Altova MobileTogether Server Advanced Edition

ALTOVA®
MobileTogether®
SERVER

User & Reference Manual

Altova MobileTogether Server Advanced Edition User & Reference Manual

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

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

Published: 2024

Table of Contents

4 Server Procedures 42

5 Web UI Reference 76

1 Welcome to MobileTogether Server

MobileTogether Server Advanced Edition (hereafter referred to as MobileTogether Server for short) 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.

Current version: 9.1

This documentation

This documentation is organized into the following sections:

Also see: Demo videos 42 about MobileTogether Server.

Altova website: 🔗 App development, Enterprise apps, Enterprise app development, RMAD, Low code app development

Last updated: 22 January 2024

2 Introduction

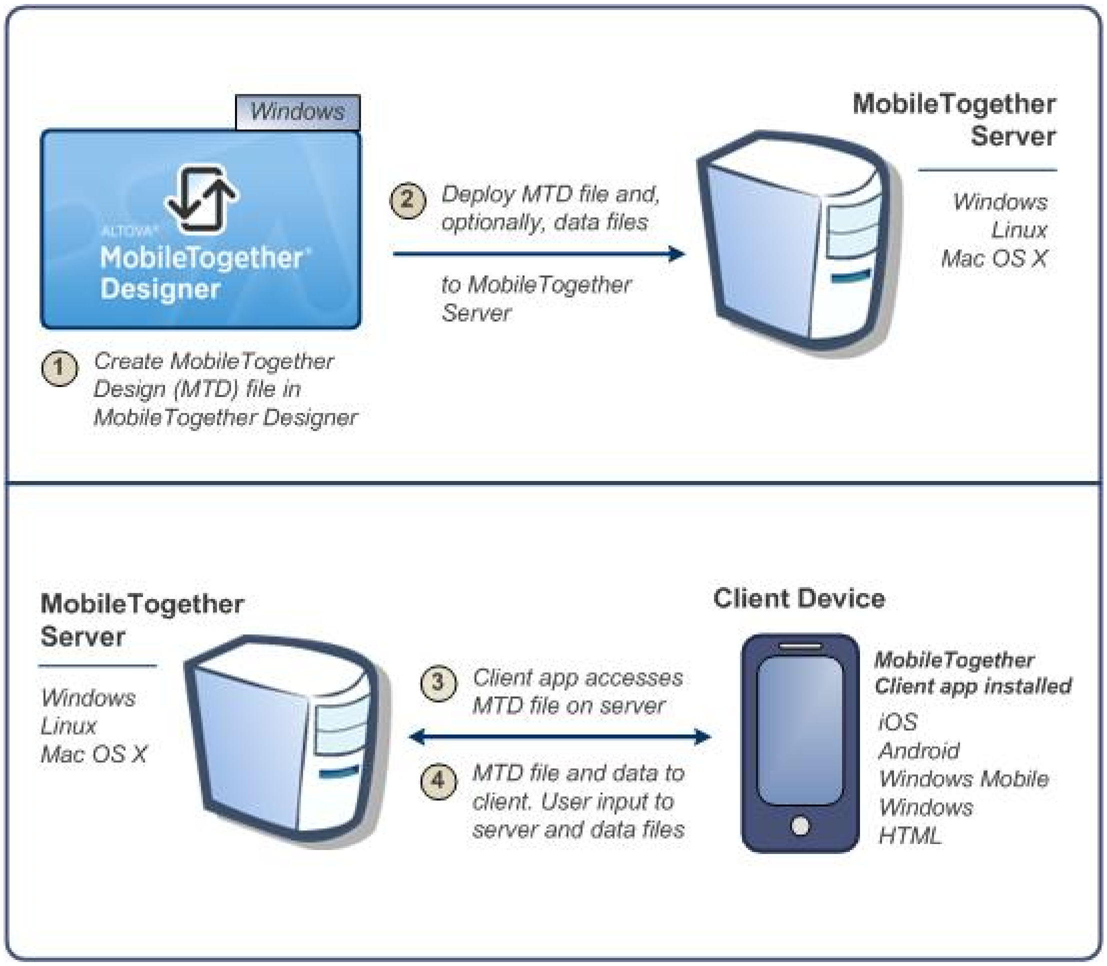
This introduction:

- MobileTogether Overview 9, which describes the MobileTogether system and the place of MobileTogether Server within that system
- Using MobileTogether Server 11 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* 42*, 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
Linux
Mac OS X
MobileTogether Server
Windows
Linux
Mac OS X
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 10, Windows 11
Windows Server	Windows Server 2012 or newer

MobileTogether Server

Windows	Windows 10, Windows 11
Windows Server	Windows Server 2012 or newer
Linux	• Red Hat Enterprise Linux 7 or newer • CentOS 7, CentOS Stream 8 • Debian 9 or newer • Ubuntu 18.04, 20.04, 22.04 • AlmaLinux 9.0 • Rocky Linux 9.0
macOS	macOS 11 or newer

MobileTogether Client

iOS	9 and higher for Apple mobile devices
Android	5.0 and higher for Android mobile devices
Windows RT, Metro	Windows 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 20, Licensing on Linux 26, and Licensing on macOS 34.

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* 47). 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 52, and other communication settings and security settings 110.

5. Set up user accounts

   MobileTogether Server is always accessed via a user account 56, 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 72. *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 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.

3 Set Up MobileTogether Server

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

3.1 Setup on Windows

This section describes the installation 14 and licensing 20 of MobileTogether Server on Windows systems.

System requirements (Windows)

- Windows 10, Windows 11
- Windows Server 2012 or newer

Prerequisites

- Perform installation as a user with administrative privileges.
- From version 2021 onwards, a 32-bit version of MobileTogether Server cannot be installed over a 64-bit version, or a 64-bit version over a 32-bit version. You must either (i) remove the older version before installing the newer version or (ii) upgrade to a newer version that is the same bit version as your older installation.

3.1.1 Install on Windows

MobileTogether Server is available for installation on Windows systems. The broad installation and setup procedure is described below. For detailed information about specific parts of the installation procedure, see their respective topics.

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. You can select your installation language from the box in the lower left area of the wizard. Note that this selection also sets the default language of MobileTogether Server. You can change the language later from the command line.

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

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

3.1.2 Install on Windows Server Core

Windows Server Core is a minimal Windows installation that does not use a number of GUI features. You can install MobileTogether Server on a Windows Server Core machine as follows:

1. Download the MobileTogether Server installer executable from the Altova website. This file is named **`MobileTogetherServerAdv.exe`**. Make sure to choose the executable matching your server platform (32-bit or 64-bit).
2. On a standard Windows machine (not the Windows Server Core machine), run the command **`MobileTogetherServerAdv_9.1.exe /u`**. This unpacks the **`.msi`** file to the same folder as the installer executable.
3. Copy the unpacked **`.msi`** file to the Windows Server Core machine.

4. If you are updating an earlier version of MobileTogether Server, shut down MobileTogether Server before carrying out the next step.
5. Use the **.msi** file for the installation by running the command **msiexec /i MobileTogetherServerAdvanced.msi**. This starts the installation on Windows Server Core.

Note: When upgrading to a major version, you can retain your MobileTogether Server settings by using the properties listed in the subsections of this section: (i) Webserver Properties 16, (ii) SSL-Webserver Properties 17, and (iii) Service Properties 18.

Important: Keep the MSI file!

Note the following points:

- Keep the extracted **.msi** file in a safe place. You will need it later to uninstall, repair, or modify your installation.
- If you want to rename the MSI file, do this before you install MobileTogether Server.
- The MSI filename is stored in the registry. You can update its name there if the filename has changed.

Register MobileTogether Server with LiceseServer

If you are installing MobileTogether Server for the first time or are upgrading to a **major version**, you will need to register MobileTogether Server with an Altova LicenseServer on your network. If you are upgrading to a non-major version of MobileTogether Server, then the previous LicenseServer registration will be known to the installation and there is no need to register MobileTogether Server with LicenseServer. However, if you want to change the LicenseServer that is used by MobileTogether Server at any time, then you will need to register MobileTogether Server with the new LicenseServer.

To register MobileTogether Server with an Altova LicenseServer during installation, run the installation command with the **REGISTER_WITH-LICENSE_SERVER** property, as listed below, providing the name or address of the LicenseServer machine as the value of the property, for example:

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

To register MobileTogether Server with an Altova LicenseServer after installation, run the following command:

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

Useful commands

Given below are a set of commands that are useful in the installation context.

To test the return value of the installation, run a script similar to that below. The return code will be in the `%errorlevel%` environment variable. A return code of **0** indicates success.

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

For a silent installation with a return code and a log of the installation process:

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

To modify the installation:

```
msiexec /m MobileTogetherServerAdvanced.msi
```

To repair the installation:

```
msiexec /r MobileTogetherServerAdvanced.msi
```

To uninstall MobileTogether Server:

```
msiexec /x MobileTogetherServerAdvanced.msi
```

To uninstall MobileTogether Server silently and report the detailed outcome in a log file:

```
start /wait msiexec /x MobileTogetherServerAdvanced.msi /q /L*v!
<pathToUninstallLogFile>
```

To install MobileTogether Server using another langauge (available language codes are: German=`de`; Spanish=`es`; French=`fr`):

```
msiexec /i MobileTogetherServerAdvanced.msi INSTALLER_LANGUAGE=<languageCode>
```

Note: On Windows Server Core, the charts functionality of MobileTogether Server will not be available.

3.1.2.1 Webserver Properties

You can configure the MobileTogether Server web server by using the properties given below. To set a property, run the installation command with the property setting appended, like this:

```
msiexec /i MobileTogetherServerAdvanced.msi MTSAdmin_WebServer_Host=127.0.0.1
```

List of properties

MobileTogether Server Administrator interface

Properties of the web server of the MobileTogether Server administrator interface:

`MTSAdmin_WebServer_Host=<IP4 Address>`

Use **`127.0.0.1`** if you want to access the web server from this machine only. Use **`0.0.0.0`** to make the web server accessible globally.

`MTSAdmin_WebServer_Port=<Port Number>`

Specifies the port that is used to access the web server.

`MTSAdmin_WebServer_Enabled=<0 or 1>`

Select **`1`** to enable listening at the currently set port. Select **`0`** to disable listening at this port.

MobileTogether Server Client interface

Properties of the web server of the MobileTogether Client administrator interface:

`MTSClient_WebServer_Host=<IP4 Address>`

Use **`127.0.0.1`** if you want to access the web server from this machine only. Use **`0.0.0.0`** to make the web server accessible globally.

`MTSClient_WebServer_Port=<Port Number>`

Specifies the port that is used to access the web server.

`MTSClient_WebServer_Enabled=<0 or 1>`

Select **`1`** to enable listening at the currently set port. Select **`0`** to disable listening at this port.

3.1.2.2 SSL-Webserver Properties

You can configure the MobileTogether Server SSL web server by using the properties given below. To set a property, run the installation command with the property setting appended, like this:

```
msiexec /i MobileTogetherServerAdvanced.msi MTSAdmin_SSLWebServer_Host=127.0.0.1
```

List of properties

MobileTogether Server Administrator interface

To configure the SSL web server of the MobileTogether Server administrator interface, use the following properties:

`MTSAdmin_SSLWebServer_Host=<IP4 Address>`

Use **`127.0.0.1`** if you want to access the SSL web server from this machine only. Use **`0.0.0.0`** to make the SSL web server accessible globally.

`MTSAdmin_SSLWebServer_Port=<Port Number>`

Specifies the port that is used to access the SSL web server.

`MTSAdmin_SSLWebServer_Enabled=<0 or 1>`

Select **`1`** to enable listening at the currently set port. Select **`0`** to disable listening at this port.

MobileTogether Client Administrator interface

To configure the SSL web server of MobileTogether Server's client interface, use the following properties:

`MTSClient_SSLWebServer_Host=<IP4 Address>`

Use **`127.0.0.1`** if you want to access the SSL web server from this machine only. Use **`0.0.0.0`** to make the SSL web server accessible globally.

`MTSClient_SSLWebServer_Port=<Port Number>`

Specifies the port that is used to access the SSL web server.

`MTSClient_SSLWebServer_Enabled=<0 or 1>`

Select **`1`** to enable listening at the currently set port. Select **`0`** to disable listening at this port.

MobileTogether Server SSL certificates

Properties to specify an SSL certificate for the MobileTogether SSL web server:

`MTS_SSLWebServer_Certificate=<Path-to-certificate-file>`

Full path to a SSL certificate, enclosed in double-quotes.

`MTS_SSLWebServer_PrivateKey=<Path-to-private-key-file>`

Full path to a private key file, enclosed in double-quotes.

3.1.2.3 Service Properties

You can configure the MobileTogether Server service by using the properties given below. To set a property, run the installation command with the property setting appended, like this:

```
msiexec /i MobileTogetherServerAdvanced.msi MTS_Service_DisplayName=MobileTogetherServer
```

List of properties

To configure MobileTogether Server services, use the following properties:

`MTS_Service_DisplayName=<Serveice Display Name>`
Name that will be displayed for the service. Enclose the name in double quotes.

`MTS_Service_StartType=<Startup Type>`
Specifies how the service is started during a system start-up. Values can be one of: **`auto | auto-delayed | demand | disabled`**.

`MTS_Service_Username=<UserName>`
Specifies the log-on user for the service. Use one of: **`LocalSystem | NT Authority\LocalService | NT Authority\NetworkService | <any user with relevant rights>`**.

`MTS_Service_Password=<Password>`
The password of the service's start user in plain text.(Hint: Use the installer's user interface to avoid entering plain text passwords.) No password is required if the user name is any of: **`LocalSystem | NT Authority\LocalService | NT Authority\NetworkService`**.

3.1.3 Install LicenseServer (Windows)

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, you can install LicenseServer together with MobileTogether Server. If a LicenseServer is already installed on your network, you do not need to install another one—unless a newer version of LicenseServer is required. (*See next point, LicenseServer versions*.)

During the installation process of MobileTogether Server, check or uncheck the option for installing LicenseServer as appropriate. Note the following points:

- If you have not installed LicenseServer yet, leave the default settings as is. The wizard will install the latest version on the computer where you are running the wizard.
- If you have not installed LicenseServer yet and want to install Altova LicenseServer on another computer, clear the check box *Install Altova LicenseServer on this machine* and choose **Register**

Later. In this case, you will need to install LicenseServer separately and register MobileTogether Server afterwards.

- If LicenseServer has already been installed on your computer but is a lower version than the one indicated by the installation wizard, leave the default setting (for upgrading to the newer version) as is. In this case, the installation wizard will automatically upgrade your LicenseServer version. The existing registration and licensing information will be carried over to the new version of LicenseServer.
- If LicenseServer has already been installed on your computer or network and has the same version as the one indicated by the wizard, do the following:
  - o Clear the check box *Install Altova LicenseServer on this machine*.
  - o Under *Register this product with*, choose the LicenseServer with which you want to register MobileTogether Server. Alternatively, choose **Register Later**. Note that you can always select **Register Later** if you want to ignore the LicenseServer associations and carry on with the installation of MobileTogether Server.

For information about how to register and license MobileTogether Server with Altova LicenseServer, see the section Licensing MobileTogether Server[20].

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. Alternatively, 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 of LicenseServer 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: 3.12

3.1.4 Network and Service Configuration (Windows)

During the installation of MobileTogether Server, you can configure settings for accessing MobileTogether Server via the network and for running MobileTogether Server as a Windows service (*screenshot below*). You can configure settings for the administrator interface and client interface separately by selecting their respective tabs.
The settings listed below are available. Leave the default settings as they are if they are acceptable to you or if you are not sure about them. If you wish to change a setting, select its **Change** button (*see screenshot above*).

- The port to use for unencrypted communication with MobileTogether Server.

- Whether secure (SSL-encrypted) connections to MobileTogether Server are allowed. If yes, then on which port. By default, secure connections are disabled. For more information, see the section about setting up SSL encryption 47.
- Windows service settings. These include:
  - o The way MobileTogether Server should start as a Windows service: automatic, on demand, delayed automatic, or disabled.
  - o The user account to be used by MobileTogether Server for the Windows service: *Local System, Local Service, Network Service,* or *Other User*. If you select *Other User*, you can set the username and password of this user, similar to how this is done in the Windows Services management console. Note that the selected user must have read/write access to **`C:\ProgramData\Altova`**. Otherwise, the installation or startup could fail.

You can change the settings after installation. To modify the Windows service configuration, open the Windows Services management console (by typing **`Services.msc`** in a command line window) and change the required service from there.

3.1.5 License MobileTogether Server (Windows)

In order to use MobileTogether Server, it must be licensed with Altova LicenseServer. 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 from LicenseServer. Download the latest version of LicenseServer from the Altova website, and install it on your local machine or a machine on your network.

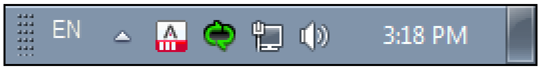
These steps are described in this section. For detailed information, see the LicenseServer user manual at the Altova website.

3.1.5.1 Start LicenseServer, MobileTogether Server

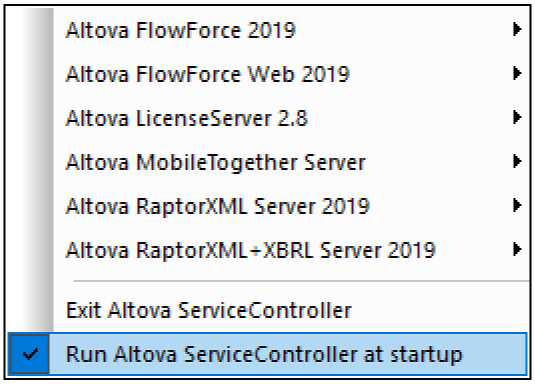
Altova LicenseServer (LicenseServer for short) and MobileTogether Server are both started via Altova ServiceController.

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 (DiffDog Server, FlowForce Server, Mobile Together Server, and RaptorXML(+XBRL) Server). ServiceController can be accessed via the system tray (*screenshot below*).

