobileTogether Server
□	workflow-user	MobileTogether Server
☑	TechWriter-01	MobileTogether Server
□	anonymous	MobileTogether Server
□	root	MobileTogether Server

Members of role 'workflow-designer'

□	Name
☑	Deploy

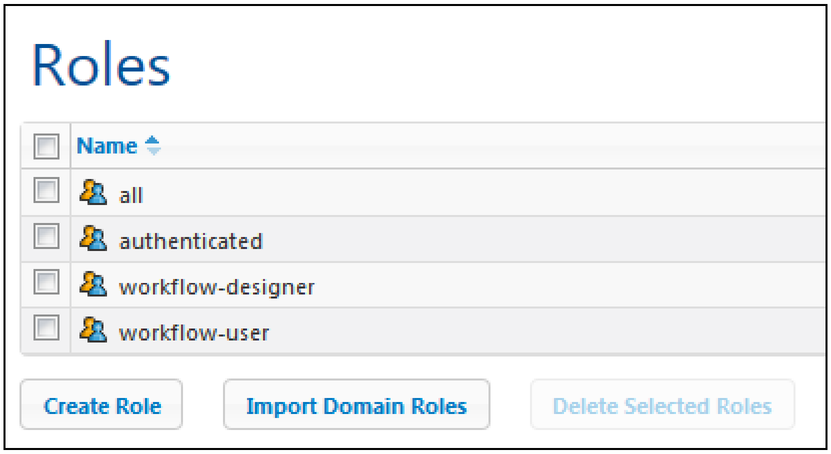
- All available users/roles are listed on the left.
- In the *Search At* combo box, you can select users and roles that have been defined for MobileTogether Server or for all enabled domains (by selecting, respectively, *MobileTogether Server* or *Windows* in the combo box). A domain's users and roles are defined by the domain's administrator. They will be available in the pane only if the Active Directory Login setting has been enabled in the Settings tab.
- You can search for a user/role by running a text search for its name in the *Search For* field.
- All users/roles that are members of the currently selected role are listed on the right.
- Select the user/role (from the list on the left) that you want to assign as a member, and click **Assign**.
- To remove an assigned user/role, select it in the list on the right, and click **Remove**.

The screenshot above, for example, shows the Members pane of the `workflow-designer` role. It has a single member, the role, `Deploy`., which will inherit all the privileges of the `workflow-designer` role.

Note that you can give a user or a role multiple sets of privileges. If a user/role is added as a member of multiple roles, it will inherit the privileges of all its parent roles. To see a listing of all the privileges of a user or role, go to the tab, *Users and Roles | Reports | Privileges by User*.

Importing a domain role

If Active Directory login has been enabled for a domain, you can import the individual roles of an enabled domain. Click **Import Domain Roles** (*see screenshot below*). In the Import Domain Roles dialog that is displayed, search for the role you want to import, select it, and click **Import Selected**.



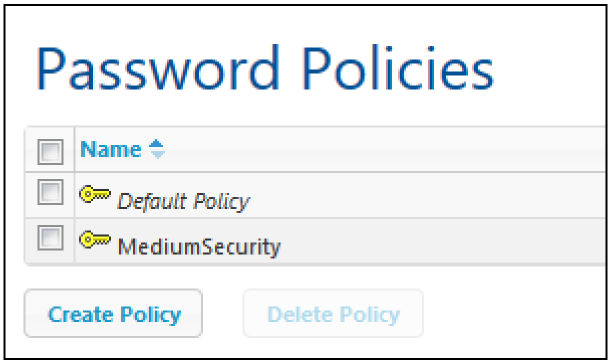
Roles

□	Name
□	all
□	authenticated
□	workflow-designer
□	workflow-user

After the role is imported, you can assign privileges to the role as for any other role. The new privileges will be allowed to those domain-specific roles..

Password Policies

A password policy defines the strength of passwords that use that policy. You can define your own password policies and apply different policies to different users. The *Users and Roles* | *Password Policies* tab (*screenshot below*) displays all defined password policies, enables you to create new policies, assign policies to users, and delete policies.

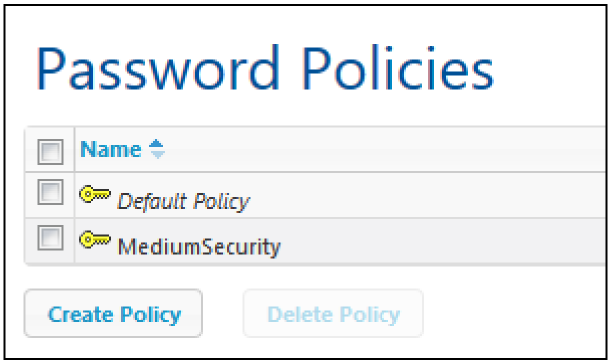


Note: By default every new user is assigned the **default password policy**, which does not define any constraint and cannot be changed. If you want users to have stronger passwords than defined by the default policy, create a strong policy and assign this policy to individual users.

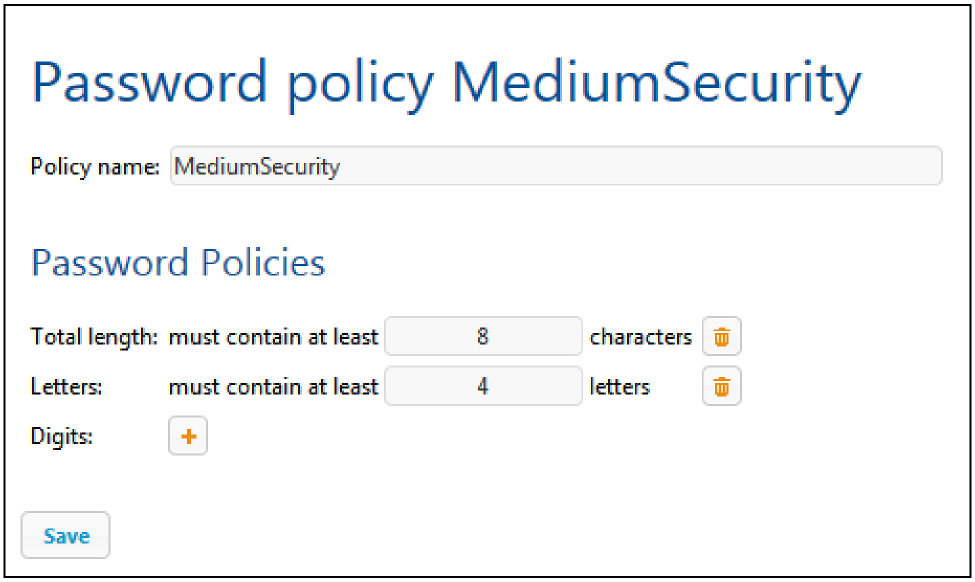
▼ Creating a password policy

A new password policy can be created by `root` user or any user that has the *Maintain users, roles, and privileges* privilege. Create a new password policy as follows:

1. In the *Users and Roles* | *Password Policies* tab, click **Create Policy** (*see screenshot below*). This displays the Create Password Policy page.



2. On the Create Password Policy page, give the policy a name.
3. To define the constraints of the password, click the plus icon next to a constraint (*Total length; Letters; Digits*), and enter a value for the constraint (*see screenshot below*).



Password policy MediumSecurity

Policy name: MediumSecurity

Password Policies

Total length: must contain at least 8 characters

Letters: must contain at least 4 letters

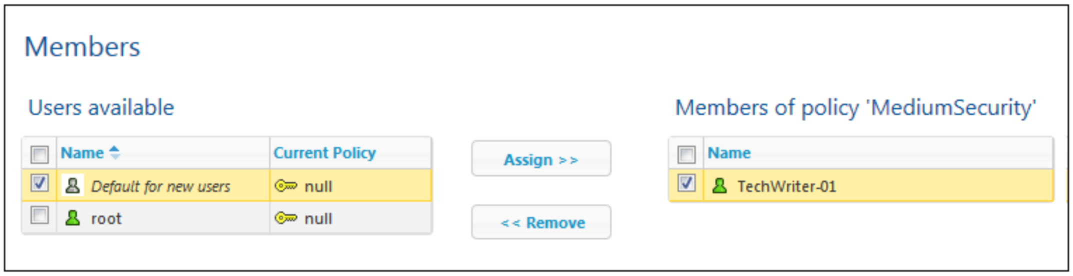
Digits:

4. Click **Save** to finish.

After you have saved the policy, you can assign users to it in the Members pane at the bottom of the page (*see next section*). You can subsequently edit a policy's properties by clicking its name in the list of policies in the *Users and Roles* | *Policies* tab.

▼ Assigning members (users) to a password policy

A password policy can be applied to a user by adding the user as a member of the policy in the Members pane at the bottom of the policy's Properties page (*see screenshot below*).



Members

Users available

☐	Name	Current Policy
☑	Default for new users	null
☐	root	null

Members of policy 'MediumSecurity'

☐	Name
☑	TechWriter-01

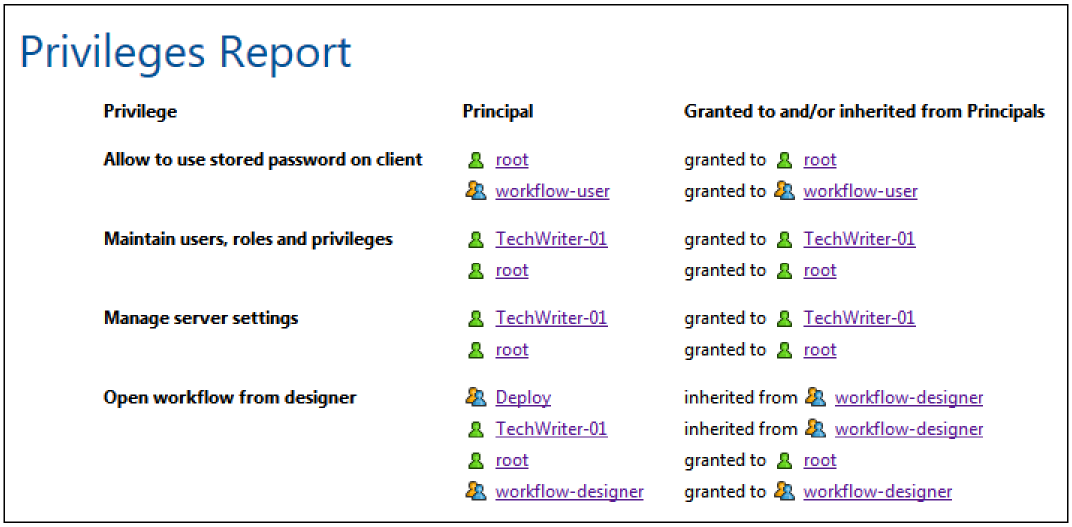
All available users are listed on the left. All users that are members of the policy are listed on the right. Select the user that you want to assign as a member from the list on the left, and click **Assign**. To remove an assigned user, select it in the list on the right, and click **Remove**. The screenshot above, for example, shows the Members pane of the `MediumSecurity` policy. It has a single member, the user `TechWriter-01`.

Reports

The *Users and Roles* | *Reports* tab provides links to reports about privileges. These reports are useful summaries of what users/roles use what privileges.

▼ Privileges Report

The Privileges Report (*screenshot below*) lists each privilege together with all the users and roles that use that privilege. The inheritance is also displayed.

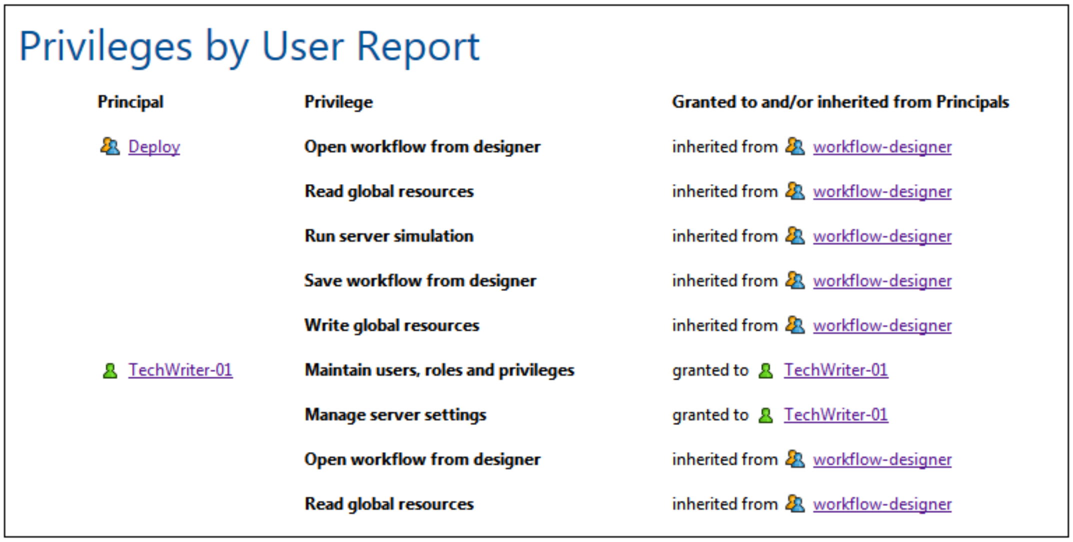


Privileges Report

Privilege	Principal	Granted to and/or inherited from Principals
Allow to use stored password on client	root	granted to root
	workflow-user	granted to workflow-user
Maintain users, roles and privileges	TechWriter-01	granted to TechWriter-01
	root	granted to root
Manage server settings	TechWriter-01	granted to TechWriter-01
	root	granted to root
Open workflow from designer	Deploy	inherited from workflow-designer
	TechWriter-01	inherited from workflow-designer
	root	granted to root
	workflow-designer	granted to workflow-designer

▼ Privileges-by-User Report

The Privileges-by-User Report (*screenshot below*) lists each user/role with a summary of its privileges. The inheritance is also displayed.

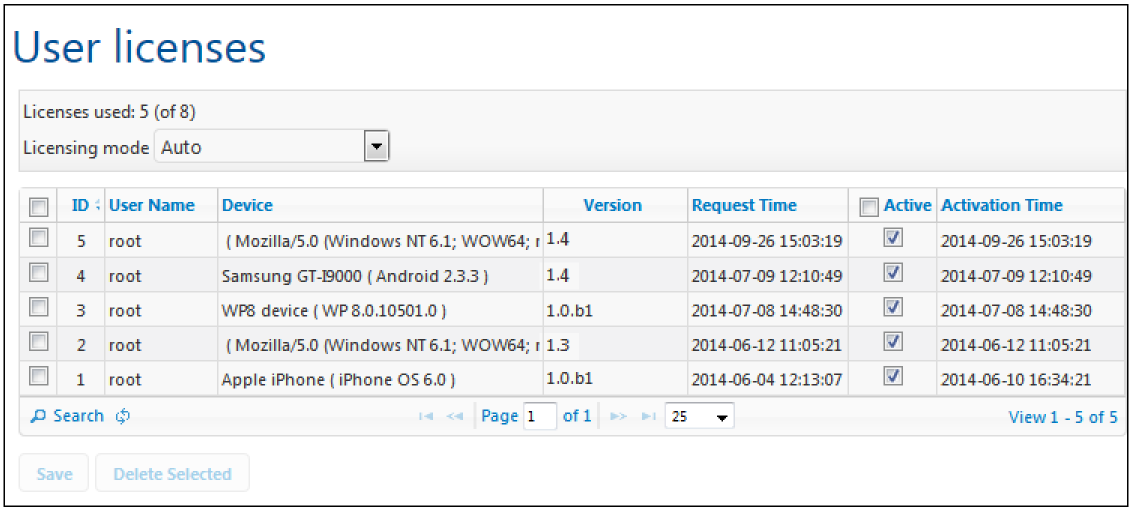


Privileges by User Report

Principal	Privilege	Granted to and/or inherited from Principals
Deploy	**Open workflow from designer**	inherited from workflow-designer
	Read global resources	inherited from workflow-designer
	Run server simulation	inherited from workflow-designer
	Save workflow from designer	inherited from workflow-designer
	Write global resources	inherited from workflow-designer
TechWriter-01	**Maintain users, roles and privileges**	granted to TechWriter-01
	Manage server settings	granted to TechWriter-01
	Open workflow from designer	inherited from workflow-designer
	Read global resources	inherited from workflow-designer

5.3 User Licenses

The **User Licenses** tab (*screenshot below*) displays license information about the devices currently connected to and licensed with MobileTogether Server, and enables licenses to be activated and deactivated.



User licenses

Licenses used: 5 (of 8)

Licensing mode: Auto

□	ID	User Name	Device	Version	Request Time	□ Active	Activation Time
□	5	root	(Mozilla/5.0 (Windows NT 6.1; WOW64; r	1.4	2014-09-26 15:03:19	☑	2014-09-26 15:03:19
□	4	root	Samsung GT-I9000 (Android 2.3.3)	1.4	2014-07-09 12:10:49	☑	2014-07-09 12:10:49
□	3	root	WP8 device (WP 8.0.10501.0)	1.0.b1	2014-07-08 14:48:30	☑	2014-07-08 14:48:30
□	2	root	(Mozilla/5.0 (Windows NT 6.1; WOW64; r	1.3	2014-06-12 11:05:21	☑	2014-06-12 11:05:21
□	1	root	Apple iPhone (iPhone OS 6.0)	1.0.b1	2014-06-04 12:13:07	☑	2014-06-10 16:34:21

- A MobileTogether Server license allows a certain number of devices to communicate with the MobileTogether Server at any given time. This number is given in the *Licenses used* field. For example, in the screenshot above, the server is licensed to communicate with 8 devices. Five devices are connected, and all are licensed (indicated by their *Active* check boxes being selected). The *Licenses used* field therefore shows 5 out of 8 licenses used.

- Once a client device connects to the server, it will be assigned a license automatically if the *Licensing mode* option is set to `Auto` (*see screenshot above*). If this option is set to `Manual`, a newly connected mobile device is shown in the list of connected devices. It will be licensed only when an administrator checks the device's *Active* check box and clicks **Save**.

- Once the user license limit is reached, no more devices can be licensed. In order to license additional devices, an existing licensed device must first be delicensed, by deactivating its license. An administrator can activate and deactivate device licenses at any time so that new devices can be licensed without exceeding the user license limit.