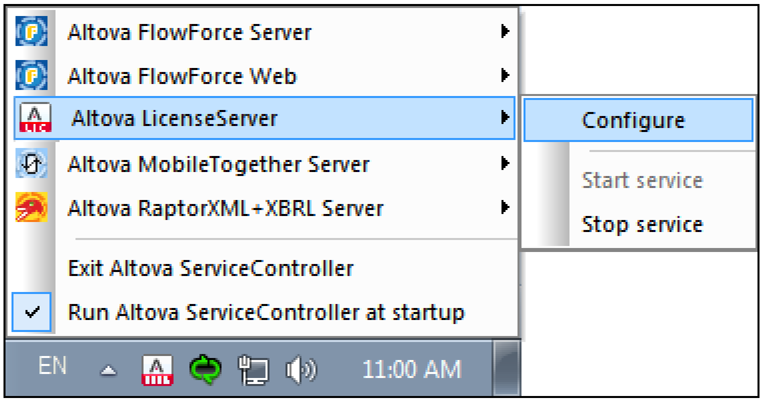


To specify that ServiceController starts automatically on logging in to the system, click the **ServiceController** icon in the system tray to display the **ServiceController** menu (*screenshot below*), and then toggle on the command **Run Altova ServiceController at Startup**. (This command is toggled 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 2019
Altova FlowForce Web 2019
Altova LicenseServer 2.8
Altova MobileTogether Server
Altova RaptorXML Server 2019
Altova RaptorXML+XBRL Server 2019
Exit Altova ServiceController
✓ Run Altova ServiceController at startup

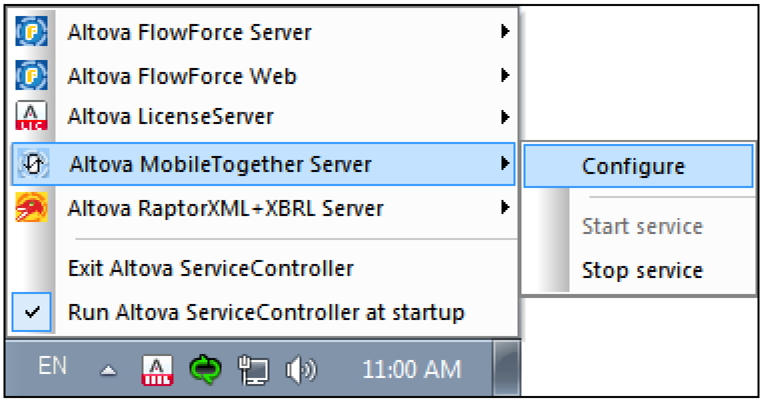
Start LicenseServer

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, then the *Start Service* option will be disabled. You can also stop the service via 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 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. You can also stop the service via 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

3.1.5.2 Register MobileTogether Server

In order to be able to license MobileTogether Server from Altova LicenseServer, MobileTogether Server must be registered with LicenseServer

To register MobileTogether Server from the command line interface, use the `licenseserver` command and supply the address of the LicenseServer machine.

```
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 110. 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 Client Management tab of LicenseServer's configuration page to assign a license to MobileTogether Server.

Note: For more information about registering Altova products with LicenseServer, see the LicenseServer user manual.

3.1.5.3 License MobileTogether Server

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

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, would 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 at a time, even if the license has unused licensing capacity. (A client machine is the machine on which the Altova server product is installed.) 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

Because of its services functionality, MobileTogether Server Advanced Edition will run only on machines with **two or more cores**.

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

Single-thread execution

If an Altova server product allows single-thread execution, an option for *Single-thread execution* will be available. In these cases, 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 in LicenseServer, select the *Limit to single thread execution* check box for that product.

Estimate of core requirements

There are various external factors that influence the data volumes and processing times your server can handle (for example: the hardware, the current load on the CPU, and memory allocation of other applications running on the server). In order to measure performance as accurately as possible, test the applications in your environment with data volumes and in conditions that approximate as closely as possible to real business situations.

The following information can be used as an estimate of core requirements: The Altova MyCollections app is being served to Android. iOS, and Windows platforms by a MobileTogether Server installation that runs on a 4-core machine. At the time of writing (March 2019), the MyCollections app has been downloaded to 50,000+ Android client devices according to Google Play statistics (iOS and Windows data was not published by the corresponding stores). By considering these statistics and evaluating the amount of functionality in the MyCollections app, you will be able to get an estimate of the processing power of cores in relation to MobileTogether Server functionality.

3.2 Setup on Linux

This section describes the installation[24] and licensing[26] of MobileTogether Server on Linux systems (Debian, Ubuntu, CentOS, RedHat).

System Requirements (Linux)

- Red Hat Enterprise Linux 7 or newer
- CentOS 7, CentOS Stream 8
- Debian 9 or newer
- Ubuntu 18.04, 20.04, 22.04
- AlmaLinux 9.0
- Rocky Linux 9.0

Prerequisites

- Perform installation either as **root** user or as a user with **sudo** privileges.
- The previous version of MobileTogether Server must be uninstalled before a new one is installed.
- If you plan to use Altova's Charts functionality, then at least one font must be installed on your system to ensure that charts will be rendered correctly. To list installed fonts, use, for example, the **`fc-list`** command of the Fontconfig library.
- 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
MobileTogether Server Advanced Edition	libidn or libidn2, krb5-libs	libidn2-0, libgssapi-krb5-2	libidn2-0, libgssapi-krb5-2, libldap-2.4 *(also see note)*

3.2.1 Install on Linux

MobileTogether Server is available for installation on Linux systems. Its installation and setup procedure is described below. Perform installation either as **root** user or as a user with **sudo** privileges.

Uninstall MobileTogether Server

If you need to uninstall a previous version of MobileTogether Server, 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 *Install MobileTogether Server*.

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

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

On Debian and Ubuntu systems, it might happen that MobileTogether Server still appears in the list of installed products after it has been uninstalled. In this case, run the **purge** command to clear MobileTogether Server from the list. You can also use the **purge** command *instead* of the **remove** command listed above.

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

Download the MobileTogether Server Linux package

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

Distribution	Package extension
Debian	.deb
Ubuntu	.deb
CentOS	.rpm
RedHat	.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-9.1-debian.deb
[Ubuntu]:   sudo dpkg --install mobiletogetherserver-9.1-ubuntu.deb
[CentOS]:   sudo rpm -ivh mobiletogetherserver-9.1-1.x86_64.rpm
[RedHat]:   sudo rpm -ivh mobiletogetherserver-9.1-1.x86_64.rpm
```

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

The MobileTogether Server package will be installed in the folder:

```
/opt/Altova/MobileTogetherServer
```

3.2.2 Install LicenseServer (Linux)

In order for MobileTogether Server to work, it 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 topic*[24]).

```
[Debian]:   sudo dpkg --install licenseserver-3.12-debian.deb
[Ubuntu]:   sudo dpkg --install licenseserver-3.12-ubuntu.deb
[CentOS]:   sudo rpm -ivh licenseserver-3.12-1.x86_64.rpm
[RedHat]:   sudo rpm -ivh licenseserver-3.12-1.x86_64.rpm
```

The LicenseServer package will be installed in:

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

For information about how to register and license MobileTogether Server with Altova LicenseServer, see the section Licensing MobileTogether Server[26]. Also see the LicenseServer documentation for more detailed information.

LicenseServer versions

- Altova server products must be licensed either with the version of LicenseServer that is appropriate to the installed 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. Alternatively, 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 of LicenseServer 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: 3.12

3.2.3 License MobileTogether Server (Linux)

In order to use MobileTogether Server, it must be licensed with Altova LicenseServer. 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 from LicenseServer. Download the latest version of

LicenseServer from the Altova website, and install it on your local machine or a machine on your network.

These steps are described in this section. For detailed information, see the LicenseServer user manual at the Altova website.

3.2.3.1 Start LicenseServer, MobileTogether Server

This topic describes how to start Altova LicenseServer (LicenseServer for short) and MobileTogether Server. To be able to start these programs, you can use one of the following options: (i) you can be the root user and leave out the `sudo` keyword from the commands listed below (leaving out `sudo` is optional), or (ii) you can run the `sudo` command as a normal user with the corresponding permissions for `sudo`.

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], [≥ CentOS 7], [≥ Ubuntu 15]	**sudo systemctl start licenseserver**

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

```
sudo systemctl stop licenseserver
```

Start MobileTogether Server

Start MobileTogether Server as a daemon with the command appropriate for your system from those listed below.

[≥ Debian 8], [≥ CentOS 7], [≥ Ubuntu 15]	**sudo systemctl start mobiletogetherserver**

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

```
sudo systemctl stop mobiletogetherserver
```

Check status of daemons

To check if a daemon is running, run the following command, replacing **<servicename>** with the name of the daemon you want to check:

```
sudo service <servicename> status
```

3.2.3.2 Register MobileTogether Server

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

```
sudo /opt/Altova/MobileTogetherServer/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/MobileTogetherServer/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/MobileTogetherServer/bin/
```

You can also register MobileTogether Server from the Settings tab of MobileTogether Server's Web UI 110. Essentially: (i) Start MobileTogether Server; (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 Client Management tab of LicenseServer's configuration page to assign a license to MobileTogether Server.

Note: For more information about registering Altova products with LicenseServer, see the LicenseServer user manual.

3.2.3.3 License MobileTogether Server

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

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, would 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 at a time, even if the license has unused licensing capacity. (A client machine is the machine on which the Altova server product is installed.) 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
Because of its services functionality, MobileTogether Server Advanced Edition will run only on machines with **two or more cores**.

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

Single-thread execution
If an Altova server product allows single-thread execution, an option for *Single-thread execution* will be available. In these cases, 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 in LicenseServer, select the *Limit to single thread execution* check box for that product.

Estimate of core requirements
There are various external factors that influence the data volumes and processing times your server can handle (for example: the hardware, the current load on the CPU, and memory allocation of other applications running on the server). In order to measure performance as accurately as possible, test the applications in your environment with data volumes and in conditions that approximate as closely as possible to real business situations.

The following information can be used as an estimate of core requirements: The Altova MyCollections app is being served to Android. iOS, and Windows platforms by a MobileTogether Server installation that runs on a 4-core machine. At the time of writing (March 2019), the MyCollections app has been downloaded to 50,000+ Android client devices according to Google Play statistics (iOS and Windows data was not published by the corresponding stores). By considering these statistics and evaluating the amount of functionality in the MyCollections app, you will be able to get an estimate of the processing power of cores in relation to MobileTogether Server functionality.

3.2.4 Notes about Environment (Linux)

Folders

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

⊟ *Installation root*

```
/opt/Altova/MobileTogetherServer/
```

⊟ *License Files*

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

⊟ *Environment settings*

`/etc/profile.d/jdbc.sh`
The environment settings file (typically named **`jdbc.sh`**) is executed at system start. The definitions in it must be specific to your particular 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, so you must be careful when modifying settings. For example, if you modify a class path in this file, then the modifications will be applied across the system. If you wish to make changes for MobileTogether Server only, you might want to consider using a unit file (explained in the section *JDBC Connections* below).

Filesystem triggers and permissions
In order for filesystem triggers to be fired, the user that started the MobileTogether Server service (**`altovamobiletogetherserver`**) must have the following permissions:

- For the triggered folder: *Read* and *Execute*
- For the triggered file: *Read*
- For ancestor folders of the triggered folder: *Read* and *Execute*

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 Solution's Working Directory 110. By default, this folder is:

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

Adding class paths to the MobileTogether service file on CentOS

If your MobileTogether Server is installed on CentOS, you will need to add the class path to the **`mobiletogether.service`** file (which should be located at `/usr/lib/systemd/system/mobiletogether.service`).

Add the class path as follows:

1. In the **`mobiletogether.service`** file, find the `[Service]` section, which begins with "`PIDFile=/var ...`"
2. Above the `[Service]` section add the line: **`Environment="CLASSPATH=<`*`classpaths-go-here`*`>"`**

Database connections

On Linux, the following database connections are supported:

- JDBC — You can use JDBC for all supported databases except Microsoft Access
- Native connections — Currently available for SQLite and PostgreSQL databases

If you are using JDBC, note the following points:

- The Java Runtime Environment or SDK must be installed.
- The JDBC drivers for the target database must be installed.
- The following environment variables must be set correctly for your environment:

- CLASSPATH: to find the jar-files that connect to the JDBC database; the jar-files can be entered either in (i) an executable script (like jdbc.sh) that is executed on system start or (ii) a unit file that is executed when MobileTogether Server is started as a service. Using a unit file to specify the jar-files has the advantage that the files required for MobileTogether Server's JDBC connections will be located without you having to modify the existing system configuration. A unit file is listed below.
- PATH: to find the JRE, but might not be necessary depending on the installation
- JAVA_HOME: if necessary, depending on the installation.

Listing of important files

The following shell script (or unit file) is copied to the folder **/opt/Altova/MobileTogetherServer/etc** so as not to overwrite already existing configuration files. Make the necessary changes as required. Also see the section *JDBC Connections* above. The parts highlighted in blue are environment-specific and need to be adjusted to match your environment:

⊟ Shell script (unit file)

```
#- 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/j
dbc/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/jc
onn7/jconn4.jar:/usr/local/jdbc/db2/db2jcc.jar:/usr/local/jdbc/db2/db2jcc_license_cu.ja
r:./:
```

3.3 Setup on macOS

This section describes the installation 32 and licensing 34 of MobileTogether Server on macOS systems.

System Requirements (macOS)

- macOS 11 or newer
- Java for macOS (latest version)

Prerequisites

- Ensure that Altova LicenseServer has been installed and is running
- Perform installation either as **root** user or as a user with **sudo** privileges.
- The previous version of MobileTogether Server must be uninstalled before a new one is installed.
- If you plan to use Altova's Charts functionality, then at least one font must be installed on your system to ensure that charts will be rendered correctly. To list installed fonts, use, for example, the **`fc-list`** command of the Fontconfig library.
- The macOS machine must be configured so that its name resolves to an IP address. This means that you must be able to successfully ping the host name from the Terminal using the command **`ping <hostname>`**.
- 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 2017-001) is available at: http://support.apple.com/kb/DL1572. For the **latest Java for OS X version provided by Apple**, do a search for Java for OS X at the Apple website. (Note that the Java version for OS X provided by Apple might not be the latest Java version provided by Oracle, but this is the version that you need to install.)

3.3.1 Install on macOS

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

Uninstall MobileTogether Server

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

```
    sudo launchctl unload /Library/LaunchDaemons/com.altova.MobileTogetherServer.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 following command:

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

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.

Install MobileTogether Server

1. Download the disk image (**`.dmg`**) file of MobileTogether Server from the Altova website (http://www.altova.com/download.html).
2. Click to open the downloaded disk image (**`.dmg`**). This causes the MobileTogether Server installer to appear as a new virtual drive on your computer.
3. On the new virtual drive, double-click the installer package (**`.pkg`**).
4. 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. *See also Licensing MobileTogether Server* 34.
5. 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/MobileTogetherServer` (application binaries)
`/var/Altova/MobileTogetherServer` (data files: database and logs)

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

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

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

See the topic Post-Licensing Tasks 40 for additional details.

3.3.2 Install LicenseServer (macOS)

In order for MobileTogether Server to work, it must be licensed via an Altova LicenseServer on your network. The LicenseServer installation package is available on the virtual drive you have mounted in the previous step. To install 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 Licensing on macOS 34.

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. Alternatively, 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 of LicenseServer 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: 3.12

3.3.3 License MobileTogether Server (macOS)

In order to use MobileTogether Server, it must be licensed with Altova LicenseServer. 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 from LicenseServer. Download the latest version of LicenseServer from the Altova website, and install it on your local machine or a machine on your network.

These steps are described in this section. For detailed information, see the LicenseServer user manual at the Altova website.

3.3.3.1 Start LicenseServer, MobileTogether Server

This topic describes how to start Altova LicenseServer (LicenseServer for short) and MobileTogether Server. You must have administrator (root) privileges to be able to start these programs, so you should run these commands as the root user. If you are logged in as `root`, you can leave out the `sudo` keyword from the commands listed 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.MobileTogetherServer.plist
```

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

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

3.3.3.2 Register MobileTogether Server

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

```
      sudo /usr/local/Altova/MobileTogetherServer/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/MobileTogetherServer/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/MobileTogetherServer/bin/
```

You can also register MobileTogether Server from the Settings tab of MobileTogether Server's Web UI. Essentially: (i) Start MobileTogether Server; (ii) Enter your password to access the **Settings | LicenseServer** page; (iii) Select the LicenseServer name or address, and click **Register with LicenseServer**.

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

Note: For more information about registering Altova products with LicenseServer, see the LicenseServer user manual.

3.3.3.3 License MobileTogether Server

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

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, would 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 at a time, even if the license has unused licensing capacity. (A client machine is the machine on which the Altova server product is installed.) 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

Because of its services functionality, MobileTogether Server Advanced Edition will run only on machines with **two or more cores**.

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

Single-thread execution

If an Altova server product allows single-thread execution, an option for *Single-thread execution* will be available. In these cases, 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 in LicenseServer, select the *Limit to single thread execution* check box for that product.

Estimate of core requirements

There are various external factors that influence the data volumes and processing times your server can handle (for example: the hardware, the current load on the CPU, and memory allocation of other applications running on the server). In order to measure performance as accurately as possible, test the applications in your environment with data volumes and in conditions that approximate as closely as possible to real business situations.

The following information can be used as an estimate of core requirements: The Altova MyCollections app is being served to Android. iOS, and Windows platforms by a MobileTogether Server installation that runs on a 4-core machine. At the time of writing (March 2019), the MyCollections app has been downloaded to 50,000+ Android client devices according to Google Play statistics (iOS and Windows data was not published by the corresponding stores). By considering these statistics and evaluating the amount of functionality in the MyCollections app, you will be able to get an estimate of the processing power of cores in relation to MobileTogether Server functionality.

3.3.4 Notes about Environment (macOS)

Folders

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

⊟ *Installation root*

`/usr/local/Altova/MobileTogetherServer/`

⊟ *License 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.

Filesystem triggers and permissions
In order for filesystem triggers to be fired, the user that started the MobileTogether Server service (**`altovamobiletogetherserver`**) must have the following permissions:

- For the triggered folder: *Read* and *Execute*
- For the triggered file: *Read*
- For ancestor folders of the triggered folder: *Read* and *Execute*

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 Solution's Working Directory 110. By default, this folder is:

`/var/Altova/MobileTogetherServer/SolutionFiles`

Database connections

On MacOS, the following database connections are supported:

- JDBC — You can use JDBC for all supported databases except Microsoft Access
- Native connections — Currently available for SQLite and PostgreSQL databases

If you are using JDBC, note the following points:

- The Java Runtime Environment or SDK must be installed.
- The JDBC-Connects for the target database must be installed.
- The following environment variables must be set correctly for your environment:
  - **`CLASSPATH:`** to find the jar-files; the class path is set in the `Plist` file.

- **`PATH:`** to find the JRE, but might not be necessary depending on the installation
- **`JAVA_HOME:`** if necessary, depending on the installation

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/MobileTogetherServer9.1/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/oracl
e/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/jc
onn7/jconn4.jar:/usr/local/jdbc/db2/db2jcc.jar:/usr/local/jdbc/db2/db2jcc_license_cu.ja
r:./</string>
      </dict>
   </dict>
</plist>
```

3.4 Upgrade MobileTogether Server

When you upgrade to a newer version of MobileTogether Server, the license of your previous version will be used automatically for the newer version if, during installation:

- the new version is registered with the same LicenseServer as that with which the previous version of MobileTogether Server was registered
- you accept the license agreement of MobileTogether Server.

The simplest way to carry over a license from the previous version of MobileTogether Server to the newer version is to let the installation process implement the required steps. The relevant steps during the installation process are listed below in the order in which they occur:

1. Let the installer register the new version of MobileTogether Server with the LicenseServer that holds the license used by the older version of MobileTogether Server.
2. Accept the license agreement of MobileTogether Server. (If you do not accept the agreement, the new version will not be installed.)

Note: If you do not register MobileTogether Server with the correct LicenseServer during the installation process, you will need to register and license MobileTogether Server manually with your alternative LicenseServer.

3.5 Post-Licensing Tasks

After you have completed the installation and licensing of MobileTogether Server, carry out the following tasks to finish the setup.

Open the Web UI (Setup) page of MobileTogether Server 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.

3.6 Migrate MobileTogether Server to a New Machine

If you want to migrate MobileTogether Server from one machine to another (including across supported platforms), follow the guidelines that are linked to from below.

Use the Backup and Restore 73 functionality, as described in the MobileTogether Server documentation.

4 Server Procedures

This section describes important server procedures. It assumes that MobileTogether Server has already been licensed 20. 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 43
- Start MobileTogether Server 45
- Set Up SSL Encryption 47
- Set Administrator and Mobile Client Ports 52
- Users and Roles 56
- Available Privileges 59
- Configure the Firewall 62
- Configure Services 63
- Solution Usage Statistics 68
- Information for Clients 72
- How to Back Up and Restore MobileTogether Server 73

Video demos

The links below take you to videos and blogposts on the Altova website that show how to configure MobileTogether Server.

- *Install and Configure MobileTogether Server*: Shows how to install MobileTogether Server and Altova LicenseServer, and how to configure MobileTogether Server behind a corporate firewall
- *Configuring MobileTogether Server in a Network*: Also explains how to set up ports so that MobileTogether Sever can be connected to from both outside and inside the network
- *An Altova blogpost* about configuring MobileTogether Server in a network

4.1 Start Altova LicenseServer

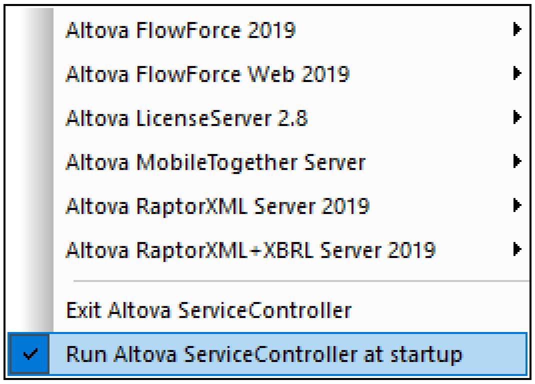
In order to run an installation of an Altova server product (i) FlowForce Server; (ii) RaptorXML(+XBRL) Server; (iii) MobileTogether Server; (iv) MapForce Server; (v) StyleVision Server, that installation must be licensed with an Altova LicenseServer on your network. LicenseServer must be running continuously 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.

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 2019
Altova FlowForce Web 2019
Altova LicenseServer 2.8
Altova MobileTogether Server
Altova RaptorXML Server 2019
Altova RaptorXML+XBRL Server 2019
Exit Altova ServiceController
✓ Run Altova ServiceController at startup

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.

```
sudo systemctl start licenseserver
```

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

▼ 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
```

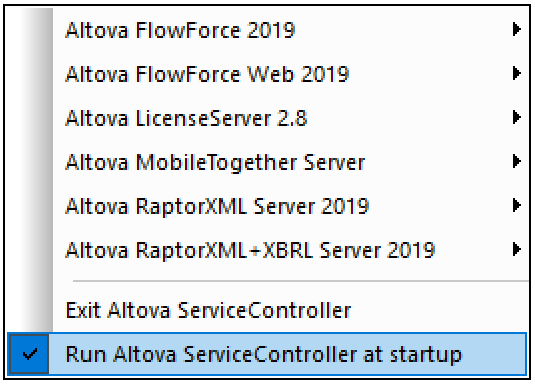
4.2 Start MobileTogether Server

In order to run MobileTogether Server, MobileTogether Server must be started as a service. 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 2019
Altova FlowForce Web 2019
Altova LicenseServer 2.8
Altova MobileTogether Server
Altova RaptorXML Server 2019
Altova RaptorXML+XBRL Server 2019
Exit Altova ServiceController
✓ Run Altova ServiceController at startup

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.

```
sudo systemctl start mobiletogetherserver
```

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

▼ 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
```

Note: If no license is assigned to MobileTogether Server, then MobileTogether Server will shut down automatically 24 hours after it has been started. After such a shutdown, you will need to restart MobileTogether Server as described above. After MobileTogether Server has been licensed, there is no automatic shutdown after 24 hours.

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*).

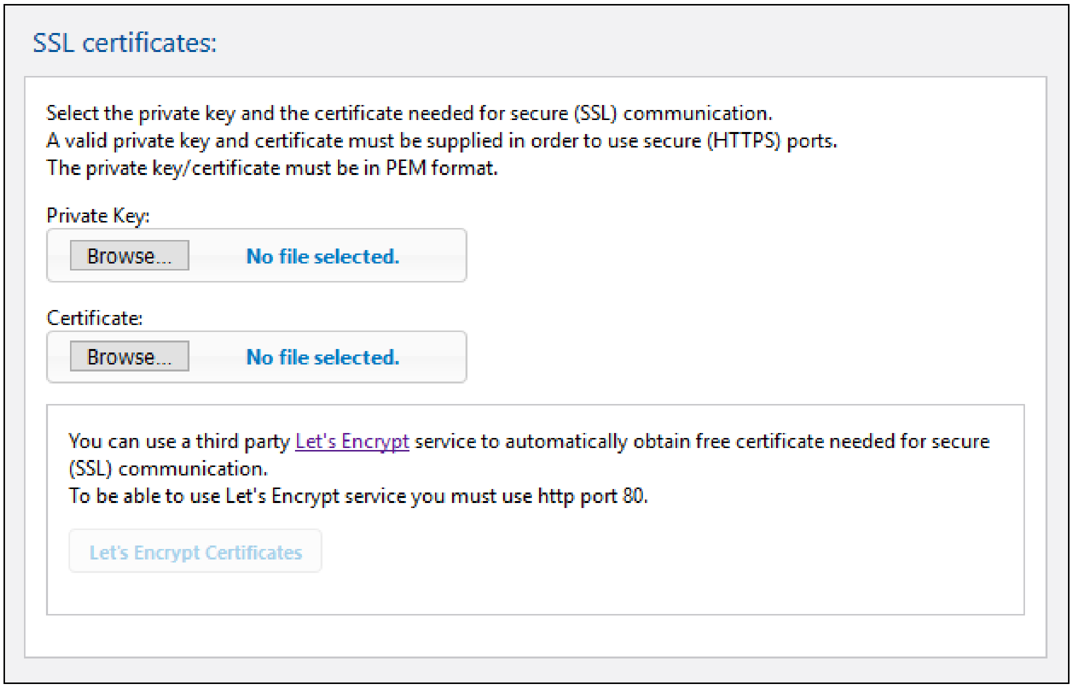
8. Install SSL certificate on MobileTogether Server

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:

Certificate:

You can use a third party Let's Encrypt service to automatically obtain free certificate needed for secure (SSL) communication.
To be able to use Let's Encrypt service you must use http port 80.

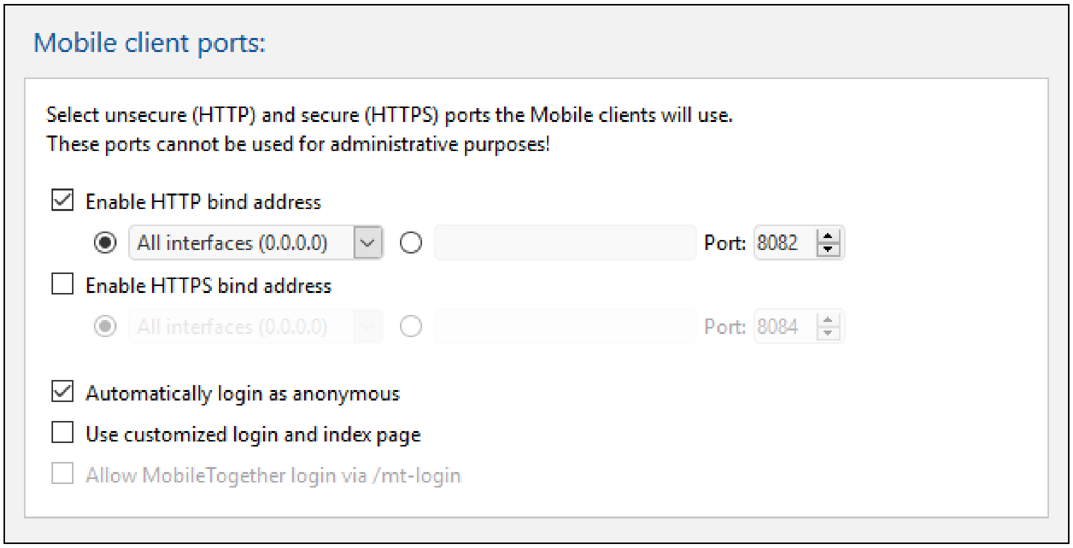
  - For the private key, select **`private.key`** (*created in Step 1*)
  - For the certificate, select **`publickey.cer`** (*created in Step 7*)

- 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!

☑ Enable HTTP bind address

● All interfaces (0.0.0.0) ○ Port: 8082

□ Enable HTTPS bind address

● All interfaces (0.0.0.0) ○ 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. 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.

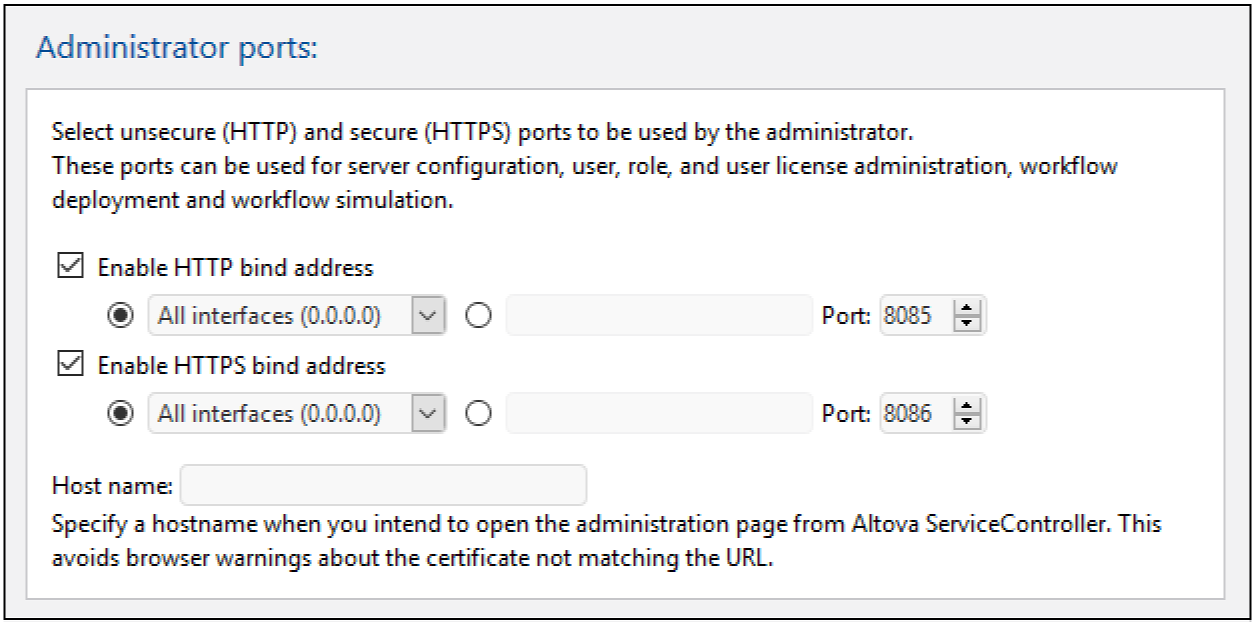
4.4 Set Administrator and Mobile Client Ports

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 87.
- 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.

☑ Enable HTTP bind address

● All interfaces (0.0.0.0) ○ Port: 8085

☑ Enable HTTPS bind address

● All interfaces (0.0.0.0) ○ Port: 8086

Host name:

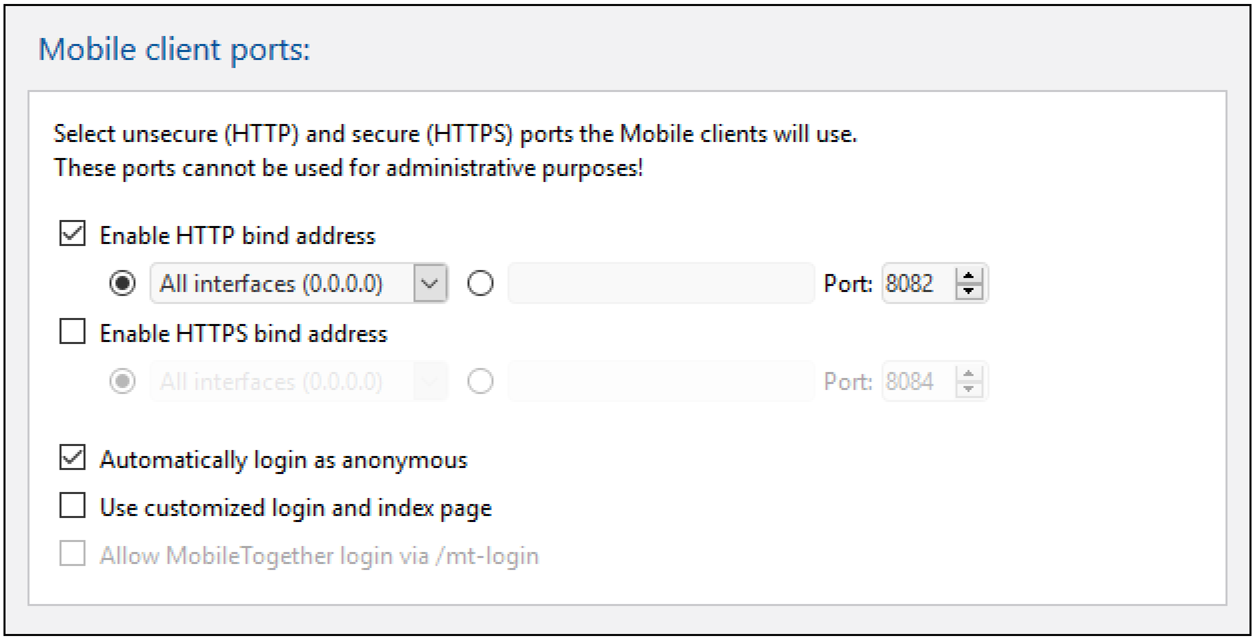
Specify a hostname when you intend to open the administration page from Altova ServiceController. This avoids browser warnings about the certificate not matching the URL.

The `HTTP` port is the unsecure port; `HTTPS` is the secure port. To use `HTTPS`, you need to set up SSL Encryption 47. 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.

You can specify 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 field of the second radio button. If you are using a dual-stack server running both IPv4 and IPv6, use a double colon `::` as the bind address; this allows both protocols on all network interfaces. Only ports with numbers from **`1`** to **`65535`** may be used.

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 47. You can specify 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 field of the second radio button. If you are using a dual-stack server running both IPv4 and IPv6, use a double colon `::` as the bind address; this allows both protocols on all network interfaces. Only ports with numbers from **`1`** to **`65535`** may be used.



Mobile client ports:

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

☑ Enable HTTP bind address

● All interfaces (0.0.0.0) ○ Port: 8082

□ Enable HTTPS bind address

● All interfaces (0.0.0.0) ○ 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[89]. 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[89] for anonymous[89].

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	`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

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Cache-Control" content="no-store"/>
    <title>Customized Login</title>
  </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>
          <!-- The user to login -->
          <tr>
            <td>
              <label for="username">Username:</label>
            </td>
            <td>
              <input type="text" name="username" id="username" size="30"/>
            </td>
          </tr>
          <!-- The password of the user -->
          <tr>
            <td>
              <label for="password">Password:</label>
            </td>
            <td>
              <input type="password" name="password" id="password" size="30"/>
            </td>
          </tr>
        </table>
        <!-- The Active Directory domain details -->
        <h2>Active Directory Login:</h2>
        <table>
          <tr>
            <td>
              <label for="providernameprefix">Domain prefix:</label>
            </td>
            <td>
              <input type="text" name="providernameprefix" id="providernameprefix"
value=""/>
            </td>
          </tr>
          <tr>
            <td>
              <label for="providernamesuffix">Domain suffix:</label>
            </td>
            <td>
              <input type="text" name="providernamesuffix" id="providernamesuffix"
```

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

index.html

```
<html>
  <head>
    <meta http-equiv="Cache-Control" content="no-store" />
    <title>Custom Index</title>
  </head>
  <body>
    <img alt="Logo" src="/index/static/logo.png"></img>
    <hr/>
    <p><a href="/do_logout">Logout</a></p>
    <h1>MobileTogether Custom Login</h1>
    <p><a href='/run?d=/public/About'>Start the About app</a></p>
    <p><a href='/run?d=/public/DateCalc'>Start the Date Calculator app</a></p>
    <p><a href='/run?d=/public/WorldPopulation'>Start the World Population
Statistics app</a></p>
  </body>
</html>
```

Note: If the user is a domain user, the login credentials will have a form something like this: **`domainPrefix@domainSuffix`**. For example: If the domain user is **`someUserName@somedomain.altova.com`**, the domain prefix is **`someUserName`**, and the domain suffix is **`@somedomain.altova.com`**.

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 then be further extended. For a list of privileges, see Available Privileges 59.

▼ 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 by using a name-and-password combination. The services exposed relate typically to access to MobileTogether solutions and their related data.