User license fields

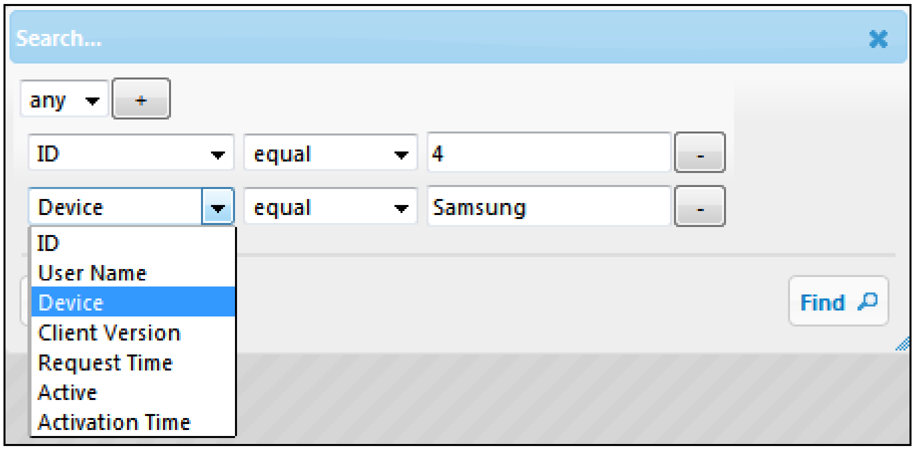
Given below is a description of the fields of the user license tab.

- *Licensing mode:* `Auto` automatically activates a license for a newly connected device, provided one is free. `Manual` requires that the administrator manually activate a license for a device, and then save the setting for the activation to take effect.
- *ID:* The internal number assigned to the licensed device.
- *User Name:* The user name with which the client device made the connection and logged in. The user name determines the privileges that are extended to the client device.

- *Device:* The mobile device or browser that requested the license.
- *Client Version:* The version of the MobileTogether Client app on the client device. Knowing the client version can be important for debugging and troubleshooting errors that might occur on the client device.
- *Active:* The *Active* check box is used to activate/deactivate a license. Click **Save** to finalize the change.
- *Request Time, Activation Time:* The times when a license is requested and activated, respectively.

Search

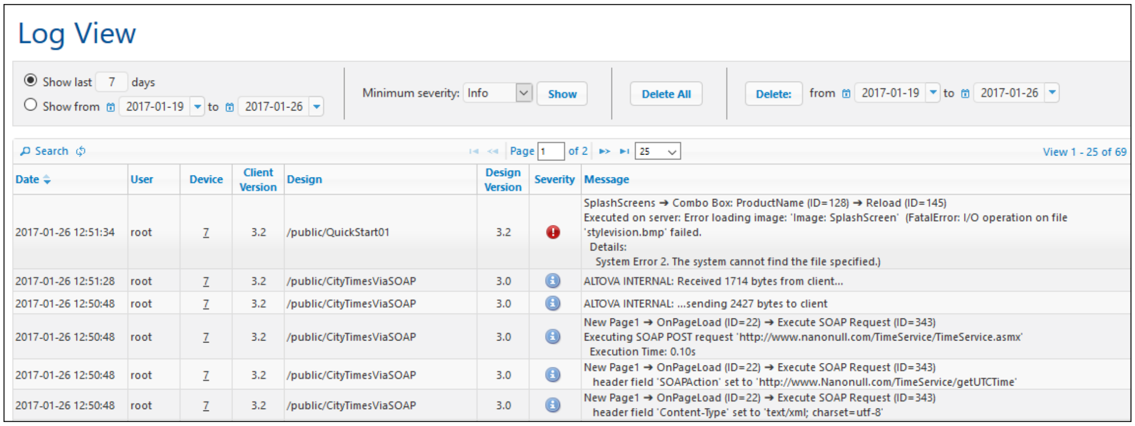
Click the **Search** button to open the Search dialog (*screenshot below*) and search by a combination of one or more user license fields.



- The *Any/All* combo box specifies whether all the rules you define must be satisfied, or any one rule.
- The **Add Rule** icon next to the *Any/All* combo box adds a rule to the search definition.
- Each rule consists of three parts: (i) a user license field, (ii) a relationship definition, and (iii) a value.
- A submitted value must exactly match a value in the specified field to return a match.
- An empty value part will use an empty string as that field's submitted value.
- The **Delete Rule** icon next to each rule deletes that rule.
- Click **Find** to start the search.
- Click **Reset** to show all user licenses.

5.4 Log

The **Log** tab shows the logged actions according to the selected filters, located at the top of the tab (*see screenshot below*). If you wish to see all the logged actions (rather than only warnings and errors), go to In the *Settings* tab and make sure that the Logging level detail is set to `Info`. The log columns relating to each design action show the following: the name of the user, the client device (identified by an ID, the associated details of which can be seen in the *User Licenses* tab), the version number of the MobileTogether Client app on the device, the version of MobileTogether Designer with which the design was created, and the severity of the message (Info, Warning, Error).



Log View

● Show last 7 days
○ Show from 2017-01-19 to 2017-01-26
Minimum severity: Info
from 2017-01-19 to 2017-01-26

Page 1 of 2 | 25 | View 1 - 25 of 69

Date	User	Device	Client Version	Design	Design Version	Severity	Message
2017-01-26 12:51:34	root	7	3.2	/public/QuickStart01	3.2	!	SplashScreens → Combo Box: ProductName (ID=128) → Reload (ID=145) Executed on server: Error loading image: 'Image: SplashScreen' (FatalError: I/O operation on file 'stylevision.bmp' failed. Details: System Error 2. The system cannot find the file specified.)
2017-01-26 12:51:28	root	7	3.2	/public/CityTimesViaSOAP	3.0	i	ALTOVA INTERNAL: Received 1714 bytes from client...
2017-01-26 12:50:48	root	7	3.2	/public/CityTimesViaSOAP	3.0	i	ALTOVA INTERNAL: ...sending 2427 bytes to client
2017-01-26 12:50:48	root	7	3.2	/public/CityTimesViaSOAP	3.0	i	New Page1 → OnPageLoad (ID=22) → Execute SOAP Request (ID=343) Executing SOAP POST request 'http://www.nanonull.com/TimeService/TimeService.asmx' Execution Time: 0.10s
2017-01-26 12:50:48	root	7	3.2	/public/CityTimesViaSOAP	3.0	i	New Page1 → OnPageLoad (ID=22) → Execute SOAP Request (ID=343) header field 'SOAPAction' set to 'http://www.Nanonull.com/TimeService/getUTCTime'
2017-01-26 12:50:48	root	7	3.2	/public/CityTimesViaSOAP	3.0	i	New Page1 → OnPageLoad (ID=22) → Execute SOAP Request (ID=343) header field 'Content-Type' set to 'text/xml; charset=utf-8'

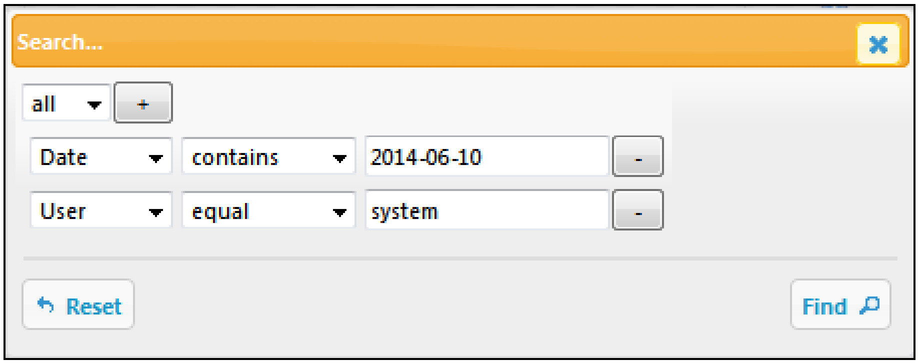
The view can be filtered by:

- *Date:* Ranges or specific dates can be set.
- *Minimum severity:* Errors is the highest severity (only errors are shown); Warnings is next (errors and warnings are shown); Info is the lowest severity, and shows errors, warnings, and info.
- *Search criteria:* Click the **Search** button at the top or bottom left of the log table to open the Search dialog (*described below*). To remove the filter defined by the search criteria, click the **Reload Grid** icon next to the **Search** button.

Logs can be deleted by clicking the **Delete All** button at the top of the tab, or by defining a date range and clicking **Delete**.

Searching for log messages

To access the Search dialog (*screenshot below*), click the **Search** button at the top or bottom left of the log table (*see screenshot above*).



For each search rule, select a field (such as *Date* or *User*), an operator (such as *contains* or *equals*), and the value to search for. Add a new search rule by clicking the **Add Rule** button. Delete a search rule by clicking its **Delete Rule** button. The **all** selector at the top specifies that the search condition is fulfilled only when all the search rules are individually fulfilled. The **any** selector specifies that the search condition is fulfilled if any one search rule is fulfilled. Click **Find** to start the search. Click **Reset** to remove the search filter.

5.5 Cache

The **Cache** tab displays the details of caches that are currently available on the server. A cache is a data file that is generated from a data source of a design at a given time. In the Web UI of MobileTogether Server, you can see details about the caches on the server, activate/deactivate caches, and delete caches.

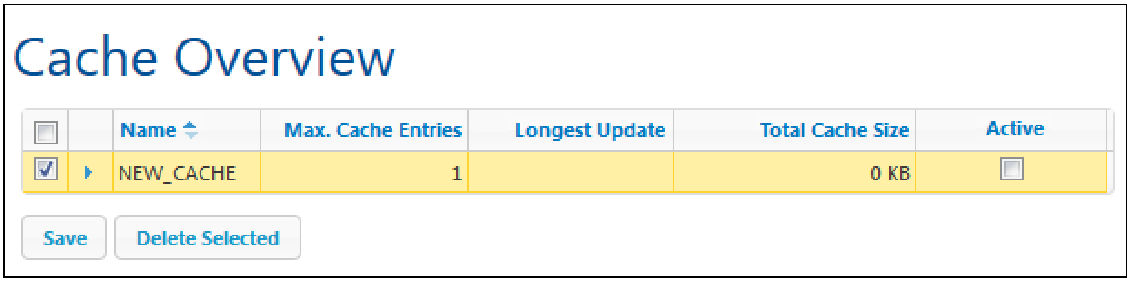
Creating caches

A new cache is defined in MobileTogether Designer for a data source. Right-click a data source in the Page Sources window, select **Cache Settings**, and specify the properties of the cache. There are two main reasons to create caches: (i) If a page data source generates reports slowly (for example, a large database); (ii) If a data source is not modified often. In such cases, execution of a solution would be faster if data is taken from data caches on the server. In order to keep caches up-to-date, the frequency of cache updates can be specified when the cache is created. Once a cache has been defined in MobileTogether Designer, it can be used by the data sources of different designs, providing the underlying data structure is compatible.

If a data source is defined as having a cache, the cached data will be used when the solution is run. Caches can be used as soon as the solution has been deployed to the server.

Details displayed and available actions

In the Cache tab you can see all caches available on the server, activate or deactivate individual caches, save any change to the active/inactive status of a cache, and delete selected caches.



☐		Name ⬍	Max. Cache Entries	Longest Update	Total Cache Size	Active
☑	▸	NEW_CACHE	1		0 KB	☐

- *Name:* Names are given at the time a cache is defined in MobileTogether Designer and cannot be changed in MobileTogether Server. The same cache can be used in multiple designs if the referenced data structure is compatible. Assigning caches to different designs is done in MobileTogether Designer. See the http://manual.altova.com/MobileTogether/mobiletogetherdesigner/index.html for more information.
- *Max. Cache Entries:* A cache can be created by using a set of parameters. Each set of parameters generates one cache entry, and if different sets of parameters are specified, multiple cache entries will be created. When defining a cache, the maximum number of cache entries can be specified so that an upper limit is placed on the amount of server space that is used for the caches of a particular data source. This number specifies how many cache entries will be stored before the first cache entry is purged and the latest cache entry is appended.

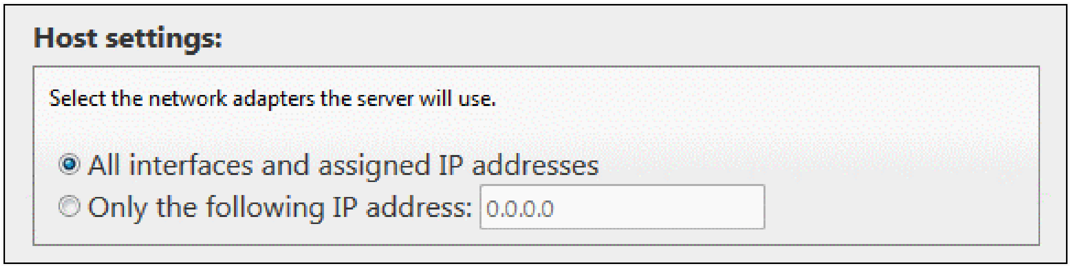
- *Longest Update:* Each cache can be updated multiple times. This column displays the time taken for the longest update.
- *Total Cache Size:* This is the total size (for all cache entries) allocated to the cache on the hard disk (or other medium). Cache size is automatically allocated.
- *Active:* This check box lets you activate or deactivate a cache on the server. When a cache is deactivated, its metadata (properties) still exists on the server but the cache is emptied and it is not available. Click **Save** to confirm the setting.
- To delete a cache from the server, select it and click **Delete Selected**. If the cache has been defined to be updated periodically, a new cache will be generated at the next update time.

5.6 Settings

The **Settings** tab is divided into a *General Settings* pane and a *LicenseServer* pane. The *General Settings* pane contains sub-panes for various MobileTogether Server settings. These sub-panes and the LicenseServer pane are described separately below. After changing any setting in the *General Settings* pane, click **Save** for the setting to take effect.

Host settings

Specifies whether the server will use a specific IP address or all interfaces and IP addresses. If a single IP address is to be used, enter it in the *Only the following IP address* field.



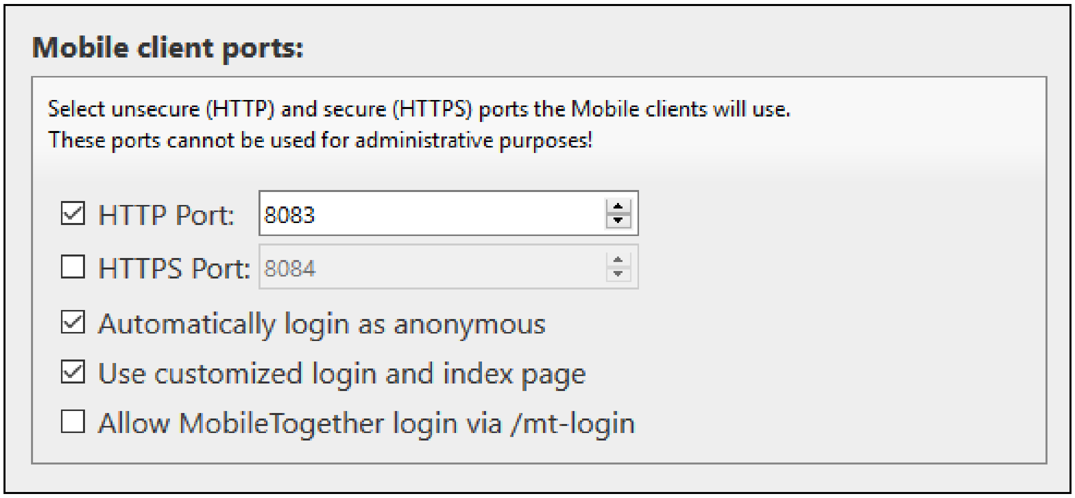
Host settings:

Select the network adapters the server will use.

● All interfaces and assigned IP addresses
○ Only the following IP address: 0.0.0.0

Mobile client ports

The ports that mobile devices will use to connect to the server. The `HTTP` port is the unsecure port; `HTTPS` is the secure port. To use `HTTPS`, you need to set up SSL Encryption.



Mobile client ports:

Select unsecure (HTTP) and secure (HTTPS) ports the Mobile clients will use.
These ports cannot be used for administrative purposes!

☑ HTTP Port: 8083
□ HTTPS Port: 8084
☑ Automatically login as anonymous
☑ Use customized login and index page
□ Allow MobileTogether login via /mt-login

Automatically login as anonymous
If selected, clients will be logged in automatically as `anonymous`. The login page is skipped, and the server's first page is shown directly. The first page is either the standard page that displays the root folder, or it is a custom page that you have defined (*see next point*). If this option is **not** selected, the client will need to login with the appropriate credentials via the default login page. If anonymous login is selected, then remember to set the relevant privileges for `anonymous`.

Use customized login and index pages

Select this option if a customized login page and first page should be used. This enables you to design your own entry point for clients. Set up the customized pages as follows:

1. Create the two pages as HTML pages, and name them **`login.html`** and **`index.html`**, respectively.
2. Save the two files in the **`index`** folder that is located in the MobileTogether Server ***application data folder*** (*see table below*). Additional files, such as image files and CSS files, are best saved in a subfolder of the **`index`** folder (for instance in one that is called, say, **`static`**).