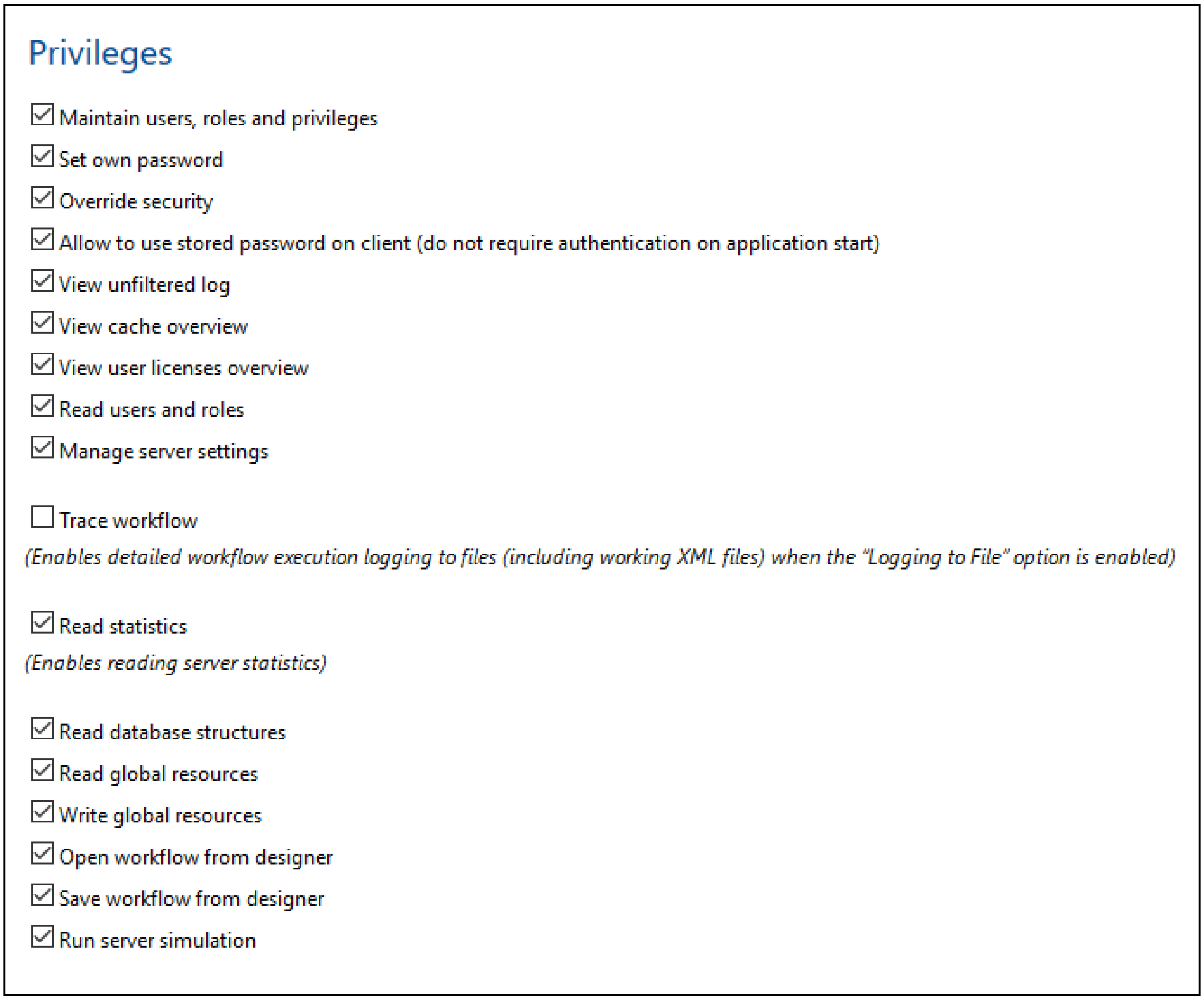
Two special users are predefined:

`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 to 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 statistics

(Enables reading server statistics)

☑ Read database structures

☑ 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*[100] 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:

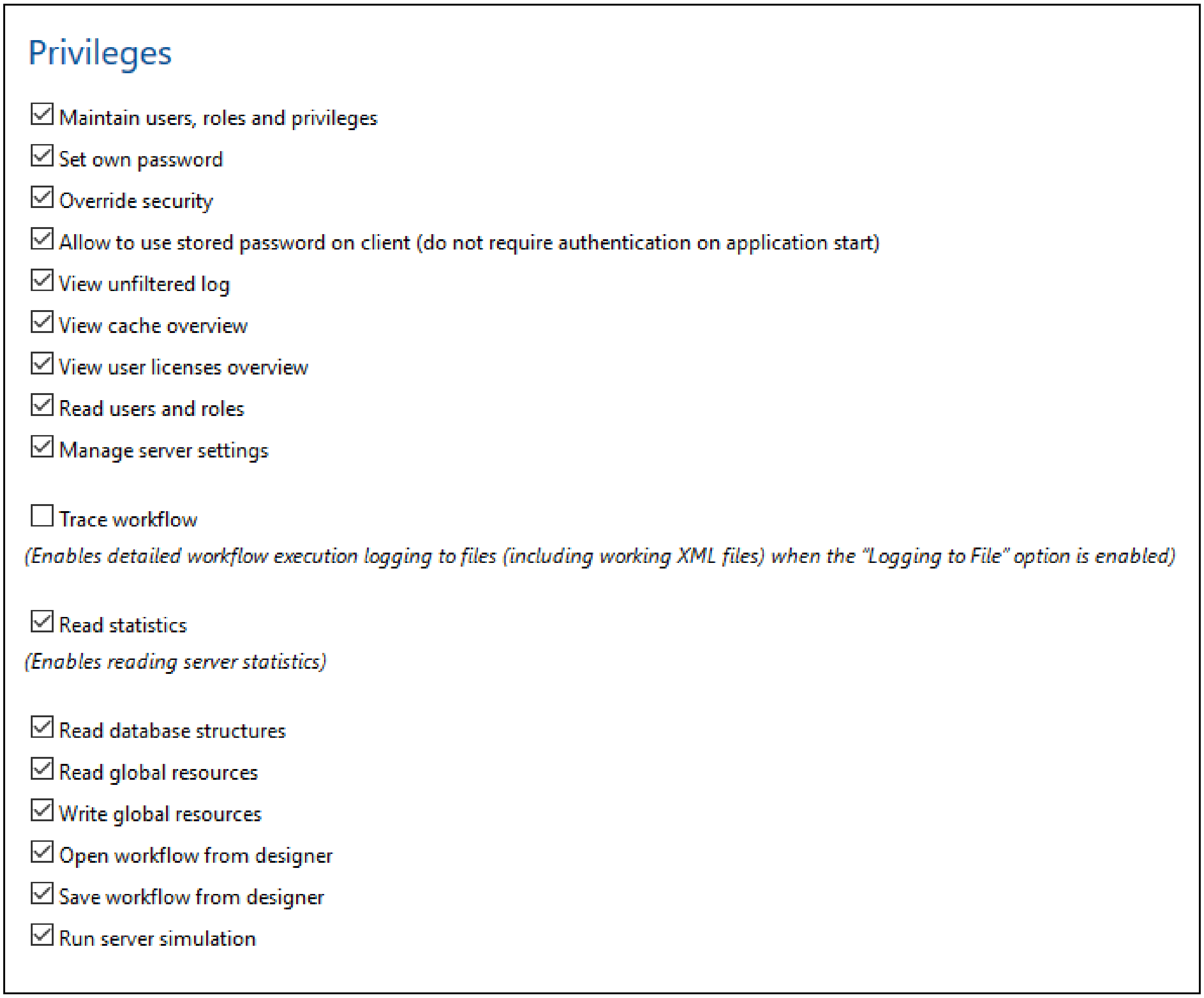
- **`authenticated`** is automatically assigned to every user **except `anonymous`**. So a user with a name-and-password is assigned the `authenticated` role.
- **`all`** is automatically assigned to every user **including `anonymous`**.
- **`workflow-designer`** is assigned to users that design workflows 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.

- **`admin`** has all available privileges and is intended for users that are to function as administrators.

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* 87 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 statistics
(Enables reading server statistics)

☑ Read database structures
☑ 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 to use 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 110.

⊟ *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 statistics*

Server statistics are tracked in an internal database, and can be read by opening the **`statistics.mtd`** solution. This privilege allows the user to read server statistics. Activate the feature by setting a non-zero number of days as the period for which statistics should be tracked 110. See the description of the Statistics setting 110 for more information.

⊟ *Read database structures*

Allows the user with this privilege to have read/write access to databases on the server. (Write access is implicit, assuming that the server is accessed via an administrator port and the *Manage server settings* privilege has been granted). If this privilege has not been granted, the setting *Server-side DB Connections* 110 is not displayed.

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 | Deploy 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 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.

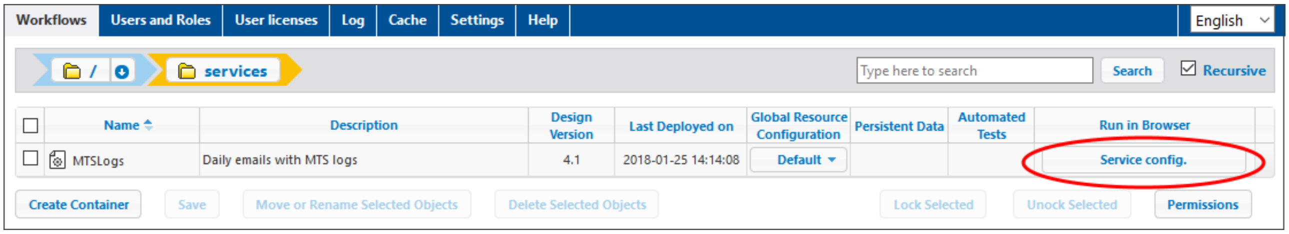
4.8 Configure Services

A server service is a set of MobileTogether actions that is created in MobileTogether Designer as a service solution and saved as a `.mtd` file. A service solution is deployed from MobileTogether Designer to **MobileTogether Server Advanced Edition**. The actions defined in the service are executed when a specified set of MobileTogether Server conditions (or triggers) is met. This section describes how to define these triggers. You can create multiple triggers for a service, and you can enable or disable any of the defined triggers.

Note: The solution file (`.mtd` file) of the service must be created in MobileTogether Designer. See the MobileTogether Designer documentation for details.

Accessing a service's configuration interface

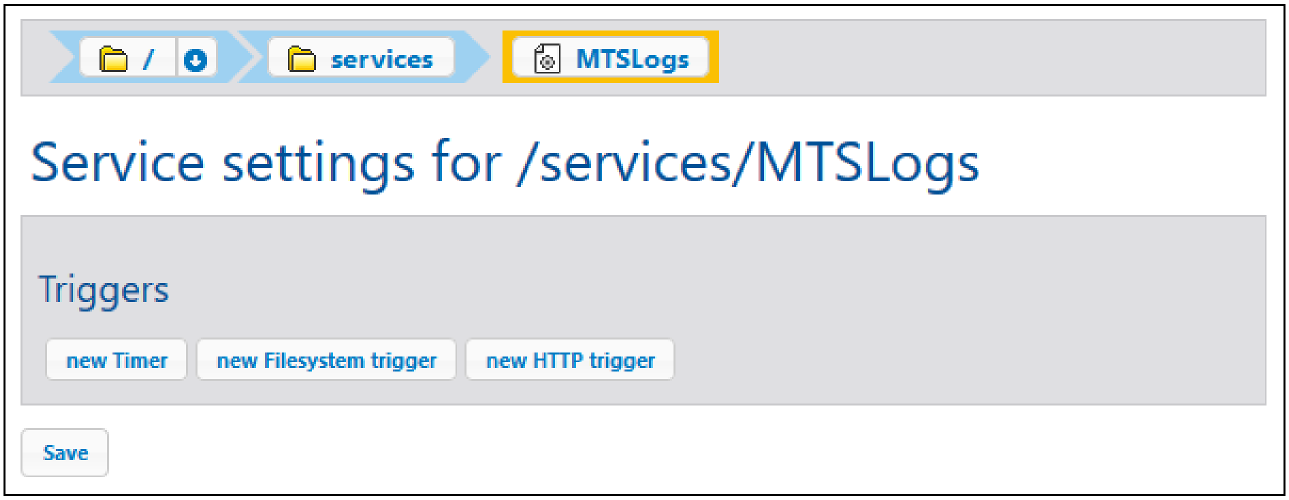
If a service has been deployed (from MobileTogether Designer), then it is listed in the Workflows tab just like any other solution. A service can be distinguished from other solutions by the **Service config** button in the *Run in Browser* column (*see screenshot below*). In the screenshot below, a service named **`MTSLogs`** has been deployed to the **`/services`** container. To access a service's configuration (or settings) interface, click **Service config**.



☐	Name	Description	Design Version	Last Deployed on	Global Resource Configuration	Persistent Data	Automated Tests	Run in Browser
☐	MTSLogs	Daily emails with MTS logs	4.1	2018-01-25 14:14:08	Default			Service config.

The Service Configuration (Settings) interface

The service's configuration (or settings) interface enables you to define and manage the triggers that run the service (*see screenshot below*).



Service settings for /services/MTSLogs

Triggers

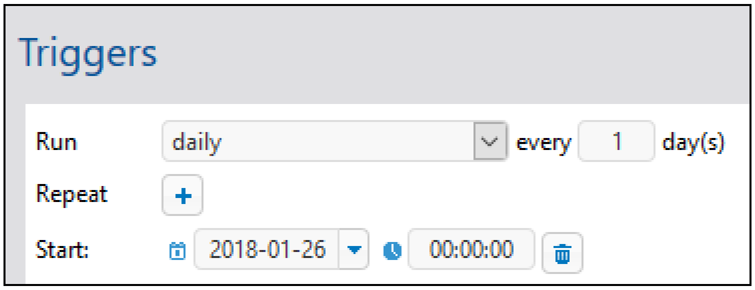
You can create the following types of triggers:

- *Timer triggers*[64], which enable you to specify at what time and with what frequency within a specified period you want the service to run.
- *File system triggers*[65], which enable you to trigger a service by checking for changes to a file or directory on the server.
- *HTTP triggers*[66], which enable you to trigger a service by checking for changes to a resource at a specified URI location.

To add a trigger, click the button corresponding to the trigger type. Each type of trigger is described in more detail in the sub-sections of this section. After a trigger has been created, use the buttons on the right-hand side of the trigger to manage it.

▶	Runs the service immediately. Enabled after settings are saved. Unavailable for filesystem triggers: redundant because any file/dir change would trigger the service.
⧉	Duplicate trigger.
🗑	Delete trigger.
↶	Undo a previous delete action.

Some trigger fields have the + and 🗑 buttons displayed next to them. You can use these buttons to set or clear the value of the trigger field. The value is considered set when it is visible in the page. For example, in the screenshot below, the value of *Repeat* is not set, while the value of *Start* is set to **`2018-01-26`**, **`00:00:00`**.



Triggers

Run: daily every 1 day(s)

Repeat: +

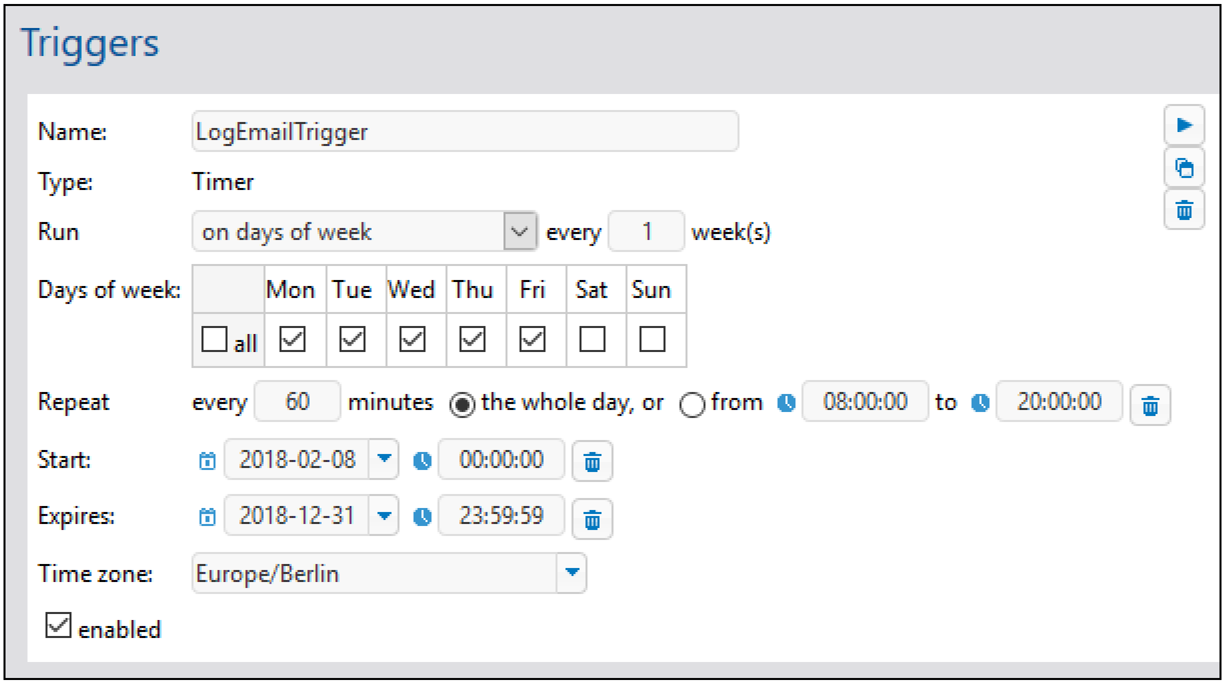
Start: 2018-01-26 00:00:00

Saving the settings of the service

After you have set the triggers of the service, click **Save** at the bottom of the page to save the settings.

4.8.1 Timer Triggers

A timer trigger enables you to define at what time and with what frequency within a specified period you want the service to run. The screenshot below (with UI opened in Firefox) illustrates how to define the settings of a timer trigger.



Triggers

Name:	LogEmailTrigger
Type:	Timer
Run	on days of week every 1 week(s)

Days of week:

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
□ all	☑	☑	☑	☑	☑	□	□

Repeat	every 60 minutes ● the whole day, or ○ from 08:00:00 to 20:00:00
Start:	2018-02-08 00:00:00
Expires:	2018-12-31 23:59:59
Time zone:	Europe/Berlin

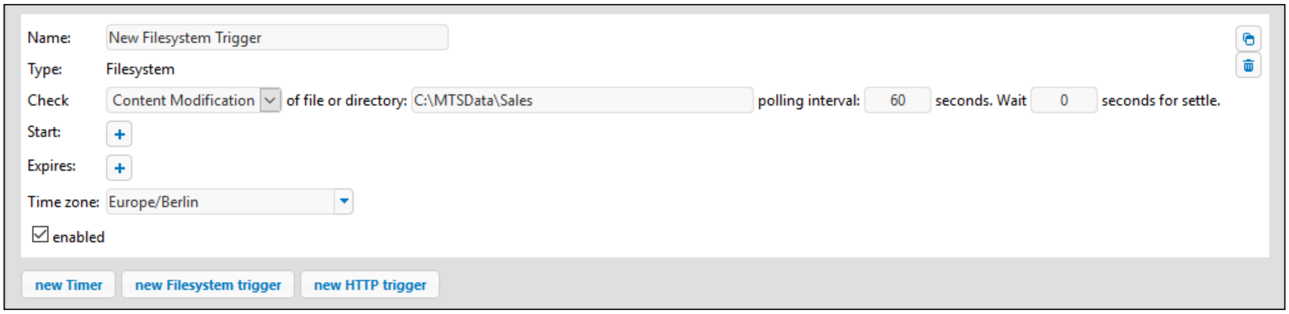
☑ enabled

The trigger is defined with the help of the following parameters:

- *Name:* The trigger's name is a string that serves as the trigger's identifier.
- *Run:* Defines whether the trigger should fire once or periodically every **N** number of days.
- *Repeat:* Defines the frequency of the service: every **x** minutes within a period you specify.
- *Start, Expires:* Defines, respectively, the start and end time of the period within which the service will run.
- *Time zone:* Specifies the timezone of the values in the *Start* and *Expires* fields.
- *Enabled:* This check box allows you to enable/disable the trigger.

4.8.2 File System Triggers

A file system trigger enables you to monitor a file or directory for changes such as newly added files or modified files (note that deleted files cannot be monitored). You can configure the polling interval, and you can optionally set the start and expiry date of the trigger. You can also use wildcards to filter specific files of the directory. The screenshot below (with UI opened in Firefox) illustrates how to define the settings of a file system trigger.



Name: New Filesystem Trigger

Type: Filesystem

Check Content Modification of file or directory: C:\MTSData\Sales polling interval: 60 seconds. Wait 0 seconds for settle.

Start: +

Expires: +

Time zone: Europe/Berlin

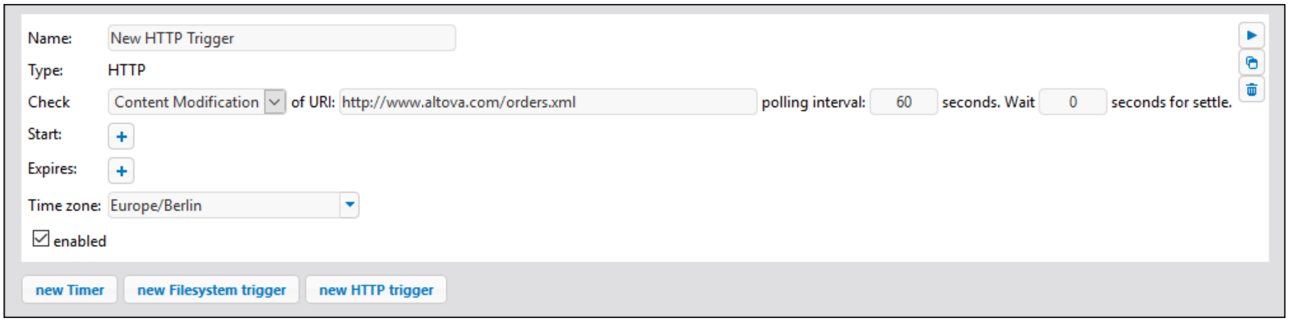
☑ enabled

The trigger is defined with the help of the following parameters:

- *Name:* The trigger's name is a string that serves as the trigger's identifier.
- *Check Content:* Computes and stores a hash code of the specified file or files in the directory. After the polling interval has passed, the hash code is recomputed and compared with the stored value/s. If there is a difference, the trigger fires. (Note that this can place considerable load on the server when a directory is checked.) The trigger also fires if a new file has been added to a directory or if a date has changed.
- *Check Modified Date:* Checks the last-modified timestamp. If this has changed, then the trigger fires.
- *Polling interval:* Specifies the frequency, in seconds, with which the file or directory will be polled.
- *Wait N seconds to settle:* Defines the time in seconds that the server will wait before starting the next service.
- *Start, Expires (optional):* Defines, respectively, the start and end time of the period within which the service will run.
- *Time zone:* Specifies the timezone of the values in the *Start* and *Expires* fields.
- *Enabled:* This check box allows you to enable/disable the trigger.

4.8.3 HTTP Triggers

An HTTP trigger enables you to monitor a URI for changes by checking for changes to the **`Last-Modified`** and **`Content-MD5`** HTTP header fields. You can configure the polling interval, and you can optionally set the start and expiry date of the trigger. The screenshot below (with UI opened in Firefox) illustrates how to define the settings of a HTTP trigger.



Name: New HTTP Trigger

Type: HTTP

Check Content Modification of URI: http://www.altova.com/orders.xml polling interval: 60 seconds. Wait 0 seconds for settle.

Start: +

Expires: +

Time zone: Europe/Berlin

☑ enabled

The trigger is defined with the help of the following parameters:

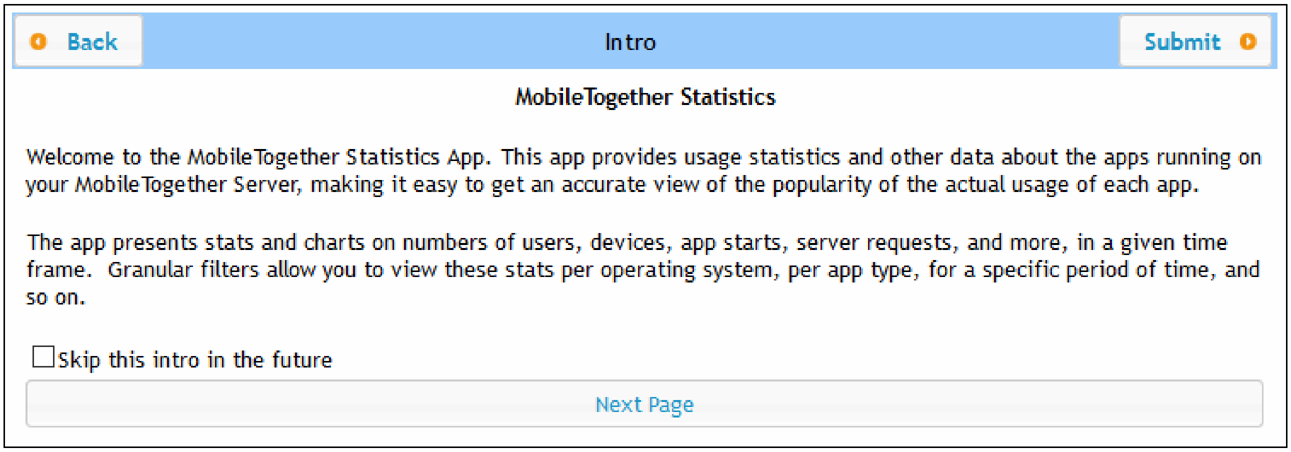
- *Name:* The trigger's name is a string that serves as the trigger's identifier.

- *Check Content:* Checks the optional HTTP header **`Content-MD5`**. This is a 128-bit "digest" used as a message integrity check. If the header has changed after the polling interval has passed, then the trigger fires. If the header is not provided by the server at the HTTP location, then the content is retrieved and hashed locally; the hashes are compared at subsequent polls.
- *Check Modified Date:* Checks the HTTP header **`Last-Modified`**. If the header is missing, the **`Content-MD5`** header is checked (*see above*).
- *Polling interval:* Specifies the frequency, in seconds, with which the URI will be polled.
- *Wait N seconds to settle:* Defines the time in seconds that the server will wait before starting the next service.
- *Start, Expires (optional):* Defines, respectively, the start and end time of the period within which the service will run.
- *Time zone:* Specifies the timezone of the values in the *Start* and *Expires* fields.
- *Enabled:* This check box allows you to enable/disable the trigger.

4.9 Solution Usage Statistics

Statistics of solution usage can be viewed in the **Statistics** solution, which is located by default in the **/admin** container. The **Statistics** solution displays a variety of statistics about individual solutions over a user-selected period. A variety of filters is available, which enables you to see such usage data as the number of users, the type of device or OS, peak-time usage, etc.

The screenshot below shows the intro page of the **Statistics** solution.



MobileTogether Statistics

Welcome to the MobileTogether Statistics App. This app provides usage statistics and other data about the apps running on your MobileTogether Server, making it easy to get an accurate view of the popularity of the actual usage of each app.

The app presents stats and charts on numbers of users, devices, app starts, server requests, and more, in a given time frame. Granular filters allow you to view these stats per operating system, per app type, for a specific period of time, and so on.

☐ Skip this intro in the future

Statistics solution: setting up

From MobileTogether Server version 4.0 onwards, the **Statistics** solution is pre-deployed with MobileTogether Server, and is located in the **/admin** container. The **Statistics** solution is periodically updated to provide improved reporting. To make use of the latest features of the solution, we recommend that you update to the latest version of the **Statistics** solution.

If your version does not have the **Statistics** solution pre-deployed (because it is an older version than 4.0) or if you want to update to the latest version of the solution, do the following:

1. Update your MobileTogether Server software to the latest version (currently 9.1).
2. Access the MobileTogether Server administrator interface[76] in a web browser by typing this URL: **http://<serverIPAddressOrName>:8085/**.
3. Enter your login information and go to the Workflows tab[78].
4. Click **Create Container**, type **admin** as the container name, and click **Save and go there**.
5. In MobileTogether Designer, open the **Statistics.mtd** file. This file is located in the **Solutions** folder of your MobileTogether Server AppData folder (*see table below*).
6. After the file **Statistics.mtd** has been opened in MobileTogether Designer, deploy it to the **/admin** container of MobileTogether Server. Use MobileTogether Designer's menu command **File | Deploy to MobileTogether Server** to do this.
7. In the MobileTogether Server administrator interface[76], go to the Settings[110] tab and, in the Statistics pane of the Misc tab[126], set *Statistics Limit* to a positive integer to activate the tracking of statistics data.
8. To see solution statistics from this time onwards, start the **Statistics** solution. Do this as follows: In MobileTogether Server, go to the Workflows tab[78], open the **/admin** container, and start the

Statistics solution. Alternatively, enter this URL: **http://<serverIPAddressOrName>:8085/run?d=/admin/Statistics/**.

Note: You can deploy the **Statistics** solution to any container you like. To run the solution, modify the solution's URL to take the correct container into account.

Location of the MobileTogether Server AppData folder on various operating systems

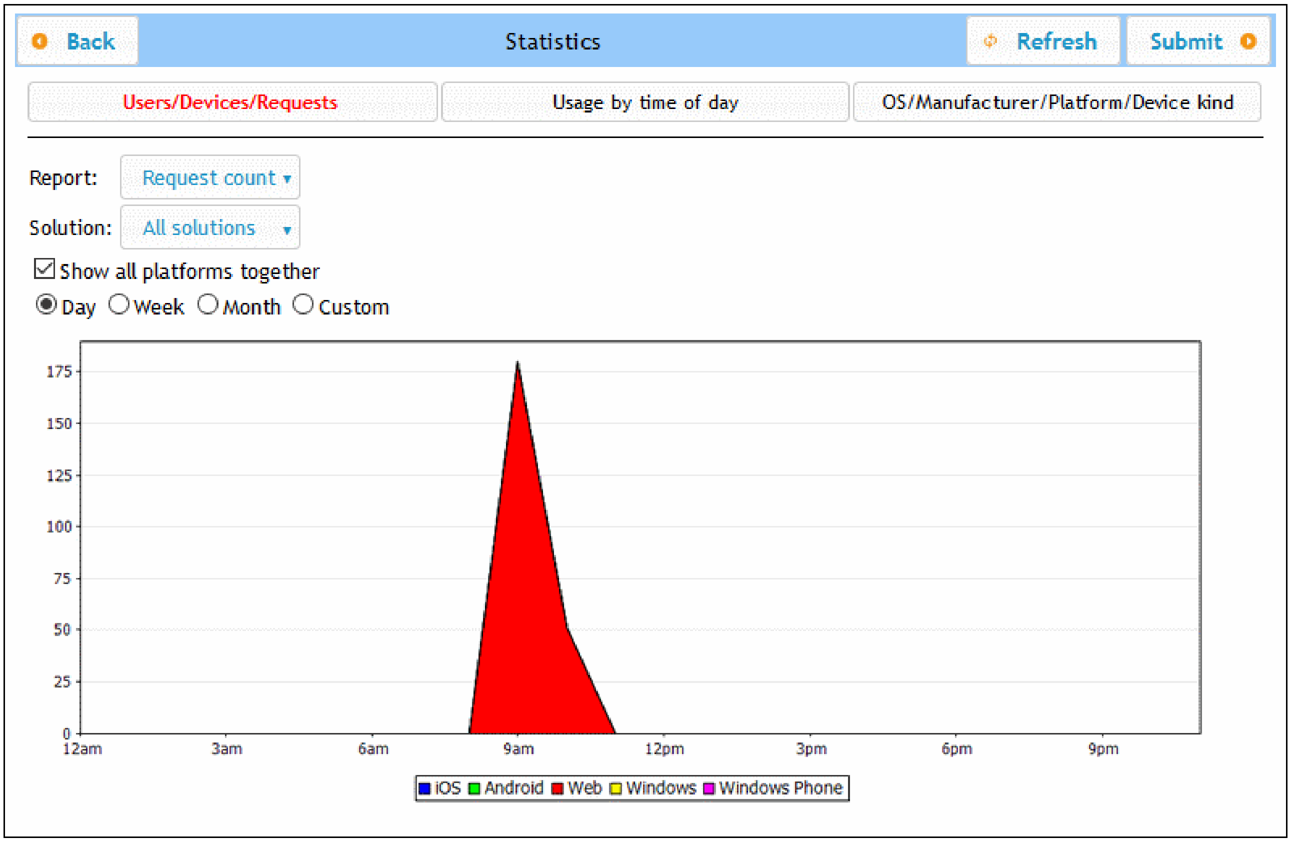
Linux	`/var/opt/Altova/MobileTogetherServer`
Mac	`/var/Altova/MobileTogetherServer`
Windows	`C:\ProgramData\Altova\MobileTogetherServer`

Statistics solution: description

The **Statistics** solution interface (*screenshot below*) consists of three tabs:

- Users/Devices/Requests
- Usage by time of day
- OS/Manufacturer/Platform/Device kind

The name of the active tab is displayed in red (*see screenshot*).



Report: Request count

Solution: All solutions

☑ Show all platforms together

● Day ○ Week ○ Month ○ Custom

Each tab has two or three filters. One of the filters in each tab is always the *Solution* filter. This enables you to select a single solution from all the solutions deployed to the server; alternatively, you can select all solutions. The other filter/s enable you to select what category of statistics to display. You can also select a time interval for which the statistics are to be displayed.

Users/Devices/Requests
Shows statistics for different platforms separately, with each platform being indicated by a different color (*see screenshot above*). If you uncheck *Show all platforms together*, you can use the *Platform* filter to select the platform (iOS, Android, Web, Windows, and Windows Phone) to display. In the *Report* filter, you can select from among the following:

- *Users:* the number of users.
- *Devices:* the number of devices.
- *Request count:* The number of requests.
- *Request time total (sec):* Total time (in seconds) used to process requests.
- *Request time average (ms):* Average time (in milliseconds) for processing a single request.
- *Request time maximum (ms):* Processing time (in milliseconds) of the request that took the longest to process.
- *Solution starts:* The number of solution starts, subdivided by platform.
- *Incoming traffic (MB):* Total incoming traffic (in MB) from MT clients; does not include HTTP traffic from other servers to the MT server.
- *Outgoing traffic (MB):* Total outgoing traffic (in MB) to MT clients; does not include HTTP traffic from the MT server to other servers.
- *Files read:* Number of files read on the server.
- *Files read size (MB):* Total size of all files read on the server.
- *Files written:* Number of files written on the server.
- *Files written size (MB):* Total size of all files written on the server.
- *Database read/writes:* Number of read/writes from/to DBs used in the solution; does not include MT internal database.
- *HTTP requests:* Number of HTTP requests from MT server to other servers.
- *HTTP requests incoming traffic (MB):* Incoming HTTP traffic (in MB) from other servers to the MT server; does not include traffic with MT clients.
- *HTTP requests outgoing traffic (MB):* Outgoing HTTP traffic (in MB) from the MT server to other servers; does not include traffic with MT clients.
- *Images:* Number of images loaded on the server; does not include charts.
- *Charts:* Number of charts created on the server.
- *Chart time total (sec):* Total time (in seconds) used to generate charts on server.
- *Chart time average (ms):* Average time (in milliseconds) for generating a single chart.
- *Chart time maximum (ms):* Longest time (in milliseconds) needed to generate a single chart.

If you select *Show all platforms together*, then all platforms (iOS, Android, Web, Windows, and Windows Phone) are shown together in one graphic, with each platform being represented by a different color. If *Show all platforms together* is unselected, then the graphic for each platform can be shown separately by selecting the respective platform in the *Platform* filter.

Usage by time of day
Shows intensity of usage of the selected solution in two-hour time segments across a period of 24 hours for each day of the past week. If you uncheck *Show all platforms together*, you can use the *Platform* filter to select the platform (iOS, Android, Web, Windows, and Windows Phone) to display. In the *Report* filter, you can select from among the following:

- *Users:* the number of users.
- *Devices:* the number of devices.
- *Request count:* The number of requests.
- *Request time total (sec):* Total time (in seconds) used to process requests.
- *Request time average (ms):* Average time (in milliseconds) for processing a single request.
- *Request time maximum (ms):* Processing time (in milliseconds) of the request that took the longest to process.
- *Solution starts:* The number of solution starts, subdivided by platform.
- *Incoming traffic (MB):* Total incoming traffic (in MB) from MT clients; does not include HTTP traffic from other servers to the MT server.
- *Outgoing traffic (MB):* Total outgoing traffic (in MB) to MT clients; does not include HTTP traffic from the MT server to other servers.
- *Files read:* Number of files read on the server.
- *Files read size (MB):* Total size of all files read on the server.
- *Files written:* Number of files written on the server.
- *Files written size (MB):* Total size of all files written on the server.
- *Database read/writes:* Number of read/writes from/to DBs used in the solution; does not include MT internal database.
- *HTTP requests:* Number of HTTP requests from MT server to other servers.
- *HTTP requests incoming traffic (MB):* Incoming HTTP traffic (in MB) from other servers to the MT server; does not include traffic with MT clients.
- *HTTP requests outgoing traffic (MB):* Outgoing HTTP traffic (in MB) from the MT server to other servers; does not include traffic with MT clients.
- *Images:* Number of images loaded on the server; does not include charts.
- *Charts:* Number of charts created on the server.
- *Chart time total (sec):* Total time (in seconds) used to generate charts on server.
- *Chart time average (ms):* Average time (in milliseconds) for generating a single chart.
- *Chart time maximum (ms):* Longest time (in milliseconds) needed to generate a single chart.

OS/Manufacturer/Platform/Device kind

For each criterion (OS, manufacturer, platform, and device kind), shows usage of the selected solution in terms of proportions of that criterion's instances. For example, for the platform criterion, each platform is shown as a proportion of total usage. The graphic in each case is a pie chart, with each instance of the criterion shown as a slice of the pie. In the *Report* filter, you can select from among the following:

- *OS:* Each OS is shown in a different color.
- *Manufacturer:* Each manufacturer is shown in a different color.
- *Platform:* Each platform is shown in a different color.
- *Device kind:* Each device kind is shown in a different color.

4.10 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* 52 setting
SSL	Whether communication should be SSL encrypted or not 47
User name	As what user to log in. This will determine access rights. See Users and Roles 56
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.

Running multiple workflows on web clients

A web client can run multiple workflows (solutions) in parallel, each in a separate tab. Additionally, in each tab, the previous workflow is kept in memory as long as the session is active, and the previous workflow can be reloaded by pressing **F5** (Reload). Note the following: (i) Running multiple workflows in parallel will use up the server's memory; (ii) While a solution is running in the active tab, solutions in background tabs can timeout.

4.11 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 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 110 or the Command Line Interface 134.

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	`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* 110 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 110 of MobileTogether Server (port number, solutions directory, etc)
`mobiletogetherserver.licsid`	File with ID of registered LicenceServer 13 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 | Misc 126 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 45.
3. Copy the backed-up files (*see above*) into the correct folders on the new installation.
4. Start MobileTogether Server 45.