Linux	`/var/opt/Altova/MobileTogetherServer`
Mac	`/var/Altova/MobileTogetherServer`
Windows Vista, Windows 7, 8, 10	`C:\ProgramData\Altova\MobileTogetherServer`

The code listings of a sample login page and sample first (index) page are given below. These listings are basic, but you can modify the code as you like.

login.html

```
<html>
  <header>
    <title>Customized Login</title>
  </header>
  <head>
    <meta http-equiv="Cache-Control" content="no-store" />
  </head>
  <body>
    <div>
      <h1>Sign in</h1>
      <p>A bare-basics custom page for client logins to
MobileTogether Server. Modify this page as required, and use the
Static sub-folder to save CSS stylesheets, images, etc.</p>
      <form method="post" action="/do_login" name="loginform">
        <table>
          <tbody>
            <!-- The user to login -->
            <tr><td>Username:</td></tr>
            <tr><td><input type="text" name="username" size="30"></
td></tr>
            <!-- The password of the user -->
            <tr><td>Password:</td></tr>
            <tr><td><input type="password" name="password"
size="30"></td></tr>
            <!-- The Active Directory domian details -->
            <tr><td>&nbsp;</td></tr>
            <tr><td>Active Directory Login:</td></tr>
            <tr>Domain suffix: <td><input type="providernamesuffix"
name="providernamesuffix" value=""></td></tr>
            <tr>Domain prefix: <td><input type="providernameprefix"
name="providernameprefix" value=""></td></tr>
```

```
            <!-- The Sign-In button -->
            <tr><td><input type="submit" value="Sign in"></td></tr>
          </tbody>
        </table>
        <!-- The page to redirect to after a successful login. -->
        <input type="hidden" name="from_page" value="/index"></
input><br>
      </form>
    </div>
  </body>
</html>
```

index.html

```
<html>
  <header>
    <title>Custom Index</title>
  </header>
  <head>
    <meta http-equiv="Cache-Control" content="no-store" />
    <title>Custom Index</title>
  </head>
  <body>
    <img src="/index/static/logo.png"></img><hr/>
    <a href="/do_logout">Logout</a>
    <p>MobileTogether Custom Login</p>
    <div><a href='/run?d=/public/About'>Start the About app</a></
div>
    <div><a href='/run?d=/public/DateCalc'>Start the Date
Calculator app</a></div>
    <div><a href='/run?d=/public/WorldPopulation'>Start the World
Population Statics app</a></div>
  </body>
</html>
```

Allow MobileTogether login via /mt-login

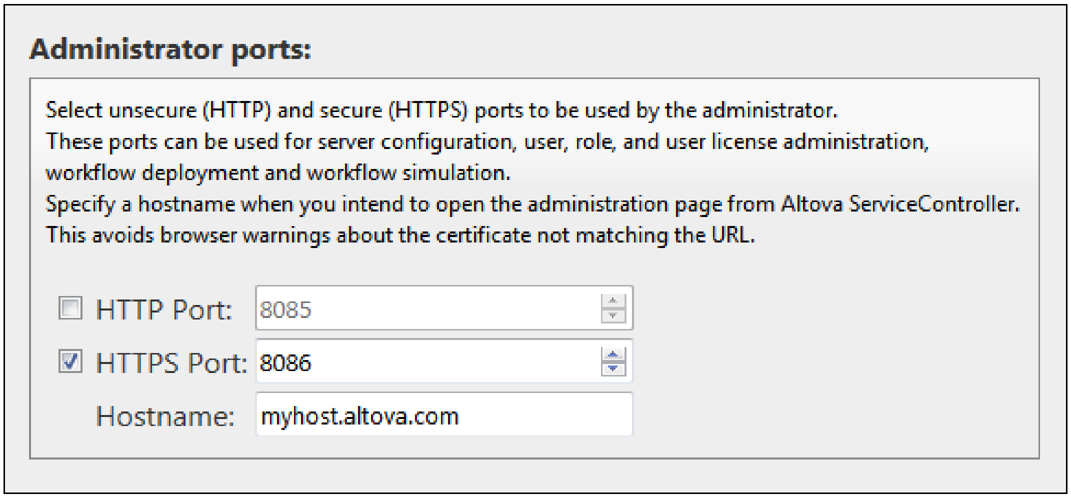
This option specifies that the login will be via the default login page and first page—and not via the customized login and index pages. This allows you to store the **`login.html`** and **`index.html`** files at the designated location, but still use the default pages. Note that the client's browser or browser settings might require that the browser cache is emptied in order for this setting to take effect.

Administrator ports

The administrator ports provide access for the following purposes:

- To connect to the server's Web UI and perform administrative functions, such as setting up Users and Roles.

- To deploy MobileTogether designs (as MobileTogether solutions) to the server. MobileTogether Designer has a setting that specifies the address and port of the MobileTogether Server to which to deploy designs.



Administrator ports:

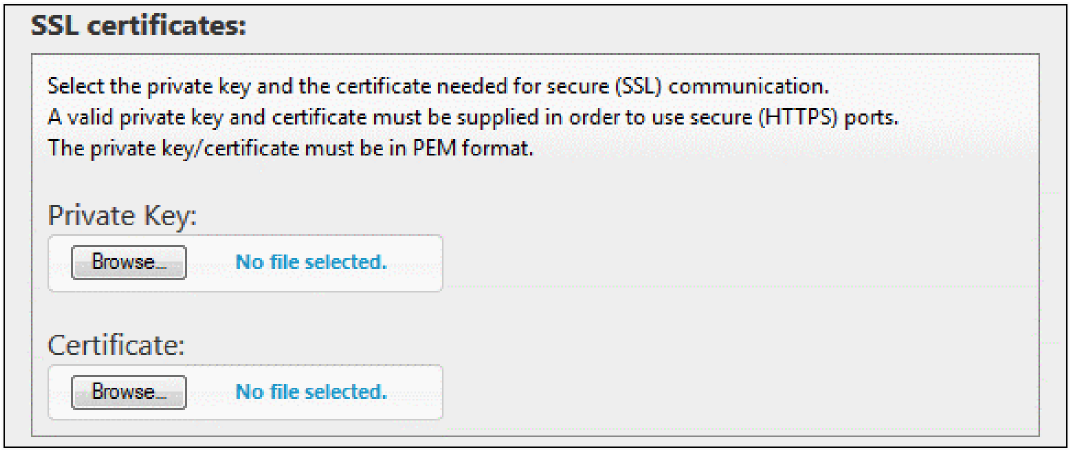
Select unsecure (HTTP) and secure (HTTPS) ports to be used by the administrator.
These ports can be used for server configuration, user, role, and user license administration, workflow deployment and workflow simulation.
Specify a hostname when you intend to open the administration page from Altova ServiceController. This avoids browser warnings about the certificate not matching the URL.

☐ HTTP Port: 8085
☑ HTTPS Port: 8086
Hostname: myhost.altova.com

The `HTTP` port is the unsecure port; `HTTPS` is the secure port. To use `HTTPS`, you need to set up SSL Encryption. If you set up the HTTPS port and wish to avoid browser warnings about the SSL certificate not matching the URL, then specify the hostname of the computer on which the MobileTogether Server configuration page will be opened.

SSL certificates

Specifies the private key certificate and public key certificate to use for SSL communication. Click **Browse**, and select the files containing the certificates (*Private Key* for private key, and *Certificate* for public key). See Setting Up SSL Encryption for more information.



SSL certificates:

Select the private key and the certificate needed for secure (SSL) communication.
A valid private key and certificate must be supplied in order to use secure (HTTPS) ports.
The private key/certificate must be in PEM format.

Private Key:

No file selected.

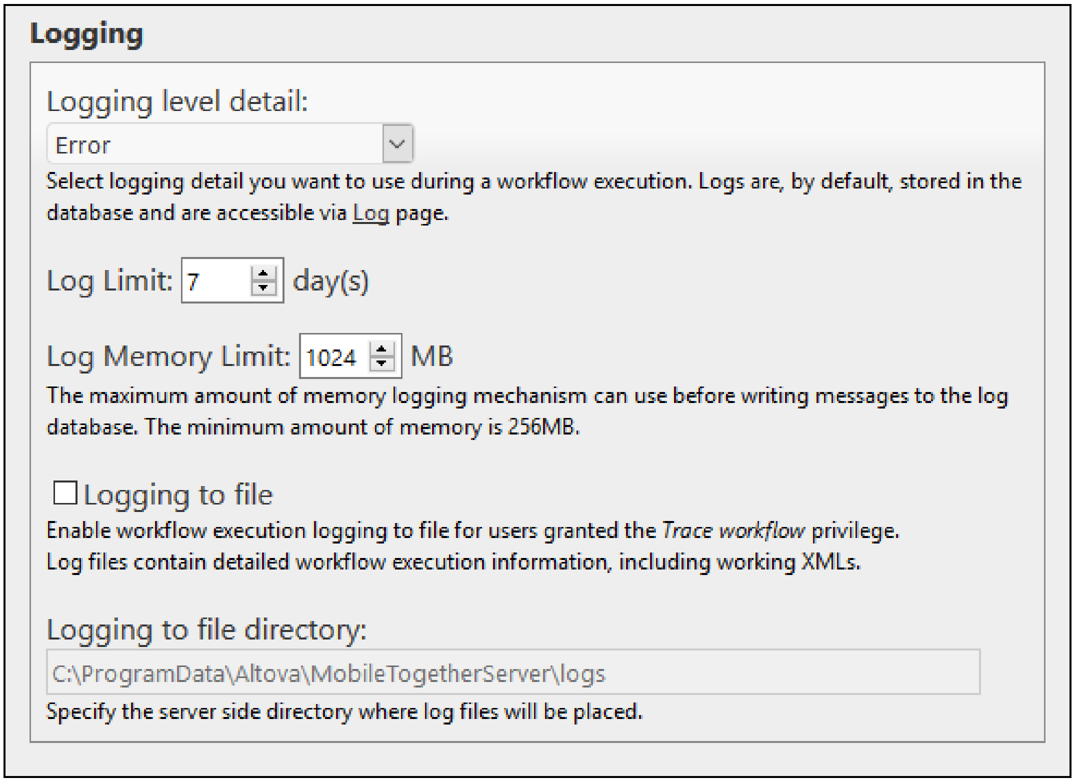
Certificate:

No file selected.

Logging

Logs contain reports of workflow activity, and they are displayed in the Log tab of the Web

UI. The settings in this pane define logging parameters.



Logging

Logging level detail:

Error

Select logging detail you want to use during a workflow execution. Logs are, by default, stored in the database and are accessible via Log page.

Log Limit: 7 day(s)

Log Memory Limit: 1024 MB

The maximum amount of memory logging mechanism can use before writing messages to the log database. The minimum amount of memory is 256MB.

□ Logging to file

Enable workflow execution logging to file for users granted the *Trace workflow* privilege.
Log files contain detailed workflow execution information, including working XMLs.

Logging to file directory:

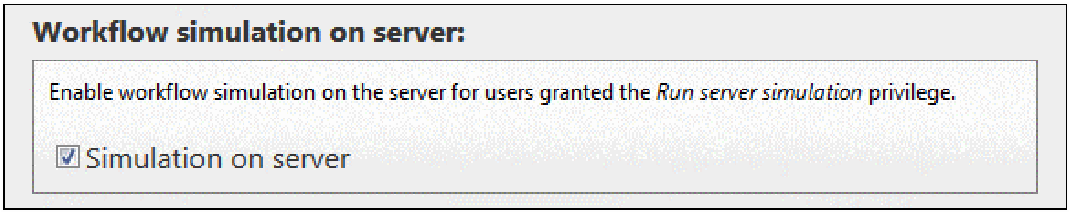
C:\ProgramData\Altova\MobileTogetherServer\logs

Specify the server side directory where log files will be placed.

- *Logging level detail:* The detail can vary from: (i) logging only errors, (ii) through logging errors and warnings, and (iii) (most detailed) logging errors, warnings, and information.
- *Log limit:* Specifies for how long logs are kept.
- *Log memory limit:* Writing messages to the log DB is given a lower priority than the executing of workflows. Consequently, messages are not written directly to the log DB, but are held in memory till a gap in workflow execution frees up processor time to write messages to the log DB. If, however, (i) there is no time to write messages to the log DB, and (ii) the amount of memory used for logging reaches the Log Memory Limit, then all the log messages in memory are discarded. A single log message then replaces the discarded messages; it records that the Log Memory was cleared. The Log Memory Limit option allows you to create more memory space (by specifying when to discard messages from memory) and so take the load off the server. Otherwise, the combination of processing load and memory load could end the MobileTogether Server process. Factors that affect your selection of the limit value will be: (i) the amount of memory on the machine, and (ii) the logging level detail. The lowest Log Memory Limit value you are allowed to enter is `256 MB`.
- *Log files:* Users that have been granted *Trace workflow* privileges can have logs saved to file if the *Logging to file* option is selected. The directory where logs are saved is specified in the *Logging to file directory* option.

Workflow simulation on server

Activating the check box allows workflow simulations to be performed on the server for users granted the *Run server simulation* privilege.



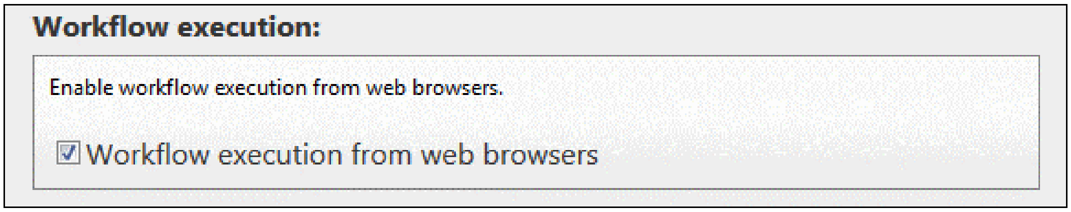
Workflow simulation on server:

Enable workflow simulation on the server for users granted the *Run server simulation* privilege.

☑ Simulation on server

Workflow execution

Activating the check box allows workflow execution to be performed from web browsers.



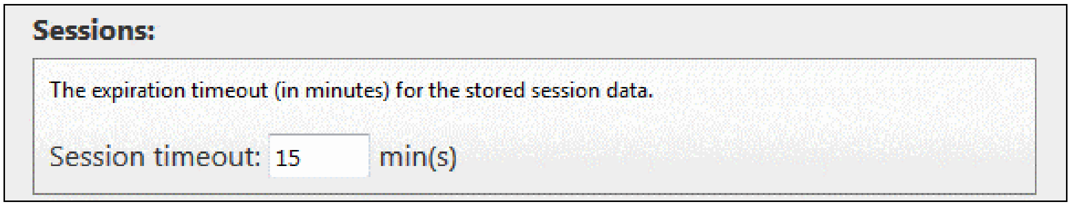
Workflow execution:

Enable workflow execution from web browsers.

☑ Workflow execution from web browsers

Sessions

Sets the timeout period in minutes before a new logon has to be performed. This timeout applies to both administrator access as well as client access.



Sessions:

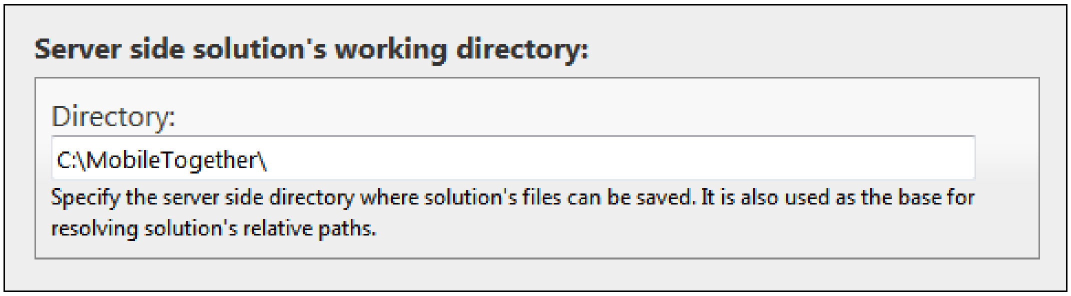
The expiration timeout (in minutes) for the stored session data.

Session timeout: 15 min(s)

Server side solution's working directory

When solutions are run on the server, this setting specifies the following:

- The base URI of all relative paths in the design. In a design, the paths of all files that are not deployed to the server will be resolved relative to the directory specified in this setting. For example, if a file in the design is addressed with a relative path of `MTSData\Test.xml`, then, if the file is not deployed, it must be located at: `<Working-Directory-Setting-Of-Server>MTSData\Test.xml`. (If the file is deployed to the server, the design uses internal mechanisms to access the files.)
- If, in the design, the file's location is given by an absolute path, then this path must point to a location inside a directory that is a sub-directory of the Working Directory specified in this setting. For example, if the file is addressed with the absolute path: `C:\MTSData\Test.xml`, then the file will be accessed only if the Working Directory is `C:\` or `C:\MTSData`.



Server side solution's working directory:

Directory:

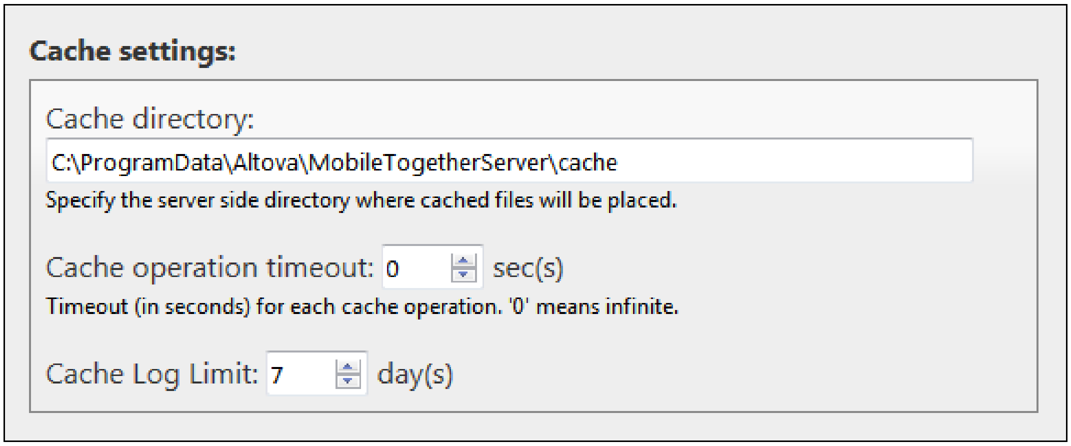
C:\MobileTogether\

Specify the server side directory where solution's files can be saved. It is also used as the base for resolving solution's relative paths.

The Working Directory setting, in effect, restricts any read/write access to local files during execution of solutions. Only files inside the Working Directory or any of its sub-directories can be accessed by MobileTogether Server for the execution of solutions.

Cache settings

Specifies: (i) the directory where cache files are saved, (ii) the timeout for each cache operation, and (iii) the duration in days for how long cache log items are displayed. See the Cache tab for more information.



Cache settings:

Cache directory:

C:\ProgramData\Altova\MobileTogetherServer\cache

Specify the server side directory where cached files will be placed.