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 72 for details.

4.12 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.

5 Web UI Reference

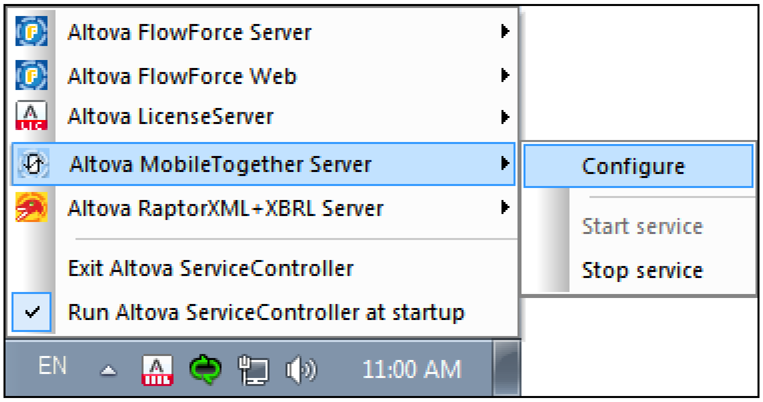
MobileTogether Server has a **Web User Interface** (Web UI) with which you can easily configure MobileTogether Server. The Web UI can be opened in any Internet browser on any of the supported operating systems 9 .

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 119 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).

You can also, in a web browser, enter the following URL: **`http://<serverIPAddressOrName>:8085/`**.

▼ 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 119 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 119 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).

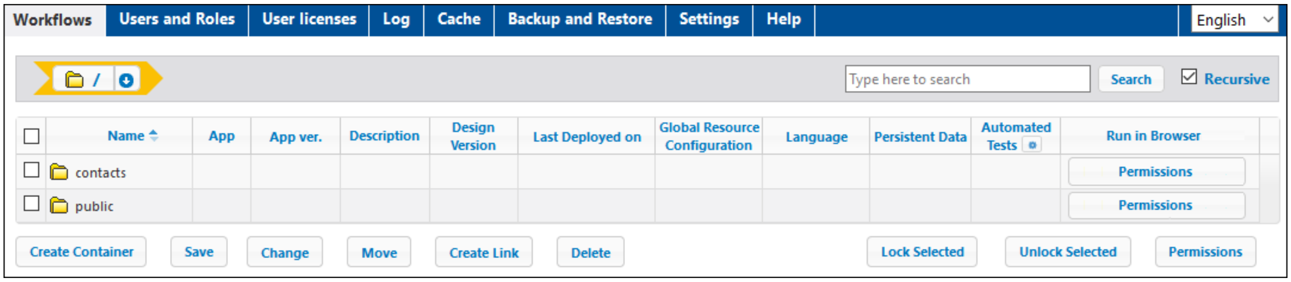
Web UI tabs

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

- Workflows 78: An interface for managing the server's container structure and container permissions.
- Users and Roles 56: 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 102: Shows the mobile devices that are currently licensed, and their license details.
- Log 104: Displays the logged server actions according to the selected filters.
- Cache 106: This tab shows the details of caches that are currently available on the server. Caches can also be activated/deactivated and deleted.
- Backup and Restore 108: Provides (i) settings for configuring backups, and (ii) the ability to restore from available backups.
- Settings 110: MobileTogether Server settings, such as access ports, log settings, and server session timeouts, are defined in this tab.

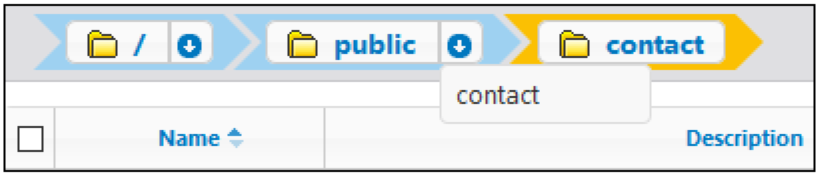
5.1 Workflows

The **Workflows** tab (*screenshot below, showing the Advanced edition*) 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 (also called 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.



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

- The *Workflows* tab initially displays the `root` container, which is denoted by the `"/"` character.
- Click the **Down** arrows next to a container's name to display the sub-containers of that container; click a sub-container in the pop-up list to go to that sub-container.
- To go to a container, click it.
- Every level that you descend in the hierarchy of containers is displayed at the top of the window as a "breadcrumbs" path. The **Down** arrow of each level displays the sub-containers of that container, so you can navigate easily to different containers.



- To **select** a container, click the container's check box. Selections are used for renaming, moving, and deleting containers (*see Functionality below*).

▼ 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
Change	Enabled when a single object is selected. It opens the Change Object dialog, in which you can change the name of the object, assign an icon, and set input parameters (*see below*). You can also open this dialog by clicking the gear icon to the right of the solution's name.
Move	If multiple objects are selected, opens the Move dialog, in which you can move the object to another container. If only one object is selected, you can, additionally to moving, also rename the object.

Create Link	Enabled when a single object is selected. It opens the Create Link dialog, in which you can change the name of the object, assign an icon, set input parameters, and create a shortcut link for the solution in a different container (*see description below*).
Delete	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

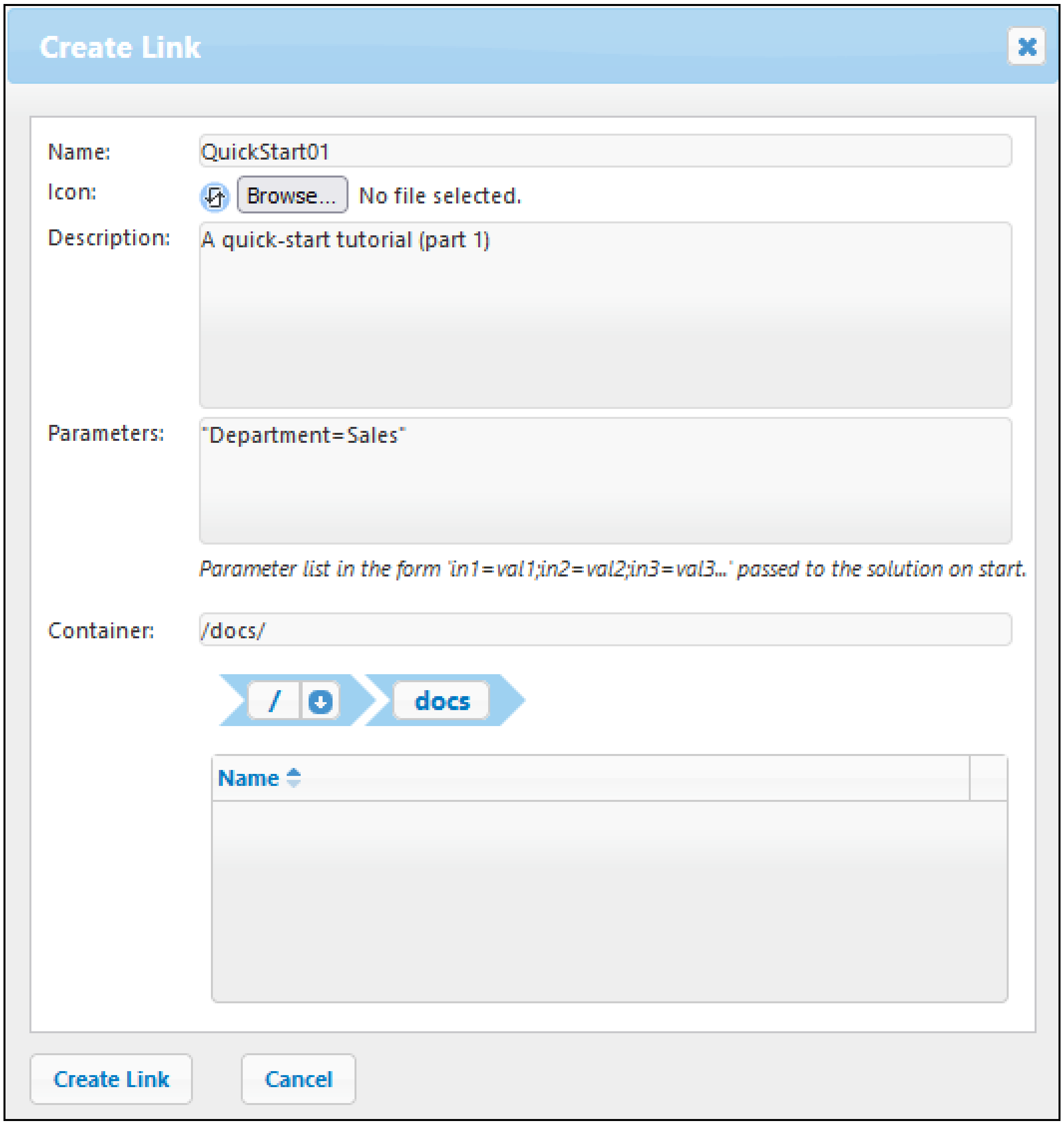
Input parameters

You can specify input parameters in the form **`"Key1=Value1;Key2=Value2;...;KeyN=ValueN"`**. If a value has spaces, use quotes around it. These parameter values are passed to the solution when the solution is started and stored in the solution's **`MT_InputParameters`** global variable (for information about which, see the MobileTogether Designer user manual). Input parameters can then be used inside the solution to achieve different goals. For example, if you specify the input parameter `"Department=Sales"`, then it would be possible to filter your database to only show records of the department named `Sales`.

Note: The **`MT_InputParameters`** global variable of a solution can receive input parameters not only from this MobileTogether Server entry point, but also from design-related originating points (see the MobileTogether Designer user manual). In such cases, the input parameters are merged. If the key name is the same, then the value defined for shortcuts in MobileTogether Server wins and will be the value that is assigned to the key.

Create shortcut link

You can create a shortcut link to a solution and place the shortcut link in a container that is not the same as the one containing the solution. This enables a solution to be accessed from different containers. The advantage of this is that by using different input parameters for each shortcut link, you can cause the solution to appear in different ways when opened via different shortcut links. For example, you could create a shortcut link in a container named *Sales* and give it an input parameter `"Department=Sales"`. In the solution, you can specify that when the solution is opened it will be filtered on the value of the department name in its input parameter. So, when the solution is opened via the shortcut link in the *Sales* container, the solution's records will be filtered to show, because of its input parameter, only the records of the *Sales* department. You could create other shortcut links in other containers (say, fro *Accounts* or *Legal*), and set corresponding input parameters for them. When the solution is opened via these links, the records that are displayed would then be filtered for the appropriate department. The solution that is opened in all these cases would be the original solution. The shortcut link feature enables you to present and process the solution in different ways according to the input parameters corresponding to the shortcut link.



Name: QuickStart01

Icon: No file selected.

Description: A quick-start tutorial (part 1)

Parameters: "Department=Sales"

Parameter list in the form 'in1=val1;in2=val2;in3=val3...' passed to the solution on start.

Container: /docs/

Alternatively to the example case described above, the availability of parameters enables various other scenarios. For example, one powerful use case would be the possibility of using an alternative database for the solution.

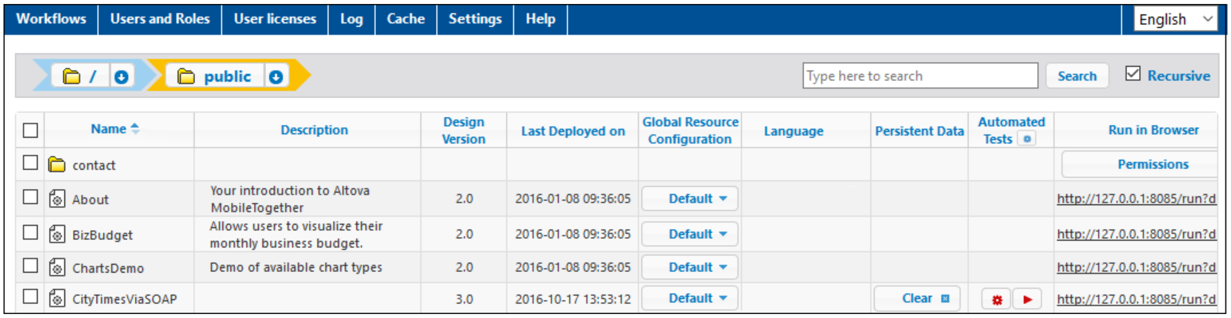
Note, however, that in all these cases, what happens when the parameter value is passed to the solution depends entirely on how the parameter is handled in the design.

Note: If, after a shortcut link has been created, you remove one of its parameters or change the value of any parameter in the dialog above, then you might also need to remove the solution's persistent data in order for the parameter modification to take effect. You can remove a solution's persistent data by clicking the **Clear** button in the solution's *Persistent data* column (*see below:* The display of a container's contents).

Whether it is necessary to remove persistent data or not depends on what has been defined to happen when the parameter value is passed to the solution. If the parameter value has a consequence that is stored as persistent data, then you might want to clear persistent data; otherwise, it is not necessary. Note also the converse effect of removing persistent data. In this case, the solution will not have access to persistent data. It will try to use data supplied via parameters and, if this data does not substitute well for the removed persistent data, then the solution could start with some data missing or with inappropriate data.

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.



□	Name	Description	Design Version	Last Deployed on	Global Resource Configuration	Language	Persistent Data	Automated Tests	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

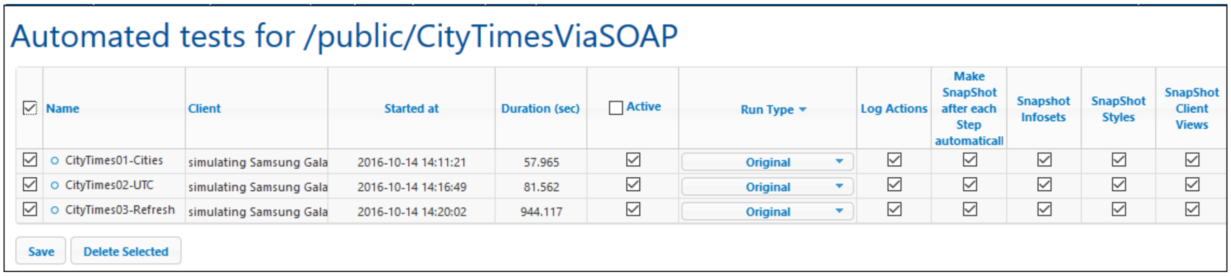
A container contains sub-containers and/or solutions (aka design files or `.mtd` files). The contents of each container are displayed as a tabular list. The columns of the table display the properties of solutions:

- *Name:* Name of the solution file as saved in MobileTogether Designer.
- *App, App version:* The *App* and *App Ver* columns *(see first screenshot at top of page)* 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.
- *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.
- *Language:* If the solution is a service solution 63, then a button with dropdown options for selecting the solution's language is available. The items of the dropdown list are: *Auto* plus the names of the languages defined in the solution. Select the language you want to use. If you select *Auto*, then the language of MobileTogether Server 158 (the server language) is used as the solution's language. If the solution has not been localized in the server language, then the default language of the solution is used as the solution language. If the default language of the solution was not explicitly given a name in the design, then it is represented in the dropdown list as *Default*.
- *Persistent Data:* A **`Clear`** button appears in this column if data has been changed while running the solution and persists in the solution. Click this button if you wish to undo the changes. Also see the note in the *Functionality* section above | *Create Shortcut Link*.

- *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*). Clicking the wheel in the column header filters the display to show only those solutions in the current folder and descendant folders that contain automated tests. 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. If the solution defines server services[63], click the **Service config** button in this column to access the service's configuration interface[63]. (For AppStore Apps, no URL is displayed because the AppStore App cannot be opened in a web browser.)

Automated testing

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



Automated tests for /public/CityTimesViaSOAP

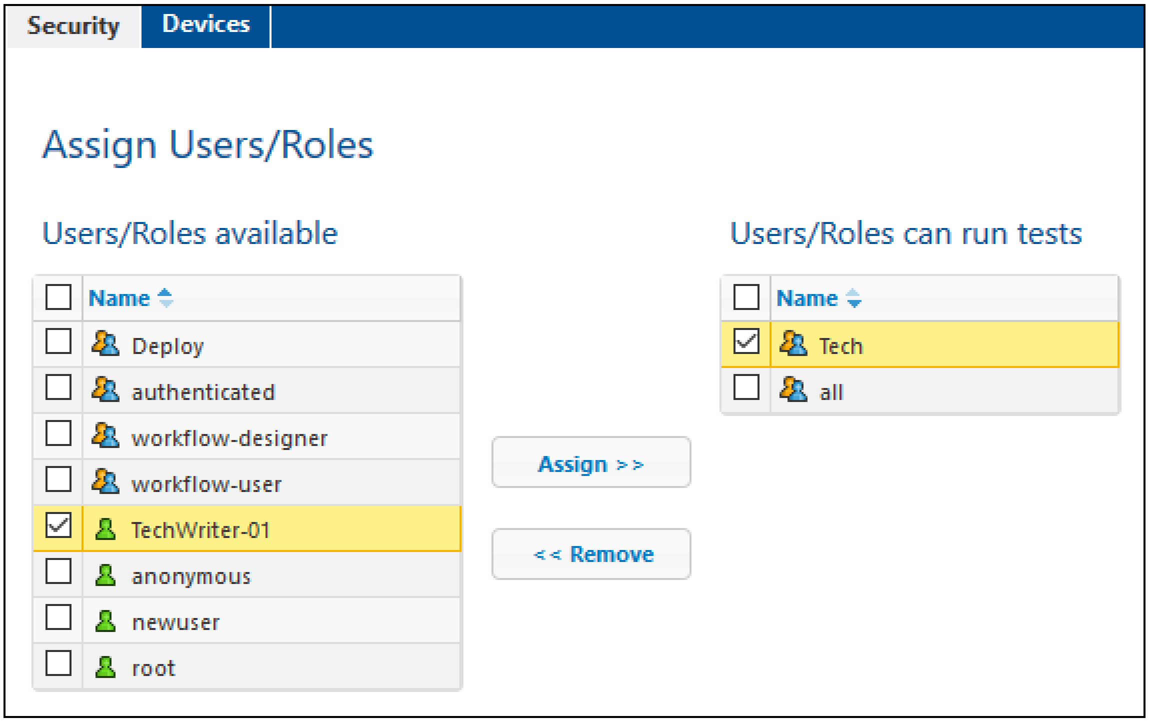
☑	Name	Client	Started at	Duration (sec)	□ Active	Run Type ▾	Log Actions	Make SnapShot after each Step automaticall	Snapshot Infosets	SnapShot Styles	SnapShot Client Views
☑	○ CityTimes01-Cities	simulating Samsung Gala	2016-10-14 14:11:21	57.965	☑	Original ▾	☑	☑	☑	☑	☑
☑	○ CityTimes02-UTC	simulating Samsung 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 can then be played back on the client. If multiple test runs are selected, then all the selected test runs will be played back when automated testing is started on the client. 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. If you want to play back a test run on the Web client, then on the Workflows page, click the **Playback** icon in the solution's *Automated Test* column.
2. Set the speed of the test run in the *Run Type* column. You can set the speed for all test runs at once by selecting the speed in the dropdown list of the column header.
3. 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.
4. 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
In the lower part of the *Automated Tests* page (*screenshot below*), you can specify: (i) what users and roles can run automated tests for the selected solution (in the *Security* tab), and (ii) the devices on which test runs can be carried out (selected in the *Devices* tab).