Cache operation timeout: 0 sec(s)

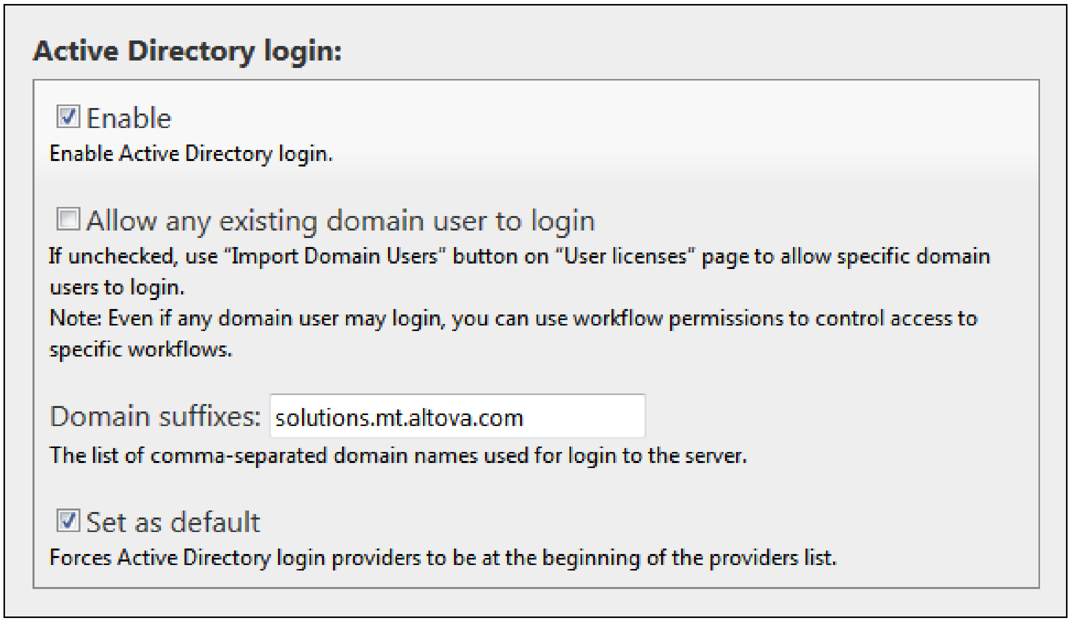
Timeout (in seconds) for each cache operation. '0' means infinite.

Cache Log Limit: 7 day(s)

Active Directory login

Enables users to log in to the server with their domain-specific user names and passwords. Each domain-specific user can be imported as a user in the Users tab, and then be assigned roles or privileges in the usual way. In the Options dialog of MobileTogether Designer, the type of user login for connecting to the server can be specified: either as a direct server-defined user or as a domain user.

After enabling Active Directory login, enter the domain or domains you wish to allow. Then go to the Users and Roles | Users tab to import specific users as MobileTogether Server users. These users can now use their domain-specific login information to log in to MobileTogether Server.



Active Directory login:

☑ Enable
Enable Active Directory login.

□ Allow any existing domain user to login
If unchecked, use "Import Domain Users" button on "User licenses" page to allow specific domain users to login.
Note: Even if any domain user may login, you can use workflow permissions to control access to specific workflows.

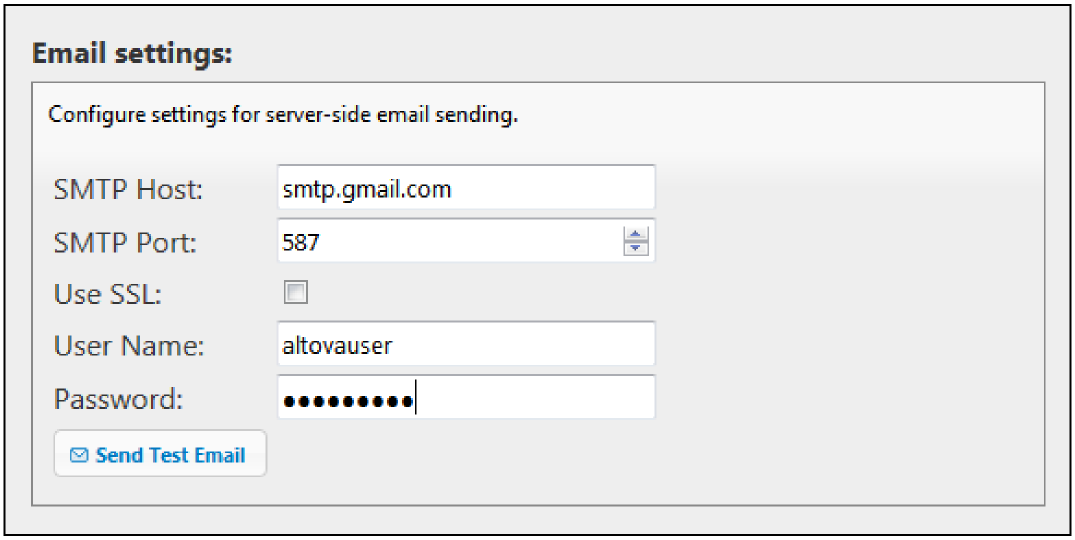
Domain suffixes: solutions.mt.altova.com
The list of comma-separated domain names used for login to the server.

☑ Set as default
Forces Active Directory login providers to be at the beginning of the providers list.

- *Allow any existing domain user to log in:* All users in the domain can log in to MobileTogether Server. If unchecked, domain users will need to be imported individually as MobileTogether Server users. This import is done via the **Import Users** button of the Users tab.
- *Domain suffixes:* Enter the domains to include, separating domains from one another with a comma.

Email settings

These settings enable emails to be sent by the end user via the server. Typically, the solution will provide an event that triggers a Send Email action that has been defined to send the email from the server. In order for the server to do this, it needs to access the SMTP server of the email service provider (usually your ISP). The settings for the SMTP server are entered in this pane (*screenshot below*).



Email settings:

Configure settings for server-side email sending.

SMTP Host: smtp.gmail.com

SMTP Port: 587

Use SSL: □

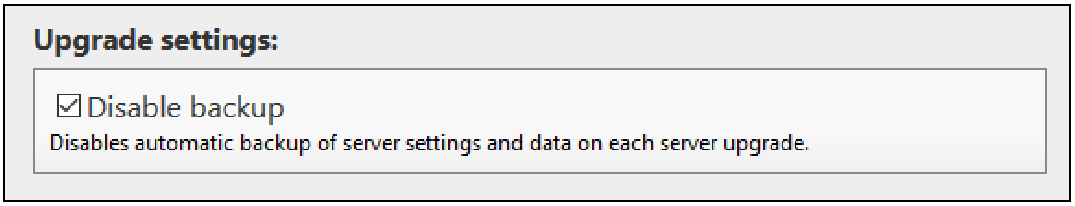
User Name: altovauser

Password: ●●●●●●●●●

- *SMTP Host and SMTP Port:* These are the SMTP host name and SMTP port of your ISP's SMTP server. These details are provided to you by your ISP.
- *User Name and Password:* The user name and password of an email account that is registered with the email service provider.

Upgrade settings

This setting (*screenshot below*) concerns a procedure related to upgrades of your MobileTogether Server from one version to a higher version. By default, a backup folder containing all important server files and folders is created when a new MobileTogether Server version is installed. When you de-install an existing MobileTogether Server installation, these MobileTogether Server files and folders are still held in the system. Subsequently, when a new MobileTogether Server package is installed, this data is copied into a backup folder that is created in the MobileTogether Server application folder.



Upgrade settings:

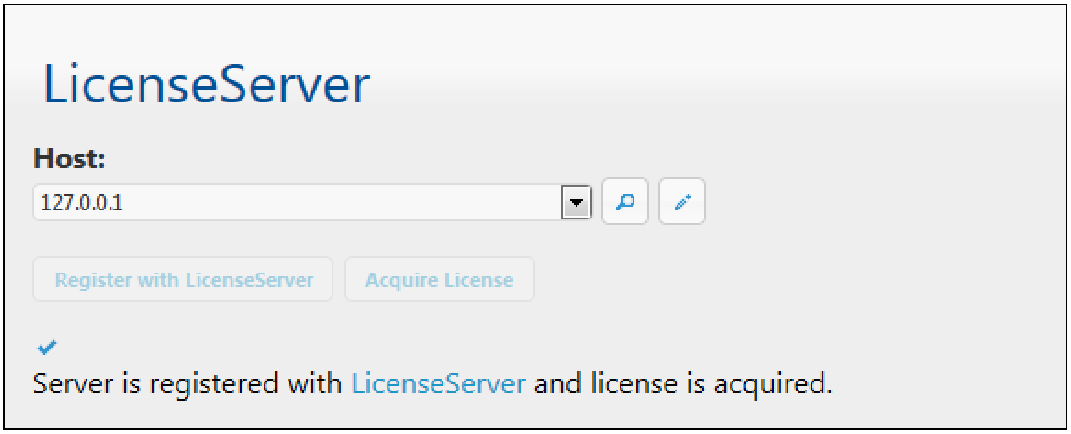
☑ Disable backup

Disables automatic backup of server settings and data on each server upgrade.

This setting enables you to disable the automatic backup for the next upgrade. You can always manually create a backup folder at any time. See the section How to Back Up and Restore MobileTogether Server for information about how to do this.

LicenseServer

MobileTogether Server must be registered with an Altova LicenseServer on your network. The LicenseServer settings specify the LicenseServer machine to connect to, and enables you to register MobileTogether Server with LicenseServer. See the section, Setting Up MobileTogether Server for information about licensing.



✓
Server is registered with LicenseServer and license is acquired.

- To search for LicenseServers on your network, click the **Search** button. The detected LicenseServers are listed in the dropdown list of the combo box. From this list, select the LicenseServer that you wish to connect to.
- To enter a server address, click the **Address** button, and enter the server address.

When the LicenseServer is found, **Register with LicenseServer** becomes enabled. Click the button to register MobileTogether Server with LicenseServer. Click **Acquire License** to go to LicenseServer and assign a license to MobileTogether Server.

Chapter 6

Command Line Usage

6 Command Line Usage

This section:

- Default location of MobileTogether Server executable
- Usage and list of CLI commands

Default location of MobileTogether Server executable

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

Windows	*<ProgramFilesFolder>*\Altova\MobileTogetherServer3.2\bin\MobileTogetherServer.exe
Linux	/opt/Altova/MobileTogetherServer3.2/bin/**mobiletogetherserver**
Mac	/usr/local/Altova/MobileTogetherServer3.2/bin/**mobiletogetherserver**

Usage and list of CLI commands

General command line syntax for `MobileTogetherServer` is:

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

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

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

▼ Casing and slashes on the command line

MobileTogetherServer *on Windows*
mobiletogetherserver *on Unix (Linux, Mac)*

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

where

<table>
<tr><td><code>--h | --help</code></td><td>Displays the help text.</td></tr>
</table>

`--version`	Displays the version of MobileTogether Server.

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

`licenseserver`	Register MobileTogether Server with LicenseServer on the local network.
`assignlicense`	Upload a license to LicenseServer and assign it to MobileTogether Server on this machine.
`verifylicense`	Check if the current MobileTogether Server is licensed; optionally, check if a given license key is assigned.
`install`	Installs MobileTogether Server as a service.
`start`	Starts MobileTogether Server as a service.
`uninstall`	Uninstalls MobileTogether Server as a service.
`debug`	Starts MobileTogether Server for debugging (not as a service).
`upgradedb`	Upgrades the internal MobileTogether Server DB to the structure of the new version, and enters correct default values where necessary.
`exportresourcestrings`	Exports all application resource strings to an XML file.
`setdeflang | sdl`	Sets the default language.
`help`	Displays help for a specific command. For example: `help generate`
`version`	Displays the version and exits.

6.1 licenseserver

This section:

- Command syntax
- Command description
- Examples
- Command options

Command syntax

Windows	**`MobileTogetherServer licenseserver [options]`** ***`Server-Or-IP-Address`***
Linux	**`mobiletogetherserver licenseserver [options]`** ***`Server-Or-IP-Address`***
Mac	**`mobiletogetherserver licenseserver [options]`** ***`Server-Or-IP-Address`***

Command description

On execution, the `licenseserver` command registers MobileTogether 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 MobileTogether Server with LicenseServer.

Once MobileTogether 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 MobileTogether Server a license. *For details, see the LicenseServer documentation*.

Examples

Examples of the `licenseserver` command:

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

The commands above specifiy, respectively, the machine named `DOC.altova.com`, and the user's machine (`localhost` and `127.0.0.1`) as the machine running Altova LicenseServer. In each case, the command registers MobileTogether Server with the LicenseServer on the machine specified.

Casing and slashes on the command line

`MobileTogetherServer` *on Windows*
`mobiletogetherserver` *on Unix (Linux, Mac)*

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

Command options

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

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

6.2 assignlicense (Windows only)

This section:

- Command syntax
- Command description
- Examples
- Command options

Command syntax

Windows **`MobileTogetherServer 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 MobileTogether 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 MobileTogether Server. *For details about licensing, see the LicenseServer documentation*.

Examples

Examples of the `assignlicense` command:

```
mobiletogetherserver assignlicense C:\MobileTogether
Server12345.altova_licenses
mobiletogetherserver assignlicense --test-only=true C:\MobileTogether
Server12345.altova_licenses
```

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

▼ Casing and slashes on the command line

`MobileTogetherServer` *on Windows*
`mobiletogetherserver` *on Unix (Linux, Mac)*

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

Command options

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

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

6.3 verifylicense (Windows only)

This section:

- Command syntax
- Command description
- Examples
- Command options

Command syntax

Windows **`MobileTogetherServer 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:

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

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

▼ Casing and slashes on the command line

`MobileTogetherServer` *on Windows*
`mobiletogetherserver` *on Unix (Linux, Mac)*

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

Command options

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

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

6.4 install

This section:

- Command syntax
- Command description
- Examples

Command syntax

Windows	**`MobileTogetherServer install`**
Linux	**`mobiletogetherserver install`**
Mac	**`mobiletogetherserver install`**

Command description

The `install` command installs MobileTogether Server as a service on the server machine. The command takes no options.

Note that installing MobileTogether Server as a service, does not automatically start the service. To start the service, use the `start` command. To uninstall MobileTogether Server as a service, use the `uninstall` command.

Examples

Examples of the `install` command:

```
mobiletogetherserver install
```

▼ Casing and slashes on the command line

`MobileTogetherServer` *on Windows*
`mobiletogetherserver` *on Unix (Linux, Mac)*

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

6.5 start

This section:

- Command syntax
- Command description
- Examples

Command syntax

Windows	**`MobileTogetherServer start`**
Linux	**`mobiletogetherserver start`**
Mac	**`mobiletogetherserver start`**

Command description

The `start` command starts MobileTogether Server as a service on the server machine. The command takes no options.

Note that, if MobileTogether Server is not installed as a service, you can install it with the `install` command. To uninstall MobileTogether Server as a service, use the `uninstall` command.

Examples

Examples of the `start` command:

```
mobiletogetherserver start
```

▼ Casing and slashes on the command line

`MobileTogetherServer` *on Windows*
`mobiletogetherserver` *on Unix (Linux, Mac)*

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

6.6 uninstall

This section:

- Command syntax
- Command description
- Examples

Command syntax

Windows	**`MobileTogetherServer uninstall`**
Linux	**`mobiletogetherserver uninstall`**
Mac	**`mobiletogetherserver uninstall`**

Command description

The `uninstall` command uninstalls MobileTogether Server as a service on the server machine. The command takes no options.

To re-install MobileTogether Server as a service, use the `install` command.

Examples

Examples of the `uninstall` command:

```
mobiletogetherserver uninstall
```

▼ Casing and slashes on the command line

`MobileTogetherServer` *on Windows*
`mobiletogetherserver` *on Unix (Linux, Mac)*

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

6.7 debug

This section:

- Command syntax
- Command description
- Examples

Command syntax

Windows	**`MobileTogetherServer debug`**
Linux	**`mobiletogetherserver debug`**
Mac	**`mobiletogetherserver debug`**

Command description

The **debug** command starts MobileTogether Server for debugging—not as a service. To stop MobileTogether Server in this mode, press **Ctrl+C**. The command takes no options.

Examples

Examples of the `debug` command:

```
mobiletogetherserver debug
```

▼ Casing and slashes on the command line

`MobileTogetherServer` *on Windows*
`mobiletogetherserver` *on Unix (Linux, Mac)*

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

6.8 upgradedb

This section:

- Command syntax
- Command description
- Examples

Command syntax

Windows	**`MobileTogetherServer upgradedb`**
Linux	**`mobiletogetherserver upgradedb`**
Mac	**`mobiletogetherserver upgradedb`**

Command description

The `upgradedb` command updates the structure of the internal MobileTogether Server database to that of the new MobileTogether Server version and inserts correct default values. This is necessary if the structure of the DB changes from one version of MobileTogether Server to the next. The DB structure needs to be updated in order for the new version to work with the existing data. The command takes no options.

Examples

Examples of the `upgradedb` command:

```
mobiletogetherserver upgradedb
```

▼ Casing and slashes on the command line

`MobileTogetherServer` *on Windows*
`mobiletogetherserver` *on Unix (Linux, Mac)*

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

6.9 exportresourcestrings

This section:

Command syntax

Windows

```
MobileTogetherServer exportresourcestrings LanguageCode XMLOutputFile
```

Linux

```
mobiletogetherserver exportresourcestrings LanguageCode XMLOutputFile
```

Mac

```
mobiletogetherserver 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 MobileTogether 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:

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

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

Casing and slashes on the command line

`MobileTogetherServer` *on Windows*
`mobiletogetherserver` *on Unix (Linux, Mac)*

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

Creating localized versions of MobileTogether Server

You can create a localized version of MobileTogether 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\MobileTogetherServer3.2\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 MobileTogether 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 `MobileTogetherServer2017_lc.dll`. The `_lc` part of the name contains the language code. For example, in `MobileTogetherServer2017_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 MobileTogether Server application to use. For the argument of the `setdeflang` command, use the language code that is part of the DLL name.

Note: Altova MobileTogether 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 MobileTogether Server's `setdeflang` command.

6.10 setdeflang

This section:

- Command syntax
- Command description
- Examples
- Supported languages

Command syntax

Windows	**`MobileTogetherServer setdeflang	sdl`** ***`LangaugeCode`***
Linux	**`mobiletogetherserver setdeflang	sdl`** ***`LangaugeCode`***
Mac	**`mobiletogetherserver setdeflang	sdl`** ***`LangaugeCode`***

Command description

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

Examples

An example of the `setdeflang` command:

```
mobiletogetherserver setdeflang DE
```

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

▼ Casing and slashes on the command line

`MobileTogetherServer` *on Windows*
`mobiletogetherserver` *on Unix (Linux, Mac)*

* Note that lowercase (`mobiletogetherserver`) works on all platforms (Windows, Linux, and Mac), while upper-lower (`MobileTogetherServer`) 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

6.11 help

This section:

- Command syntax
- Command description
- Examples
- The --help option

Command syntax

Windows	**`MobileTogetherServer help`** ***`Command`***
Linux	**`mobiletogetherserver help`** ***`Command`***
Mac	**`mobiletogetherserver 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:

```
mobiletogetherserver 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

`MobileTogetherServer` *on Windows*
`mobiletogetherserver` *on Unix (Linux, Mac)*

* Note that lowercase (`mobiletogetherserver`) works on all platforms (Windows, Linux, and Mac), while upper-lower (`MobileTogetherServer`) 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:

```
  mobiletogetherserver exportresourcestrings --help
```

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

```
  mobiletogetherserver help exportresourcestrings
```

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

6.12 version

This section:

- Command syntax
- Command description
- Examples

Command syntax

Windows	**`MobileTogetherServer version`**
Linux	**`mobiletogetherserver version`**
Mac	**`mobiletogetherserver version`**

Command description

The `version` command displays the version number of MobileTogether Server and exits. The command takes no options.

Examples

Examples of the `version` command:

```
mobiletogetherserver version
```

▼ Casing and slashes on the command line

`MobileTogetherServer` *on Windows*
`mobiletogetherserver` *on Unix (Linux, Mac)*

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

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: 2.3

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: 2.3

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: 04 May 2017

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.

7.2 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 Vista, Windows 7/8/10

▾ *Windows Server*

Windows Server 2008 R2 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: 2.3

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.

7.3 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
MobileTogether Server	qt4, krb5-libs, qt-x11	libgssapi-krb5-2	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 --list | 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 with the following command:

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

The LicenseServer package will be installed in:

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

A user named **`altovalicenseserver`** will be created. **LicenseServer must be accessed as this user**. This can be done, for example on CentOS 7, in the following way:

```
sudo runuser -l altovalicenseserver -c '/opt/Altova/LicenseServerOffline/bin/
licenseserveroffline debug'
```

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: 2.3

7.4 Installation (macOS)

Altova LicenseServer can be installed on (Mac) OS X, macOS systems (version 10.8 or higher). Since you might need to uninstall a previous version, uninstalling is described first.

System requirements

▾ *(Mac) OS X, macOS*

OS X 10.10, 10.11, macOS 10.12 or newer
Java for OS X, macOS (latest version)

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.

The LicenseServer package will be installed in the folder:

```
/usr/local/Altova/LicenseServer
```

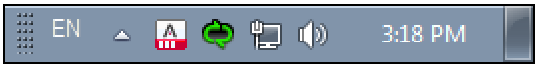
A user named **`altovalicenseserver`** will be created. **LicenseServer must be accessed as this user**.

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

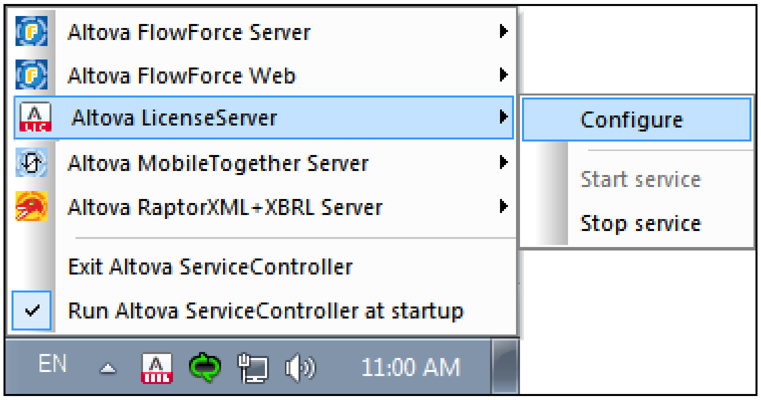
7.5 Altova ServiceController

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**.


Altova FlowForce Server
Altova FlowForce Web
Altova LicenseServer
Altova MobileTogether Server
Altova RaptorXML+XBRL Server
Exit Altova ServiceController
✓ Run Altova ServiceController at startup
Configure
Start service
Stop service
EN
11:00 AM

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.

7.6 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.

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.

Start LicenseServer

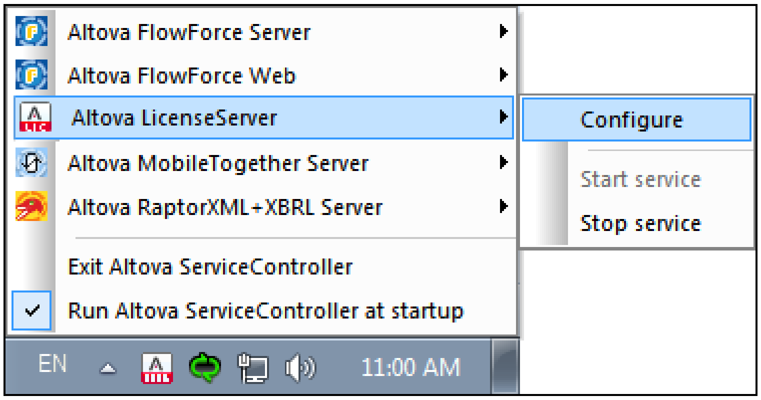
This section:

- 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.


Altova FlowForce Server
Altova FlowForce Web
Altova LicenseServer
Altova MobileTogether Server
Altova RaptorXML+XBRL Server
Exit Altova ServiceController
Run Altova ServiceController at startup
Configure
Start service
Stop service
EN
11:00 AM

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.)

---

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
```

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.

Open LicenseServer's Config Page (Windows)

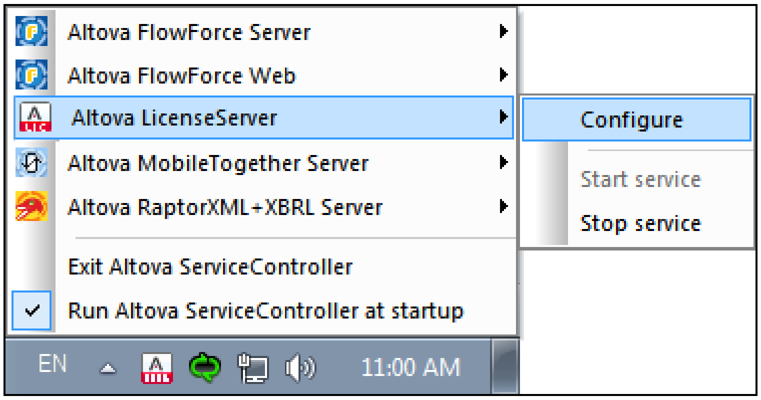
This section:

- 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.


Altova FlowForce Server
Altova FlowForce Web
Altova LicenseServer
Altova MobileTogether Server
Altova RaptorXML+XBRL Server
Exit Altova ServiceController
Run Altova ServiceController at startup
Configure
Start service
Stop service
EN
11:00 AM

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:

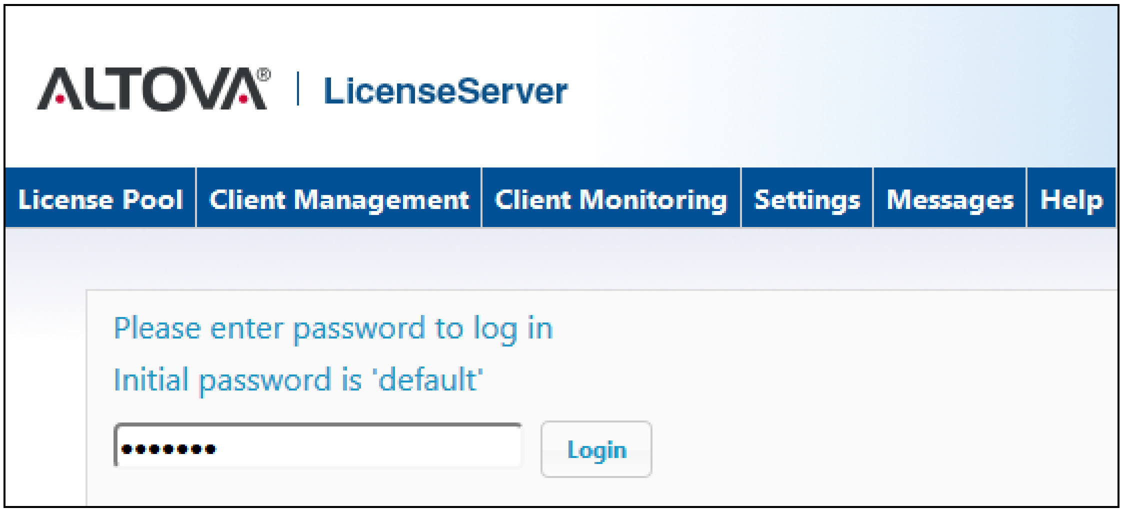
```
C:/ProgramData/Altova/LicenseServer/WebUI.html
```

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.



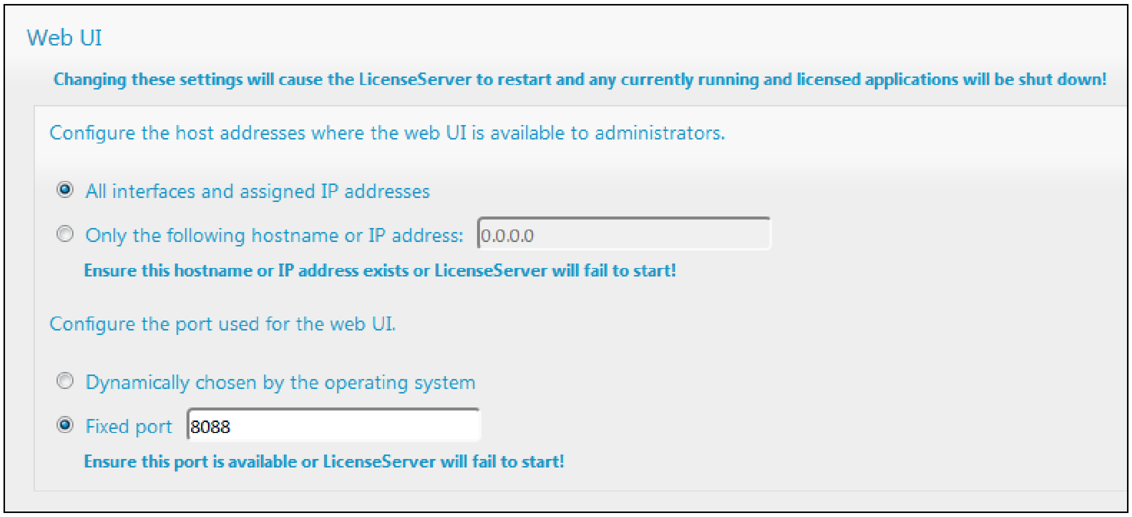
Please enter password to log in

Initial password is 'default'

•••••••

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: 0.0.0.0

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 8088

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.

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:

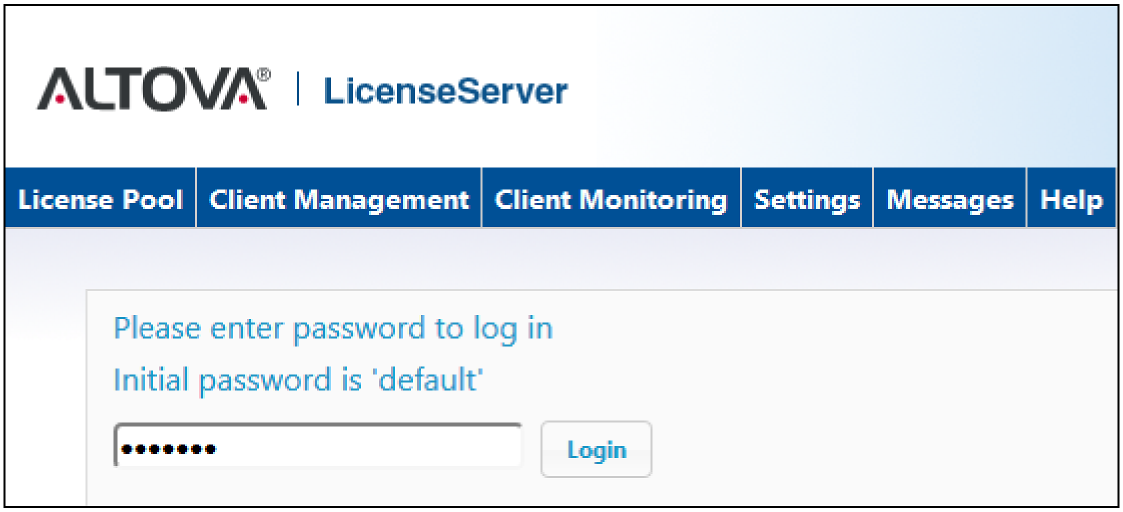
```
/var/opt/Altova/LicenseServer/webUI.html
```

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.



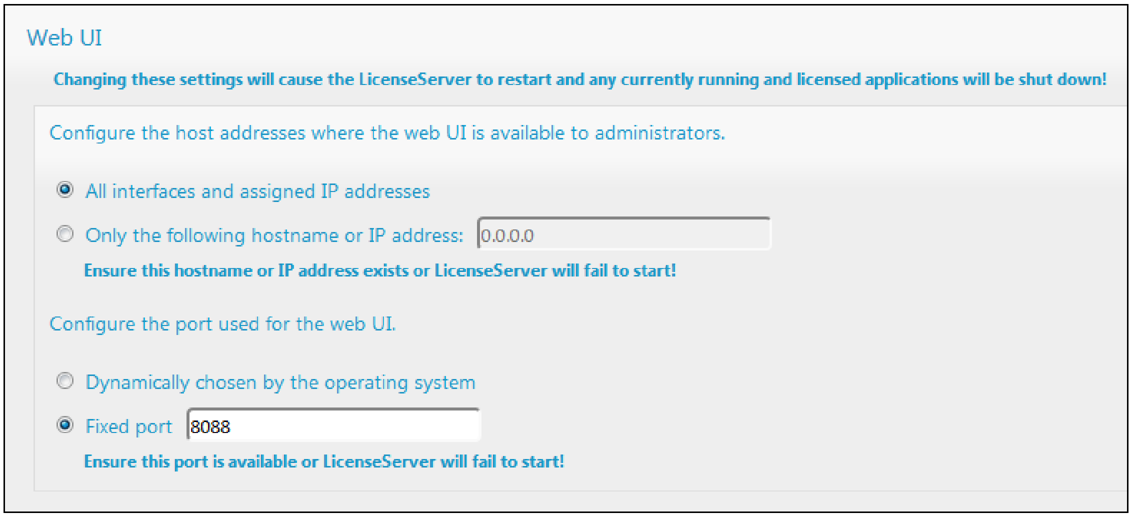
Please enter password to log in

Initial password is 'default'

•••••••

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)).



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: 0.0.0.0

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 8088

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.

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
```

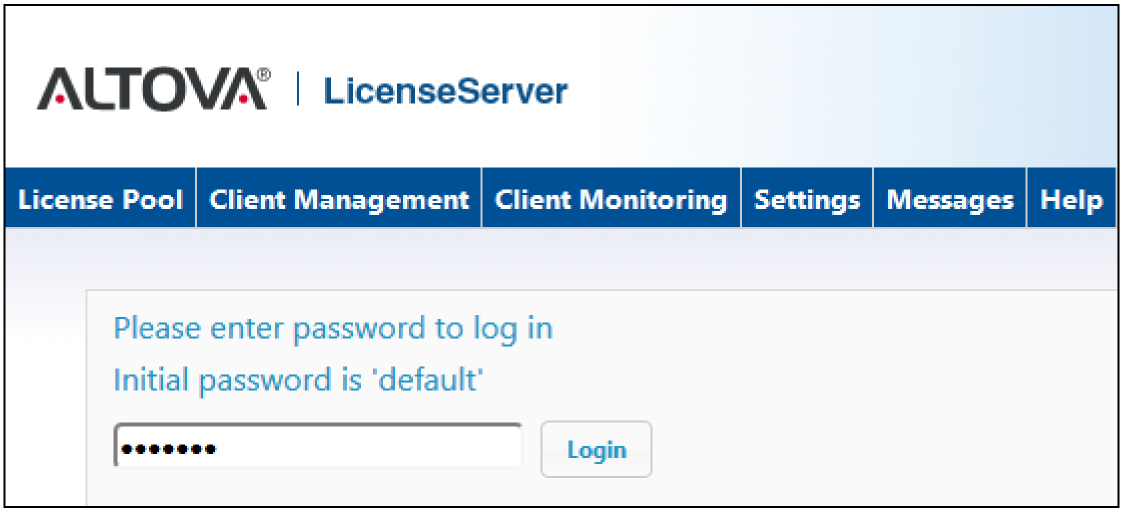
If you have set the URL of the Configuration page to be generated dynamically (in the Settings tab of the Configuration page), then a new URL is generated each time LicenseServer is started. You will need to check the current version of **`webUI.html`** to find out the current URL of the Configuration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



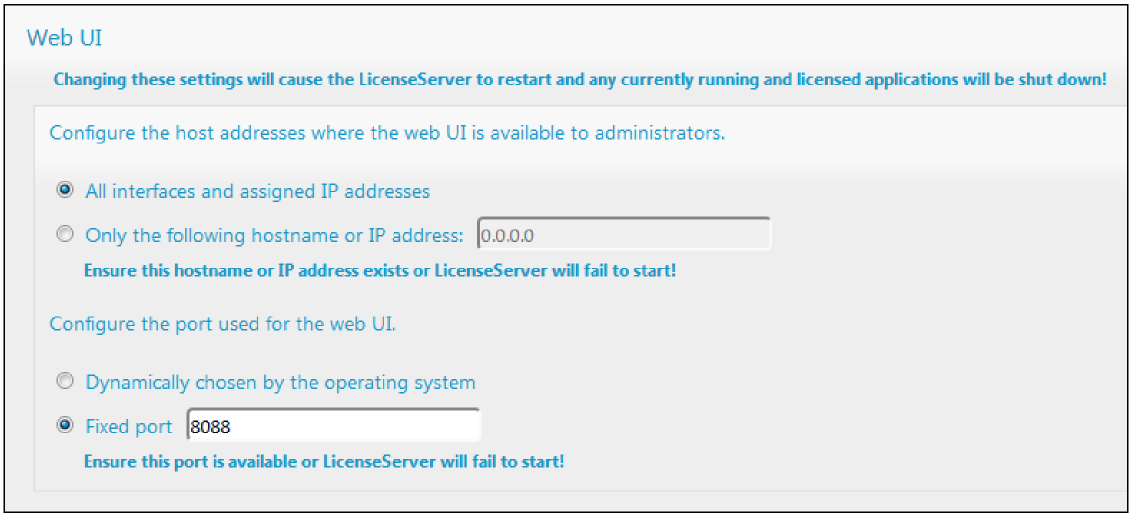
Please enter password to log in

Initial password is 'default'

•••••••

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: 0.0.0.0

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 8088

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.