Security | Devices

Assign Users/Roles

Users/Roles available

□	Name
□	Deploy
□	authenticated
□	workflow-designer
□	workflow-user
☑	TechWriter-01
□	anonymous
□	newuser
□	root

Users/Roles can run tests

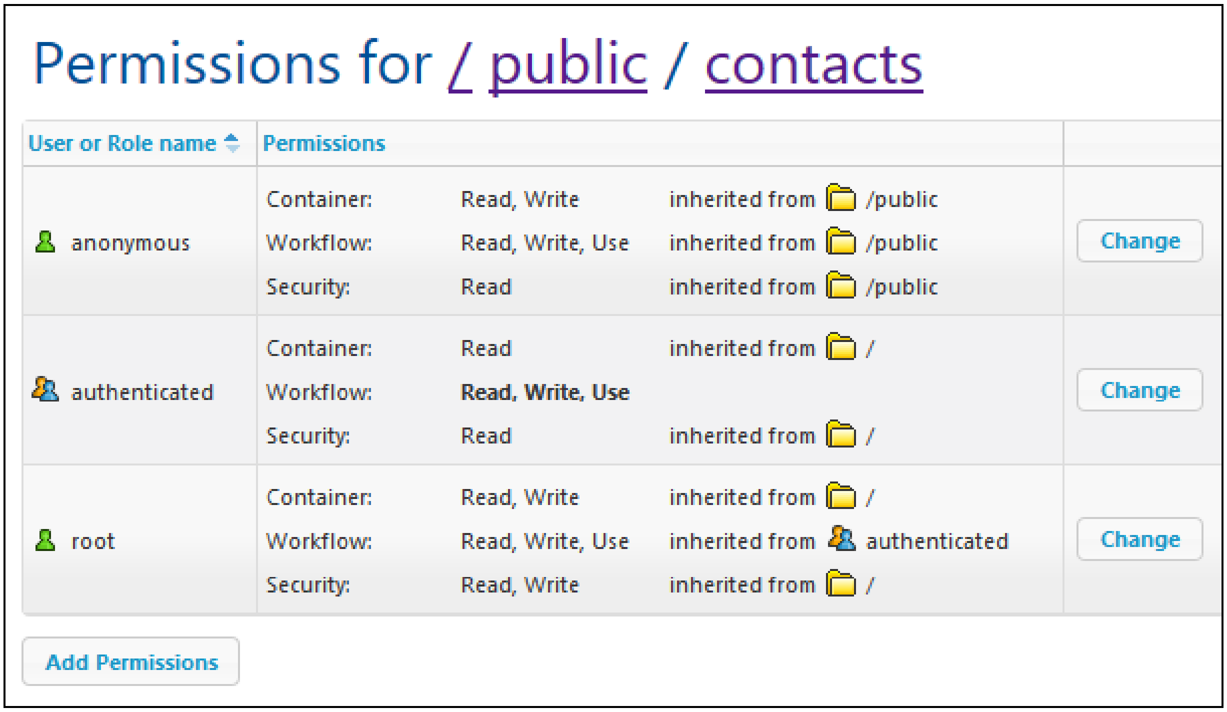
□	Name
☑	Tech
□	all

- Users and roles are selected in the *Security* tab, devices are selected in the *Devices* tab (*see screenshot above*).
- To assign a user/role or device to the Allowed list, select it in the left pane and click **Assign** (*see screenshot above*).
- Remove a user/role or device from the Allowed list by selecting it and clicking **Remove**.
- You can assign or remove multiple selections at a time.
- If no device is assigned to the Allowed list, then test runs for that solution can be run on **all** devices.

Note: All automated tests that were deployed prior to an upgrade of the server to version 4.1 (released 27 February 2018) or later will get security permissions for all users/roles; that is, all users/roles can run automated tests, which is the same behavior as that prior to the upgrade. For automated tests that are deployed subsequent to an upgrade to version 4.1, security permissions are set for **no** user/role; that is, any user or role that may run automated tests must be explicitly specified.

▼ 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: Workflow: Security:	Read, Write Read, Write, Use Read	inherited from /public inherited from /public inherited from /public	Change
authenticated	Container: Workflow: Security:	Read **Read, Write, Use** Read	inherited from / inherited from /	Change
root	Container: Workflow: Security:	Read, Write Read, Write, Use Read, Write	inherited from / inherited from authenticated inherited from /	Change

⊟ *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.
- *Inherit:* Inherit permissions from the parent container.
- *No access:* Access to the container is not granted.

Workflow

- *Read-Use:* The user can run solutions.
- *Read-Write-Use:* The user can additionally overwrite solutions, that is, deploy solutions.
- *Inherit:* Inherit permissions from the parent container.
- *No access:* Access to workflows is not granted.

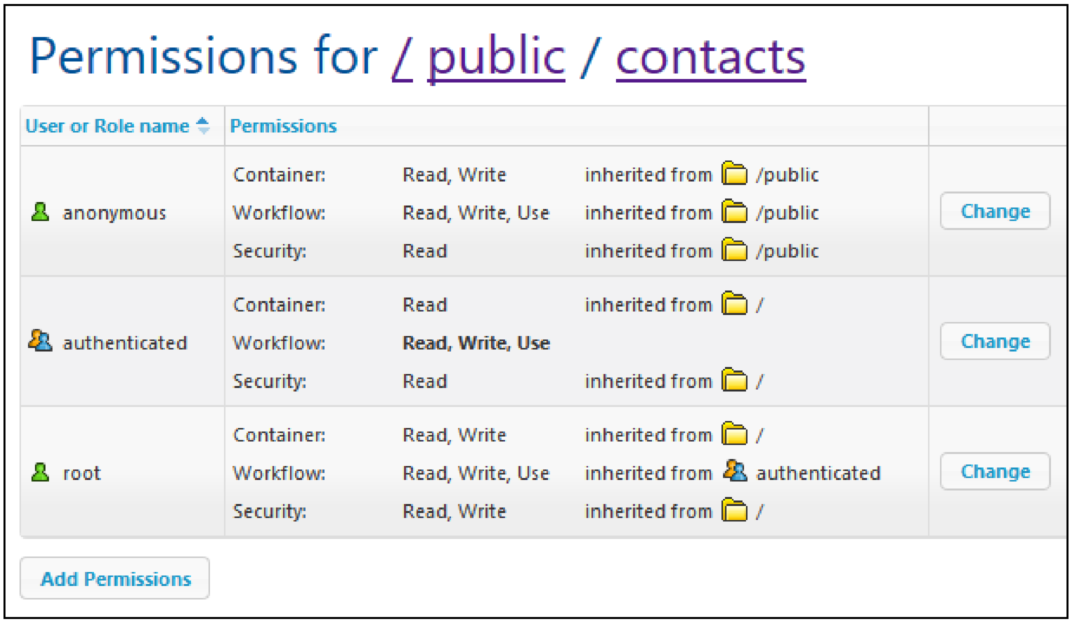
Security

- *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 87), users can read all permission entries.
- *Inherit:* Inherit permissions from the parent container.
- *No access:* Access to the permission list is not granted.

⊟ *Editing the permissions of a container*

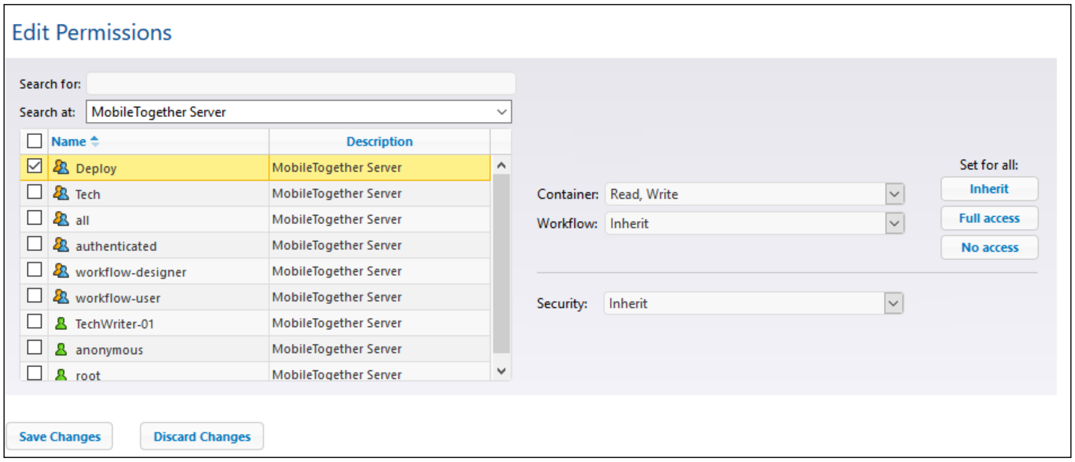
1. Click the **Permissions** button of the container. This takes you to the container's *Permissions* page (*screenshot below*).



Permissions for / public / contacts

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

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

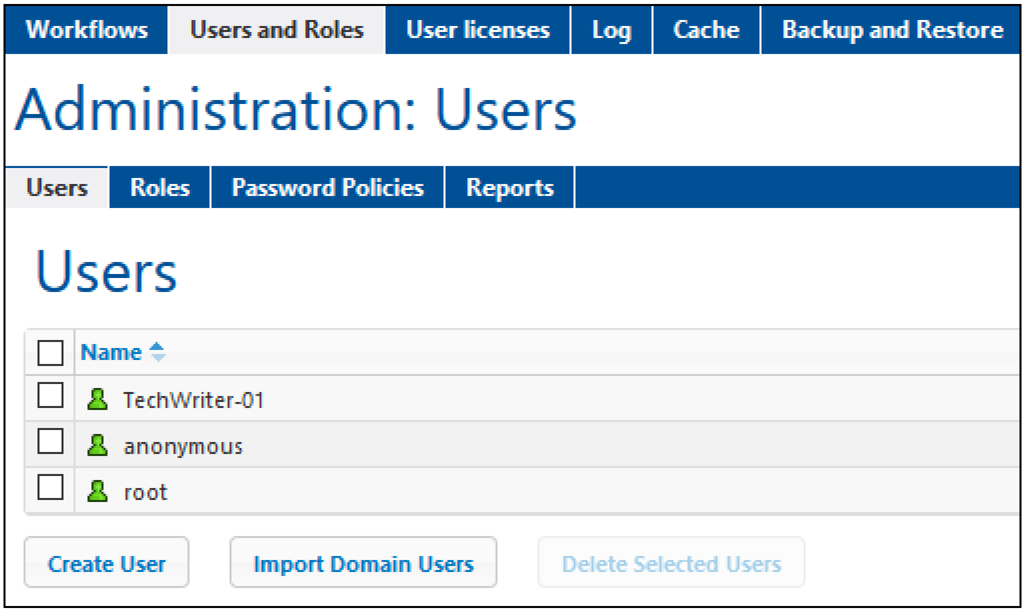
Workflow: Inherit

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[89] and roles[94] that have been defined for MobileTogether Server or for all enabled domains (by selecting, respectively, *MobileTogether Server* or *Directory Service* 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[119] has been enabled in the Settings tab[110].
4. Change the permissions as required. The *Inherit* option causes permissions to be inherited for the container and the workflow from the parent container.

5.2 Users and Roles

The **Users and Roles** tab (*screenshot below shows the Advanced Edition*) 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 by using a name-and-password combination. The services exposed relate typically to access to MobileTogether solutions and their related data.

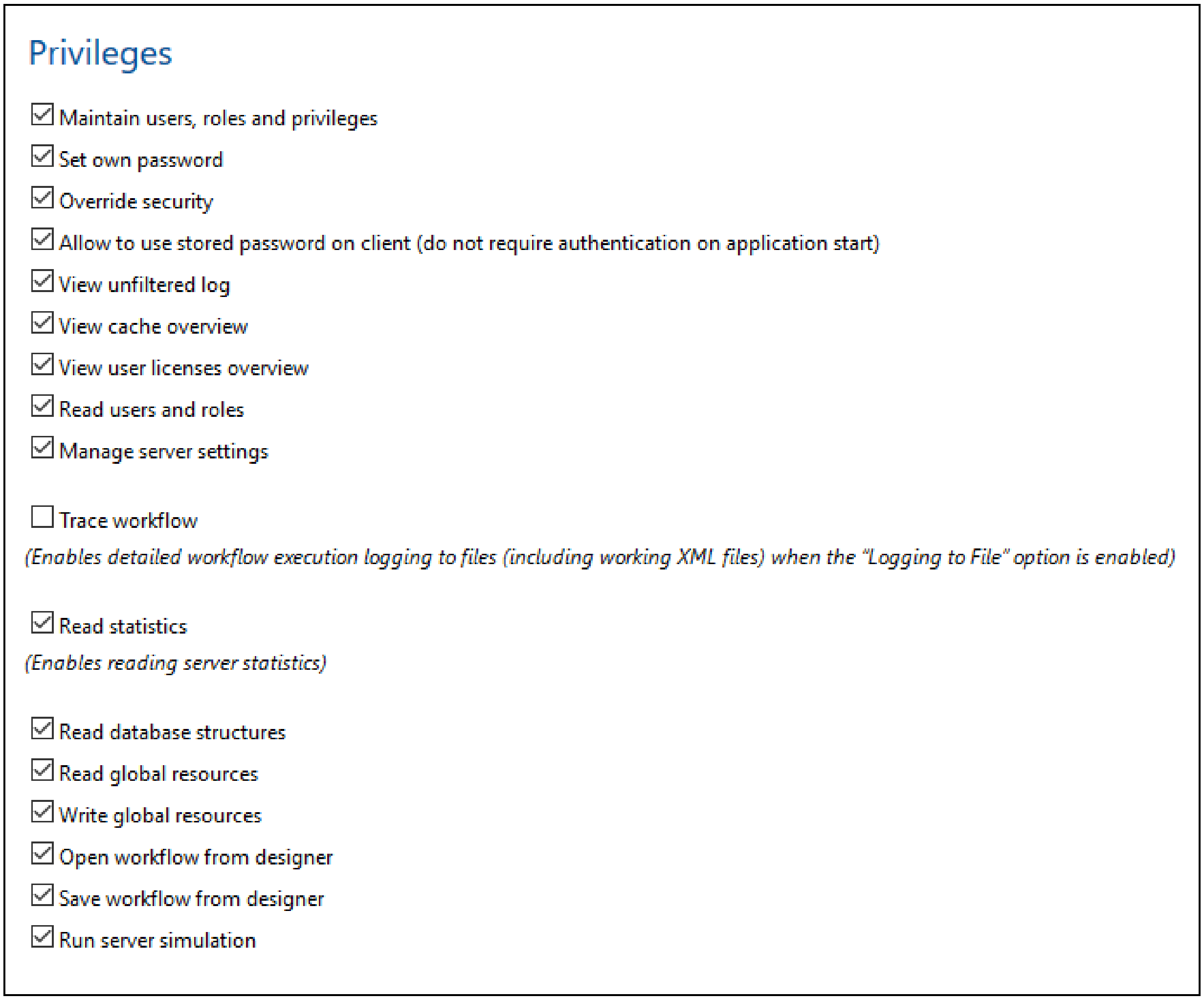
Two special users are predefined:

`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 to 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 statistics
(Enables reading server statistics)

☑ Read database structures
☑ 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* 100 provides a list of all privileges, with each privilege being listed together with all the users/roles that have that privilege.

About Roles

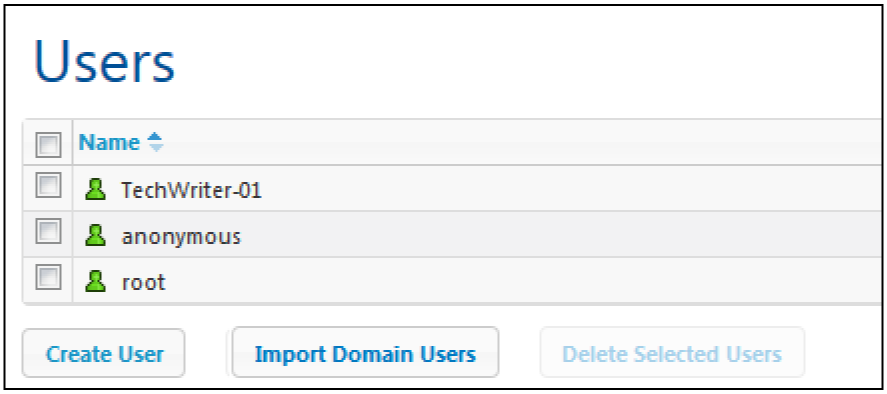
A role defines a set of privileges. It can be assigned to another role or to a user. A role's privileges automatically become the privileges of any other role or any user that the role is assigned to. A user can be assigned any number of roles. As a result, a user will have all the privileges defined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

The following roles are predefined:

- **`authenticated`** is automatically assigned to every user **`except anonymous`**. So a 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.
- **`admin`** has all available privileges and is intended for users that are to function as administrators.

5.2.1 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.



▼ 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 by using a name-and-password combination. The services exposed relate typically to access to MobileTogether solutions and their related data.