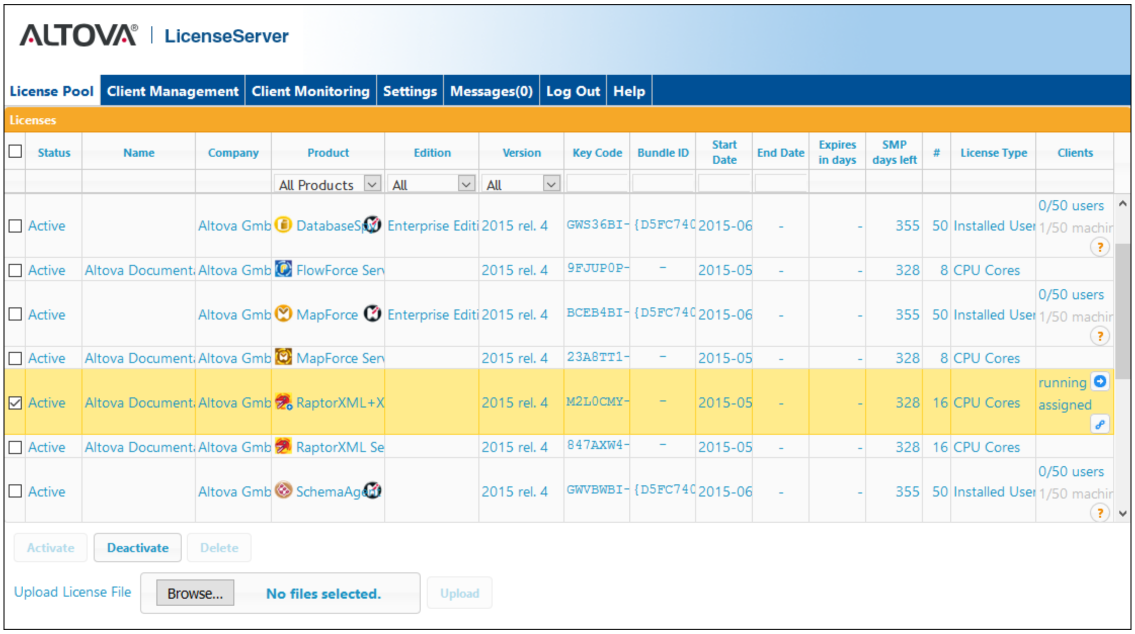
Upload Licenses to LicenseServer

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 from Altova, you must upload it to the Altova LicenseServer. Each license file can contain one or more licenses and depends on your purchase. When you upload a license 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*).



□	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									
□	Active		Altova Gmb	DatabaseSpy	Enterprise Editi	2015 rel. 4	GWS36BI-	{D5FC740	2015-06	-	-	355	50	Installed User	0/50 users 1/50 machin ?
□	Active	Altova Document	Altova Gmb	FlowForce Serv		2015 rel. 4	9FJUP0P-	-	2015-05	-	-	328	8	CPU Cores	
□	Active		Altova Gmb	MapForce	Enterprise Editi	2015 rel. 4	BCEB4BI-	{D5FC740	2015-06	-	-	355	50	Installed User	0/50 users 1/50 machin ?
□	Active	Altova Document	Altova Gmb	MapForce Serv		2015 rel. 4	23A8TT1-	-	2015-05	-	-	328	8	CPU Cores	
☑	Active	Altova Document	Altova Gmb	RaptorXML+X		2015 rel. 4	M2L0CMY-	-	2015-05	-	-	328	16	CPU Cores	running assigned
□	Active	Altova Document	Altova Gmb	RaptorXML Se		2015 rel. 4	847AXW4-	-	2015-05	-	-	328	16	CPU Cores	
□	Active		Altova Gmb	SchemaAgent		2015 rel. 4	GWVBWBI-	{D5FC740	2015-06	-	-	355	50	Installed User	0/50 users 1/50 machin ?

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 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* 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. (A product is allowed to have two active licenses assigned to it at a given time.)

- *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. Should you see a license showing up as *Blocked*, please contact Altova Support with your license information and any other relevant data.

These statuses are summarized in the table below:

Status	Meaning
Activating...	On upload, license information is sent to `altova.com` for verification. Refresh the browser to view the updated status. Verification and activation can take a few minutes.
Failed Verification	A connection to `altova.com` could not be made. After establishing a connection, either restart the service or activate the license (with the **Activate** button).
Active	Verification was successful, the license is active.
Inactive	Verification was successful, but the license is also on another LicenseServer on the network. Licenses can be made inactive with the **Deactivate** button.
Pending	A *Pending* license has a start date in the future, and it turns *Active* 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.
Blocked	Verification was not successful. License is invalid and is blocked. Contact Altova Support.

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.

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.

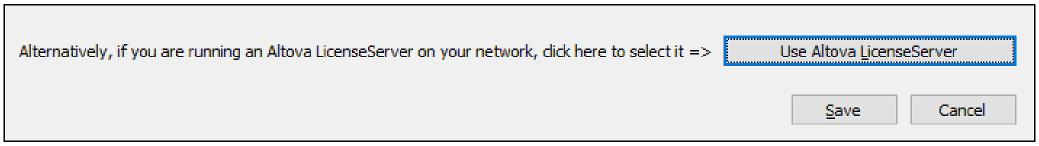
This section describes how to register different Altova products:

- Register Altova Desktop Products
- Register FlowForce Server
- Register MapForce Server
- Register MobileTogether Server
- Register RaptorXML(+XBRL) Server
- Register StyleVision Server

Register Altova Desktop Products

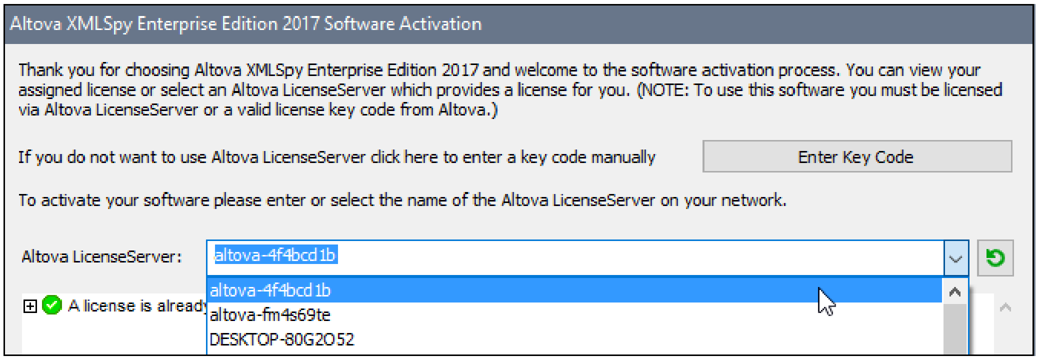
To register an Altova Desktop product with an Altova LicenseServer, do the following:

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*)



Alternatively, if you are running an Altova LicenseServer on your network, click here to select it =>

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



Altova XMLSpy Enterprise Edition 2017 Software Activation

Thank you for choosing Altova XMLSpy Enterprise Edition 2017 and welcome to the software activation process. You can view your assigned license or select an Altova LicenseServer which provides a license for you. (NOTE: To use this software you must be licensed via Altova LicenseServer or a valid license key code from Altova.)

If you do not want to use Altova LicenseServer click here to enter a key code manually

To activate your software please enter or select the name of the Altova LicenseServer on your network.

Altova LicenseServer: altova-4f4bcd1b

A license is alread

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 2017* | *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:

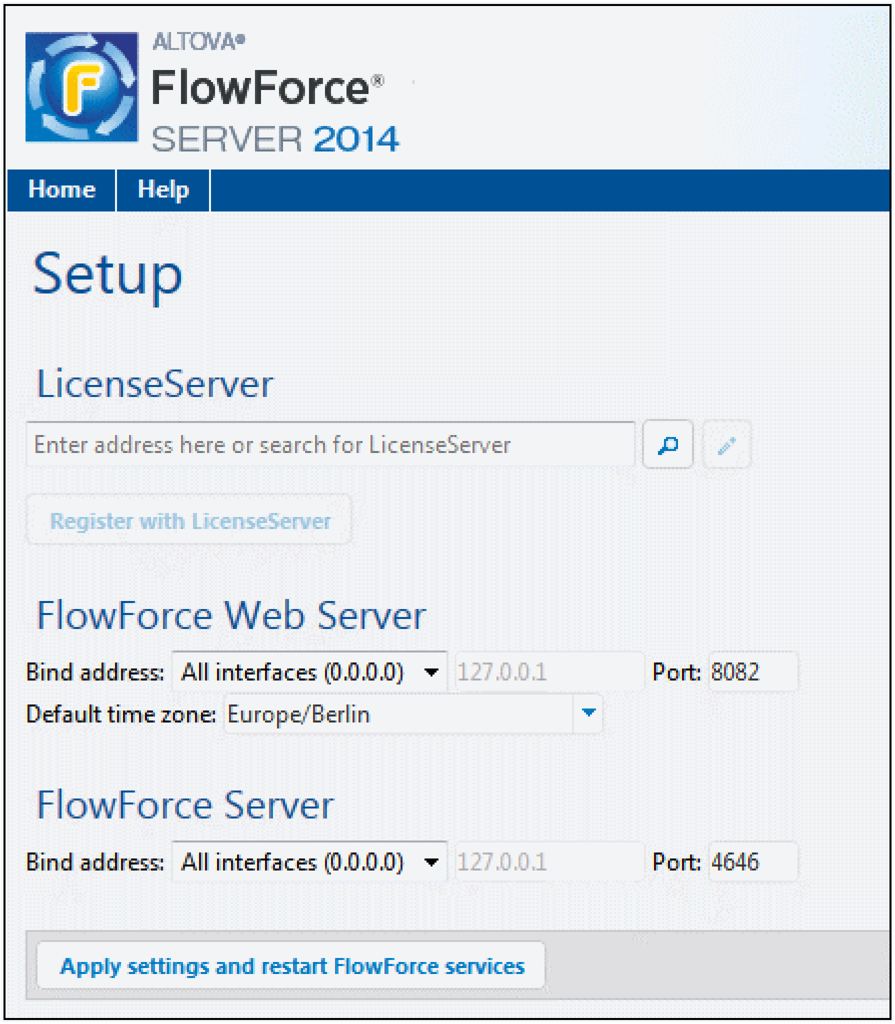
```
FlowForceWeb running on http://127.0.1.1:3459/setup?key=52239315203
```

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.



ALTOVA®
FlowForce®
SERVER 2014
Home
Help
Setup
LicenseServer
Enter address here or search for LicenseServer
Register with LicenseServer
FlowForce Web Server
FlowForce Server
Apply settings and restart FlowForce services

FlowForce Web Server

Bind address: All interfaces (0.0.0.0) 127.0.0.1 Port: 8082

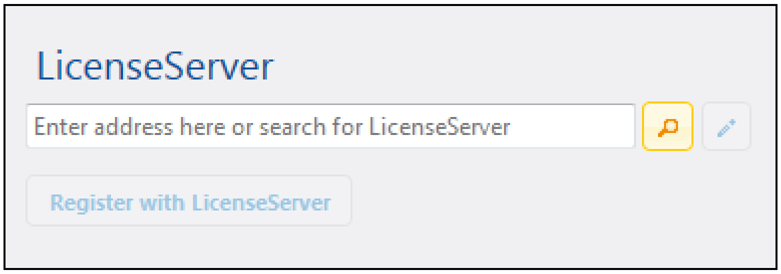
Default time zone: Europe/Berlin

FlowForce Server

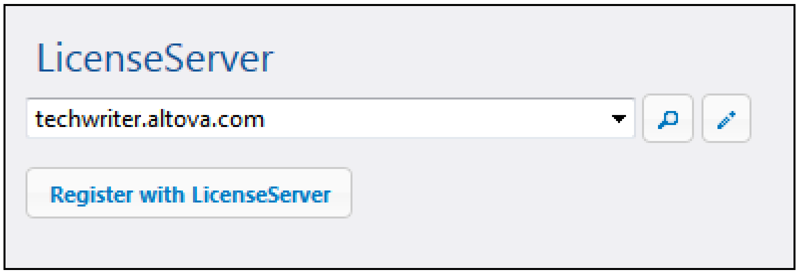
Bind address: All interfaces (0.0.0.0) 127.0.0.1 Port: 4646

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*).


LicenseServer
Enter address here or search for LicenseServer
Register with LicenseServer

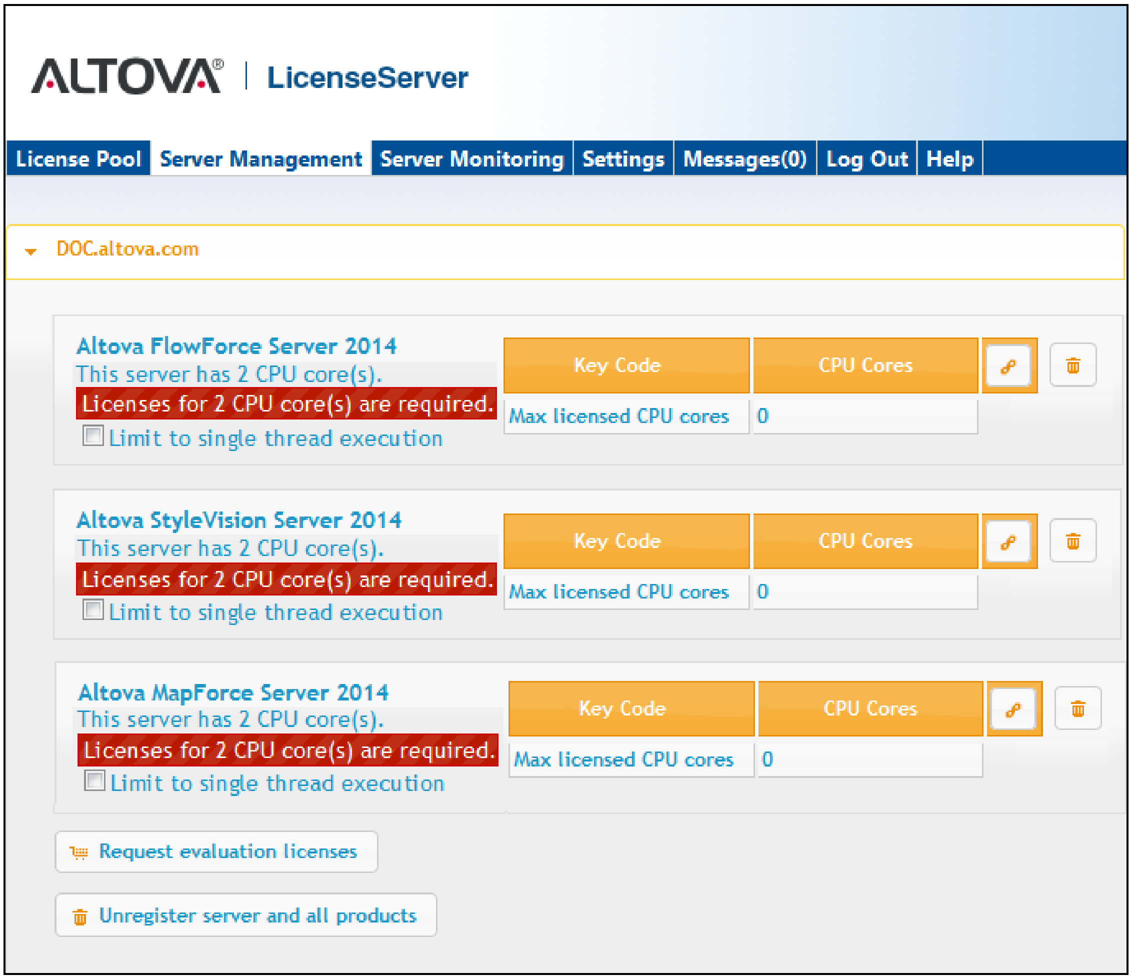
The search returns a list of available Altova LicenseServers on the network. One LicenseServer will be selected (*screenshot below*) and the others will be available in the dropdown list of the combo box. Select the LicenseServer on which your FlowForce license is stored.



- 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.



DOC.altova.com

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

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/FlowForceServer2017/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/FlowForceServer2017/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:

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/MapForceServer2017/bin/mapforceserver licenseserver localhost
```

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

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

After 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.

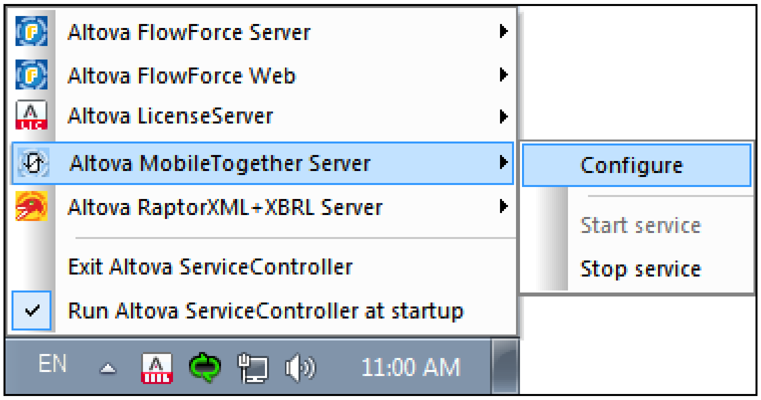
Next Steps

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.


Altova FlowForce Server
Altova FlowForce Web
Altova LicenseServer
Altova MobileTogether Server
Altova RaptorXML+XBRL Server
Exit Altova ServiceController
Run Altova ServiceController at startup
Configure
Start service
Stop service
EN
11:00 AM

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 Server:	**`RaptorXMLXBRL licenseserver`** ***`Server-Or-IP-Address`***

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

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

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/RaptorXMLServer2017/bin/raptorxmlserver licenseserver
localhost
sudo /opt/Altova/RaptorXMLXBRLServer2017/bin/raptorxmlxbrlserver 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/RaptorXMLServer2017/bin
/opt/Altova/RaptorXMLXBRLServer2017/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/StyleVisionServer2017/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/StyleVisionServer2017/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.

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

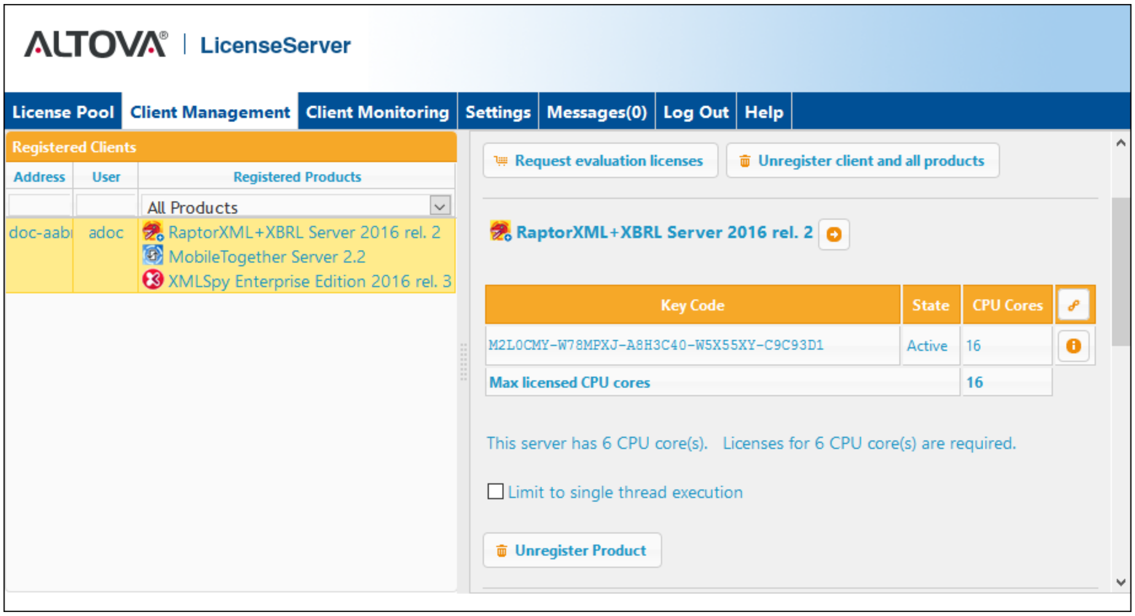
Before assigning a license

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.



Registered Clients

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

RaptorXML+XBRL Server 2016 rel. 2

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

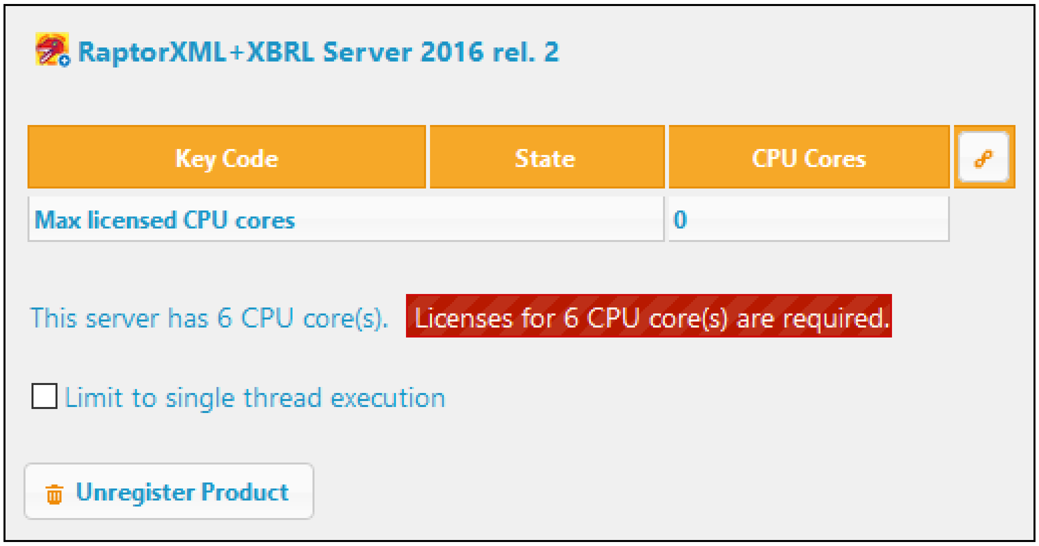
This server has 6 CPU core(s). Licenses for 6 CPU core(s) are required.

□ Limit to single thread execution

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

- 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.

ℹ *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.

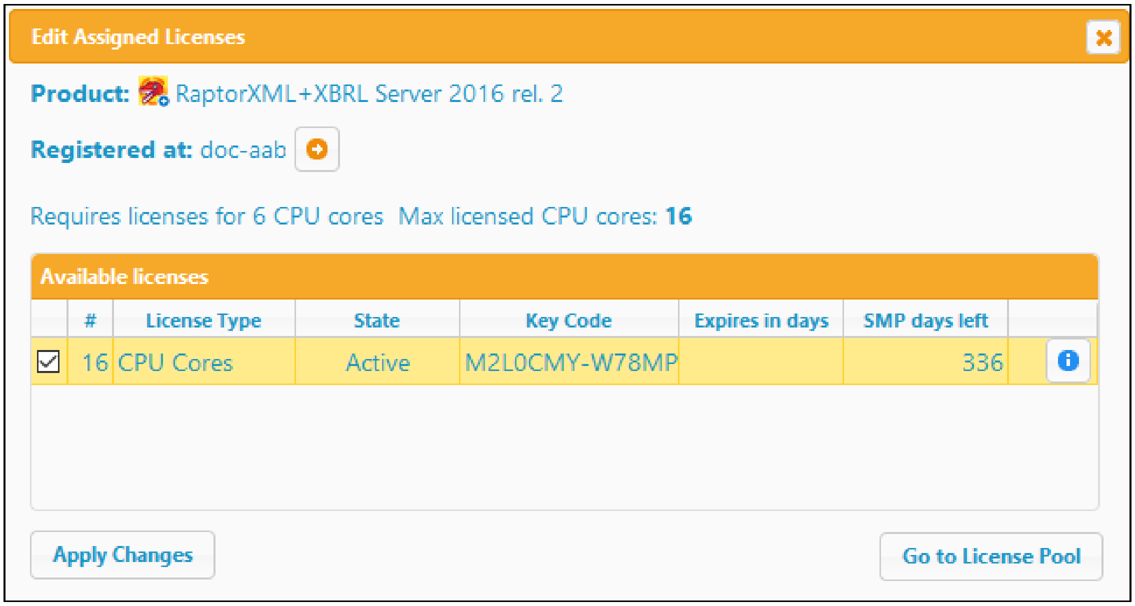
Note: Each Altova server product license can be used for only one client machine—the machine on which the Altova server product is installed—at a time, even if the license has unused licensing capacity. For example, if a 10-core license is used for a client machine that has 6 CPU cores, then the remaining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

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.

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	
☑	16	CPU Cores	Active	M2L0CMY-W78MP		336	

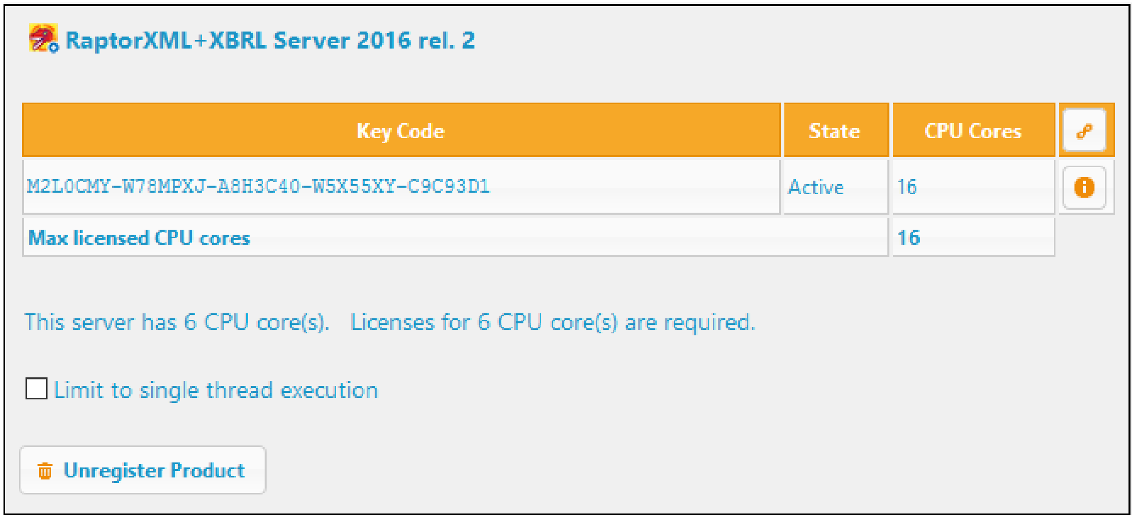
Note the following points about the licenses displayed in the Manage Licenses dialog:

- 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

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.

7.7 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

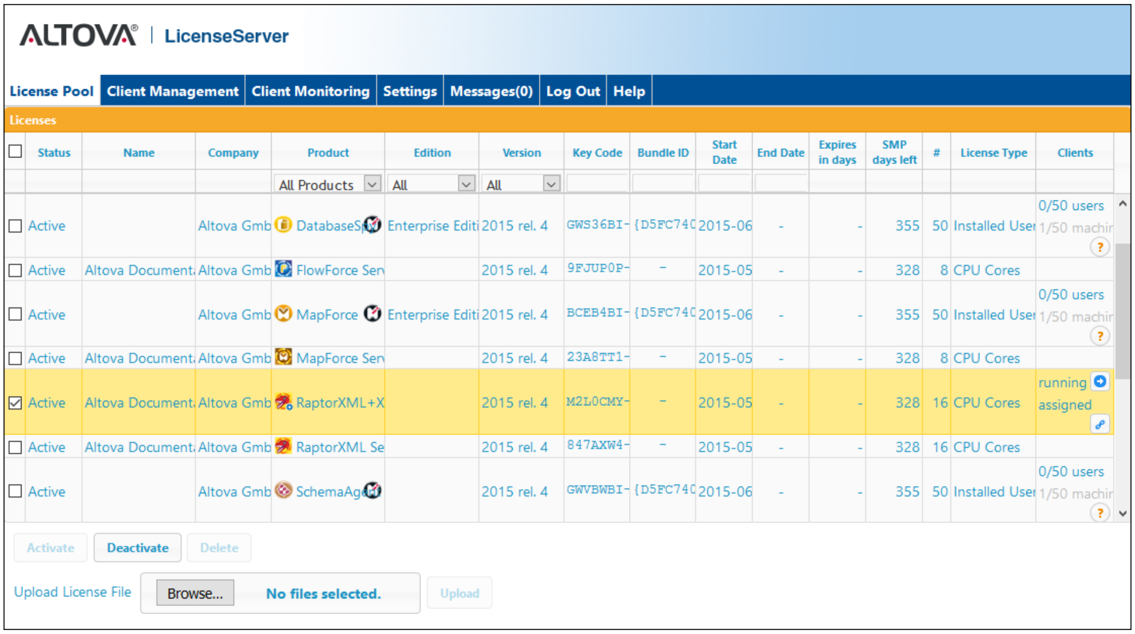
For a step-by-step guide of how to assign licenses with LicenseServer, see the section How to Assign Licenses.

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.



□	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									
□	Active		Altova Gmb	DatabaseSpy	Enterprise Editi	2015 rel. 4	GWS36BI-	{D5FC740	2015-06	-	-	355	50	Installed User	0/50 users 1/50 machir ?
□	Active	Altova Document:	Altova Gmb	FlowForce Serv		2015 rel. 4	9FJUP0P-	–	2015-05	-	-	328	8	CPU Cores	
□	Active		Altova Gmb	MapForce	Enterprise Editi	2015 rel. 4	BCEB4BI-	{D5FC740	2015-06	-	-	355	50	Installed User	0/50 users 1/50 machir ?
□	Active	Altova Document:	Altova Gmb	MapForce Serv		2015 rel. 4	23A8TT1-	–	2015-05	-	-	328	8	CPU Cores	
☑	Active	Altova Document:	Altova Gmb	RaptorXML+X		2015 rel. 4	M2L0CMY-	–	2015-05	-	-	328	16	CPU Cores	running assigned
□	Active	Altova Document:	Altova Gmb	RaptorXML Se		2015 rel. 4	847AXW4-	–	2015-05	-	-	328	16	CPU Cores	
□	Active		Altova Gmb	SchemaAgent		2015 rel. 4	GWVBWBI-	{D5FC740	2015-06	-	-	355	50	Installed User	0/50 users 1/50 machir ?

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* 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. (A product is allowed to have two active licenses assigned to it at a given time.)

- *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. Should you see a license showing up as *Blocked*, please contact Altova Support with your license information and any other relevant data.

These statuses are summarized in the table below:

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

Active	Verification was successful, the license is active.
Inactive	Verification was successful, but the license is also on another LicenseServer on the network. Licenses can be made inactive with the **Deactivate** button.
Pending	A *Pending* license has a start date in the future, and it turns *Active* 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.
Blocked	Verification was not successful. License is invalid and is blocked. Contact Altova Support.

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 licenses and desktop product licenses. It does not have entries for 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 MisisonKit 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. 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.

MobileTogether Server licenses

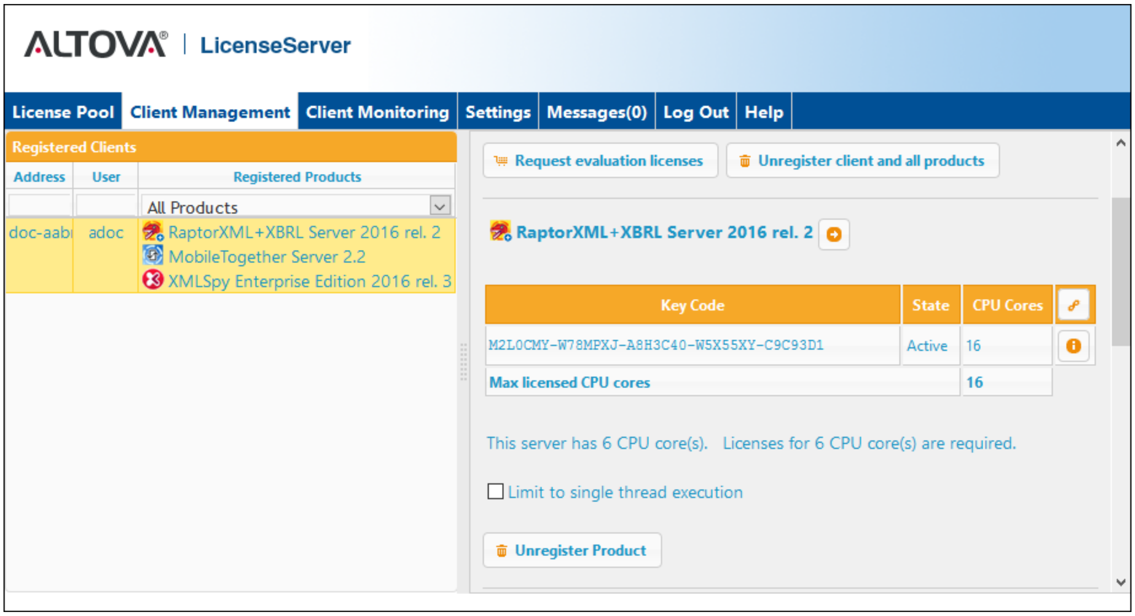
MobileTogether Server licenses are based on the number of CPU cores on the MobileTogether Server machine. Core licenses allow an unlimited number of MobileTogether Client devices to connect to the server. However, if you check the *Limit to single thread execution* check box, then only one mobile device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

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:



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

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

- *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.

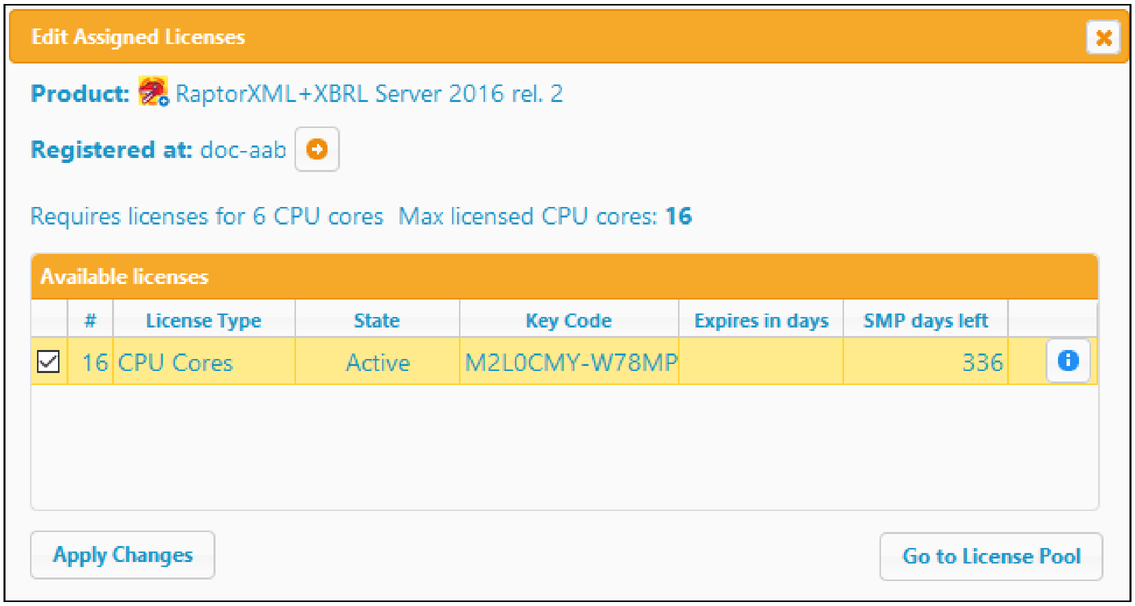
Managing licenses in the Product Licensing pane

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

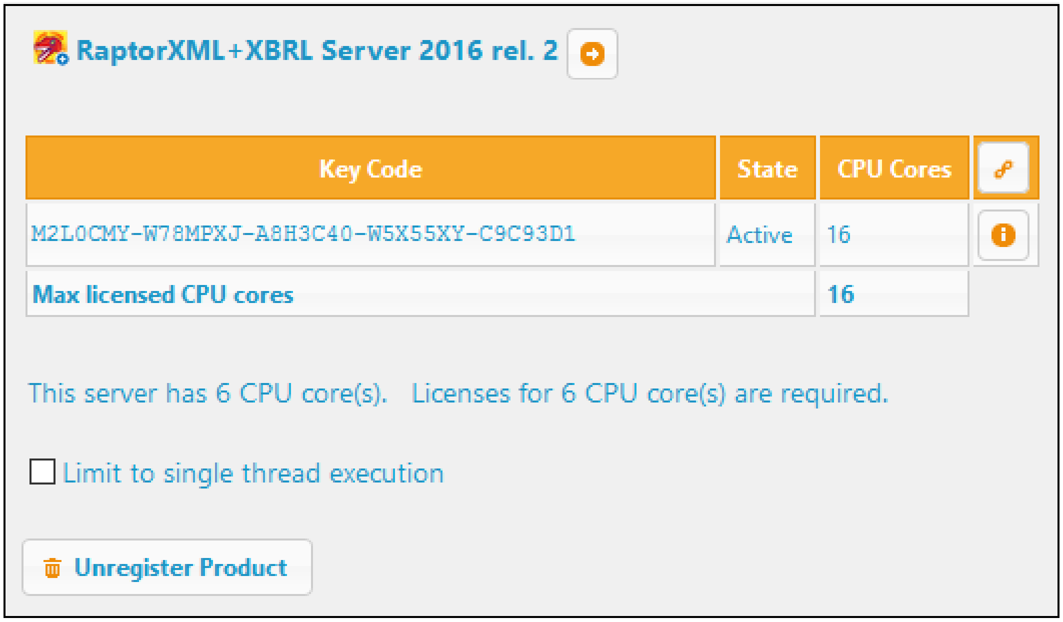
Registered at: doc-aab

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

Available licenses

	#	License Type	State	Key Code	Expires in days	SMP days left	
☑	16	CPU Cores	Active	M2L0CMY-W78MP		336	

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

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 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.

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.

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.

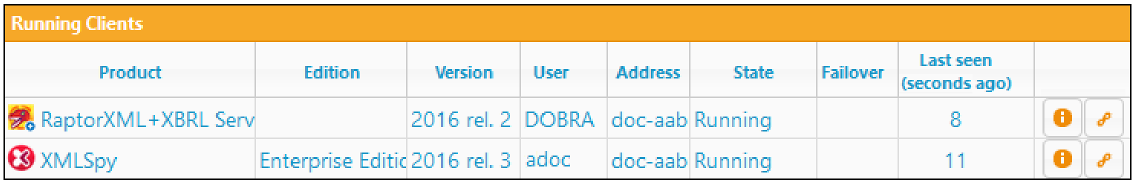
Client Monitoring

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 Editic	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.

Settings

This section:

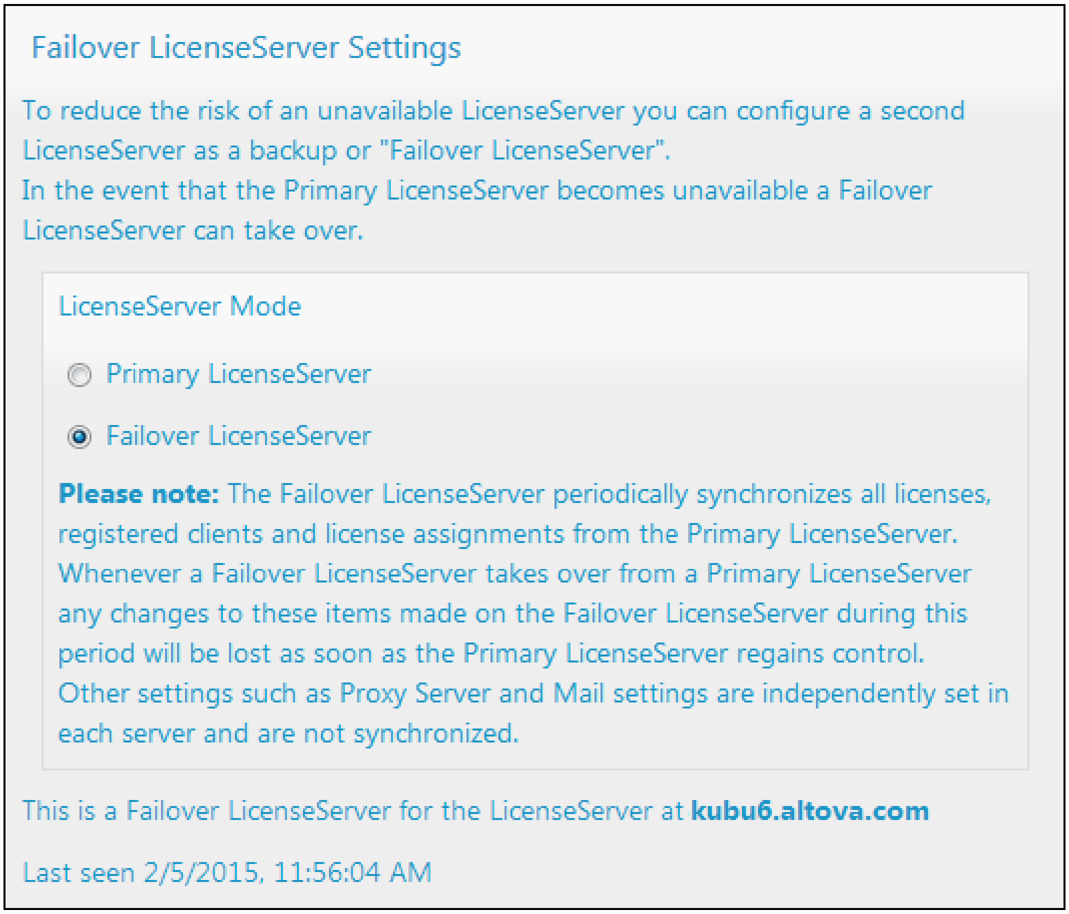
- Failover LicenseServer settings
- Network settings
- Alert Mail settings
- Miscellaneous settings

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

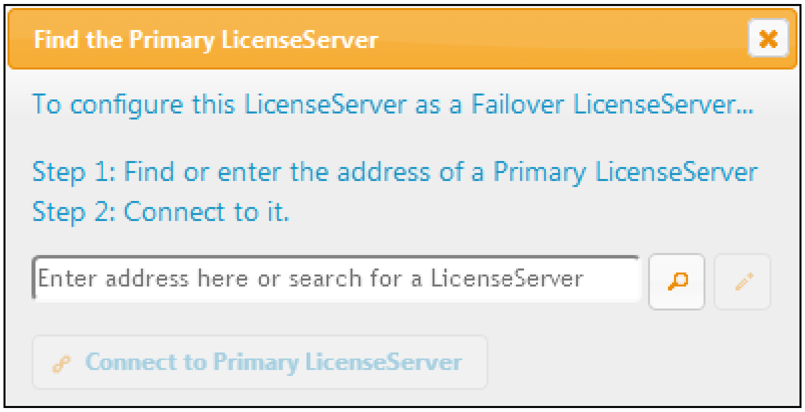
Please note: The Failover LicenseServer periodically synchronizes all licenses, registered clients and license assignments from the Primary LicenseServer. Whenever a Failover LicenseServer takes over from a Primary LicenseServer any changes to these items made on the Failover LicenseServer during this 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**.

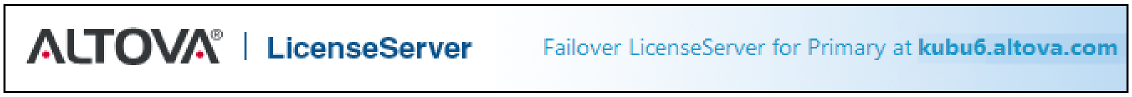

Find the Primary LicenseServer
Enter address here or search for a LicenseServer
Connect to Primary LicenseServer

To configure this LicenseServer as a Failover LicenseServer...

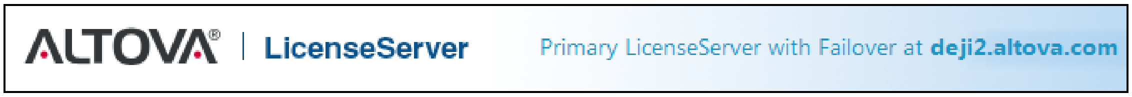
Step 1: Find or enter the address of a Primary LicenseServer
Step 2: Connect to it.

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.



Failover LicenseServer for Primary at **kubu6.altova.com**



Primary LicenseServer with Failover at **deji2.altova.com**

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).

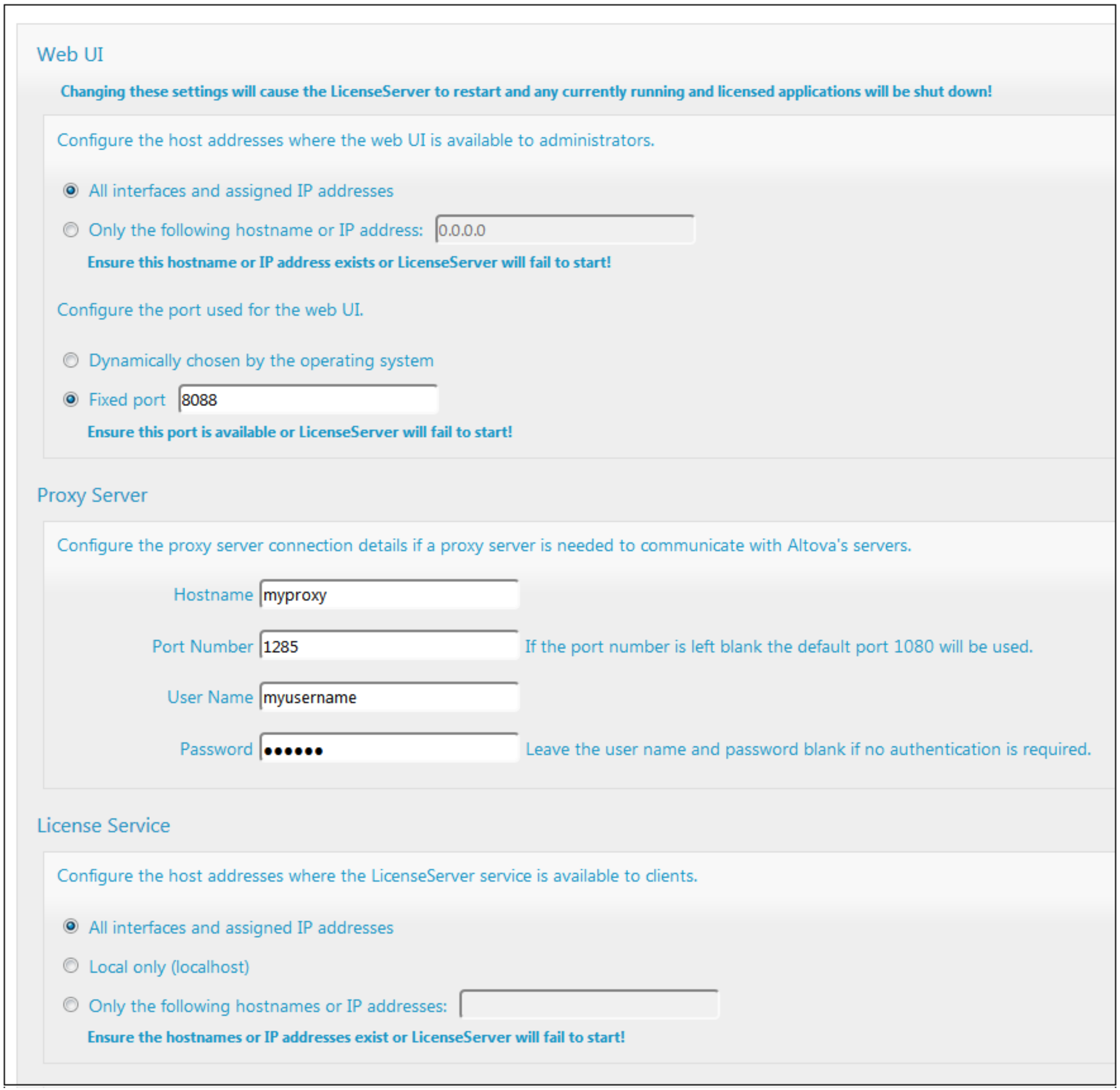
License checkout

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 teh network.

Network settings

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



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: 0.0.0.0

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 8088

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

Proxy Server

Configure the proxy server connection details if a proxy server is needed to communicate with Altova's servers.

Hostname myproxy

Port Number 1285 If the port number is left blank the default port 1080 will be used.

User Name myusername

Password •••••• Leave the user name and password blank if no authentication is required.

License Service

Configure the host addresses where the LicenseServer service is available to clients.

● All interfaces and assigned IP addresses

○ Local only (localhost)

○ Only the following hostnames or IP addresses:

Ensure the hostnames or IP addresses exist or LicenseServer will fail to start!

- *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**.

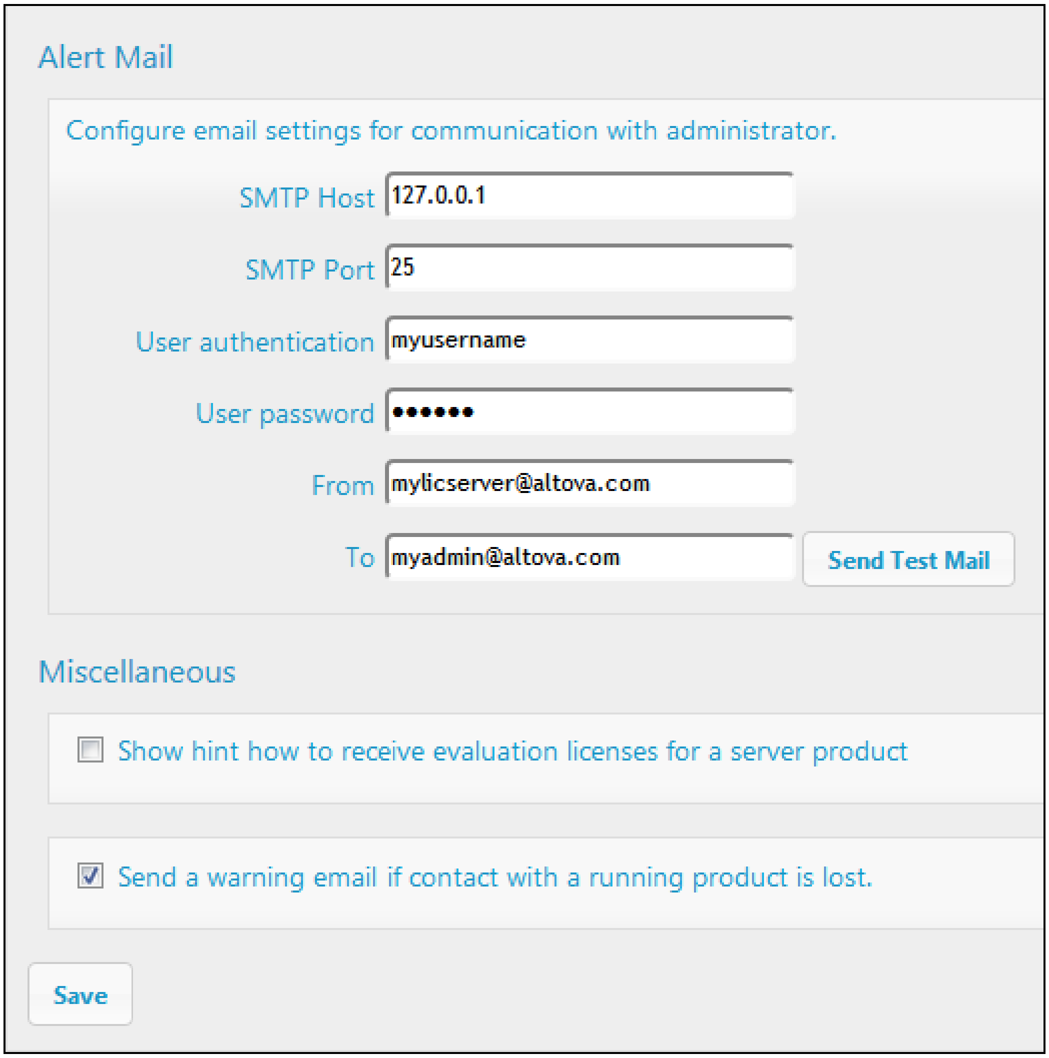
The default settings allow unrestricted access to LicenseServer and its configuration page from within the networks to which LicenseServer is connected. If you wish to restrict access to either LicenseServer or its configuration page, enter the appropriate settings and click **Save**.

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

Alert Mail settings

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: 127.0.0.1

SMTP Port: 25

User authentication: myusername

User password: ••••••

From: mylicserver@altova.com

To: myadmin@altova.com

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.

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.

7.8 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`.

Index