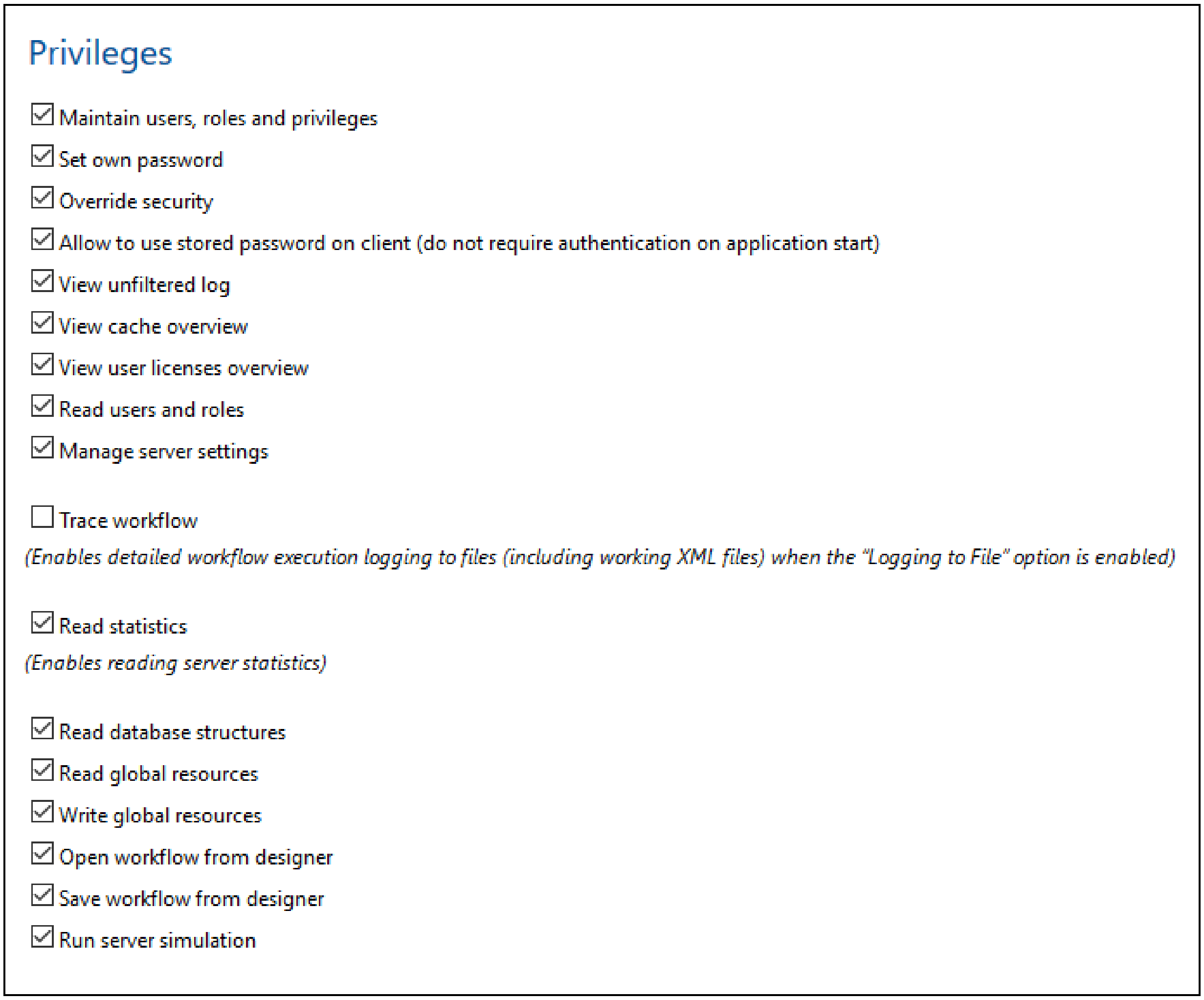
Two special users are predefined:

`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 to 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 statistics

(Enables reading server statistics)

☑ Read database structures

☑ 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*[100] 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 a 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

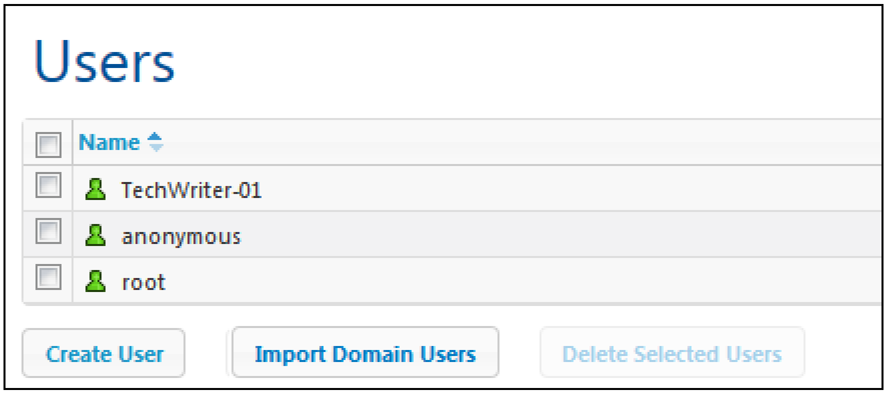
user to access the service interface without needing to log in to the server and start the solution on the client.

- **`admin`** has all available privileges and is intended for users that are to function as administrators.

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.



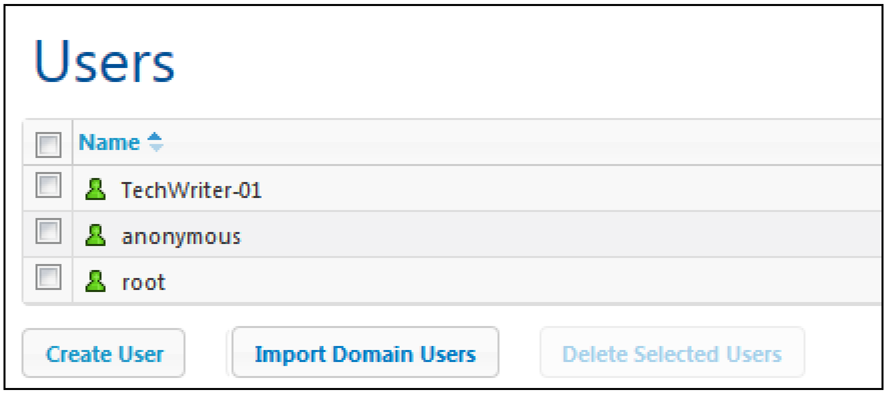
Users

□	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[98].
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[110] 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**.



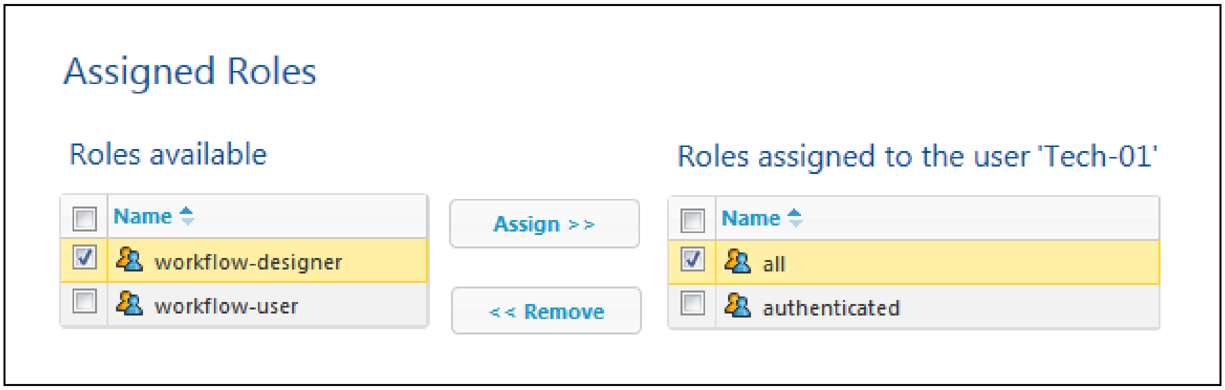
Users

□	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*).



Assigned Roles

Roles available

□	Name
☑	workflow-designer
□	workflow-user

Roles assigned to the user 'Tech-01'

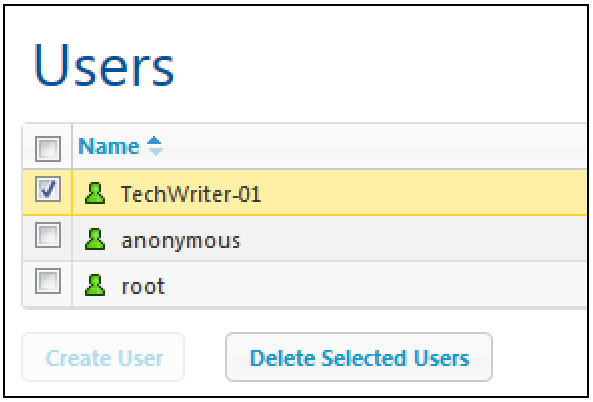
□	Name
☑	all
□	authenticated

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*[100].

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

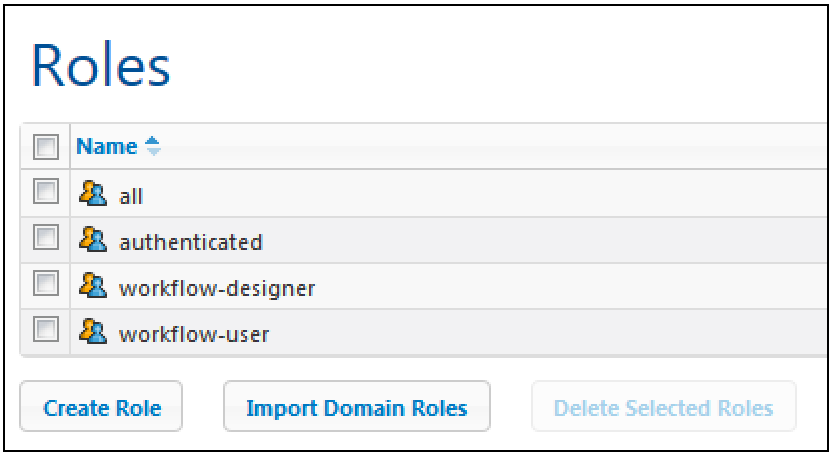
☐	Name
☑	TechWriter-01
☐	anonymous
☐	root

5.2.2 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 a 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.
- **`admin`** has all available privileges and is intended for users that are to function as administrators.



Roles

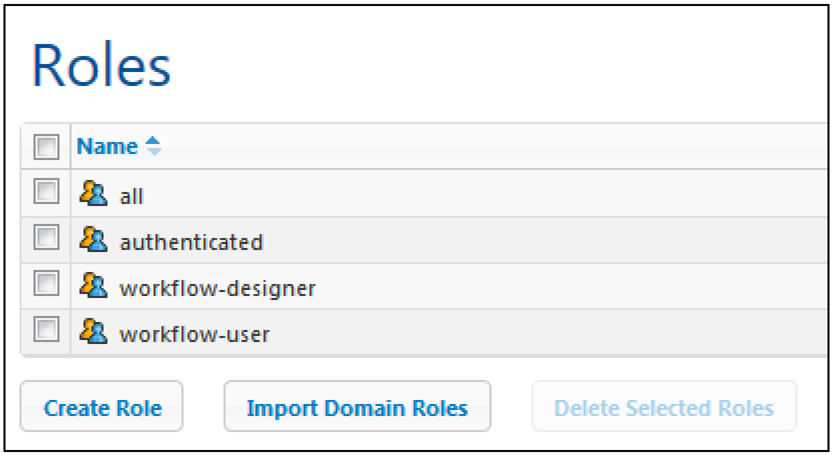
□	Name
□	all
□	authenticated
□	workflow-designer
□	workflow-user

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:

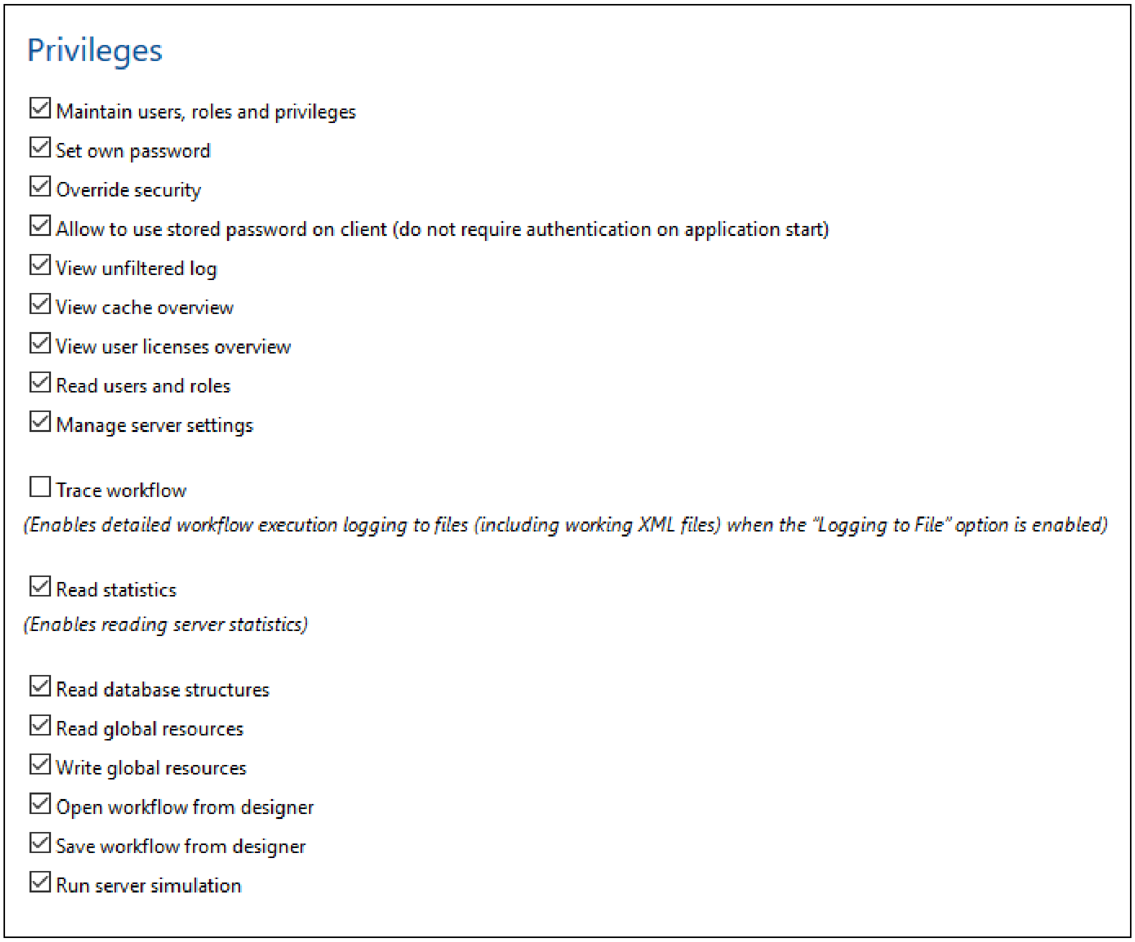
1. In the *Users and Roles* | *Roles* tab, click **Create Role** (*see screenshot below*). This displays the Create Role page.



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 statistics

(Enables reading server statistics)

☑ Read database structures

☑ Read global resources

☑ Write global resources

☑ Open workflow from designer

☑ Save workflow from designer

☑ Run server simulation

4. Click **Save** to finish.

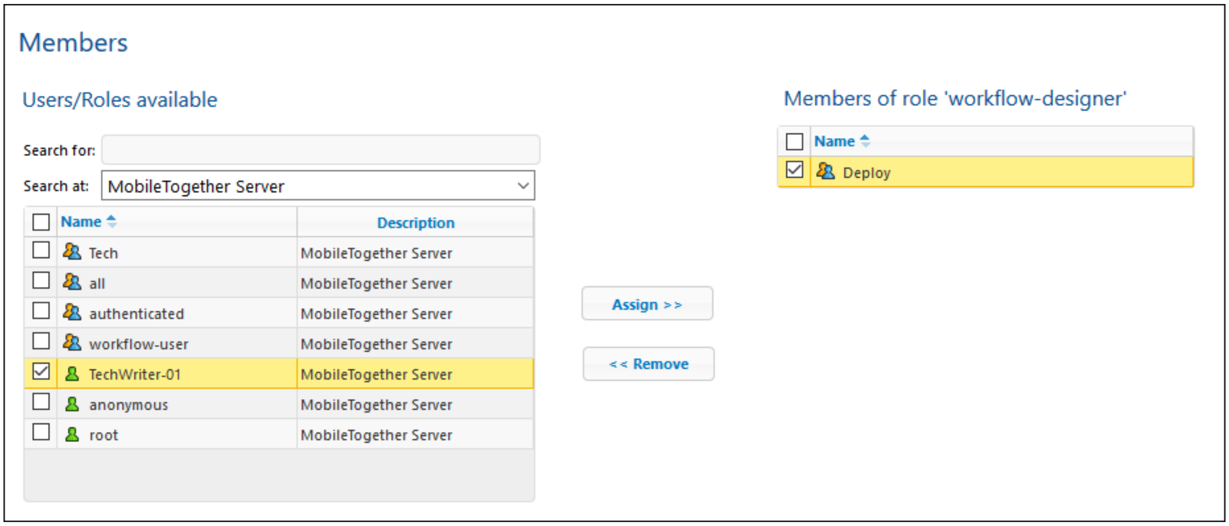
After you have saved the role, you can assign members to it in the Members pane at the bottom of the page (*see next section*). A member can be a user or another role. You can subsequently edit a role's properties by clicking the role's name in the list of roles in the *Users and Roles* | *Roles* tab.

To see a listing of all the privileges of a role, go to the tab, *Users and Roles | Reports | Privileges by User* 100.

▼ 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

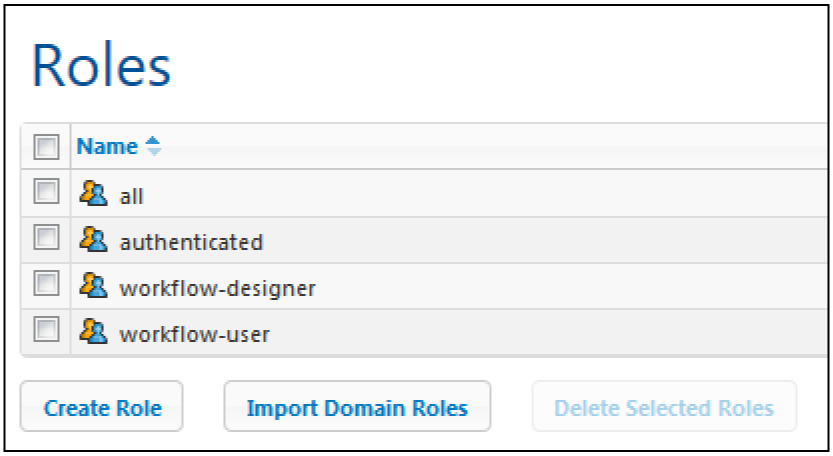
- All available users/roles are listed on the left.
- In the *Search At* combo box, you can select users[89] and roles[94] that have been defined for MobileTogether Server or for all enabled domains (by selecting, respectively, *MobileTogether Server* or *Directory Service* in the combo box). A domain's users and roles are defined by the domain's administrator. They will be available in the pane only if the Active Directory Login setting[119] has been enabled in the Settings tab[110].
- 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*[100].

Importing a domain role

If Active Directory login[119] 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**.



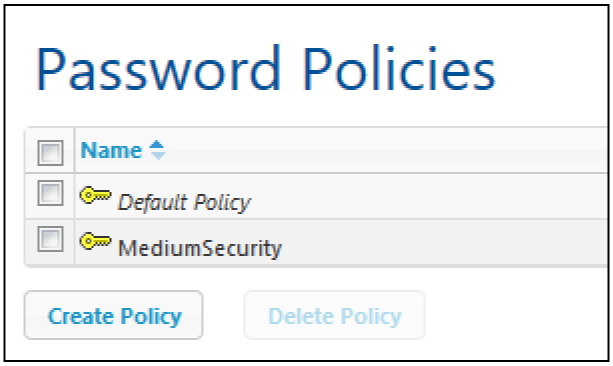
Roles

□	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..

5.2.3 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.



Password Policies

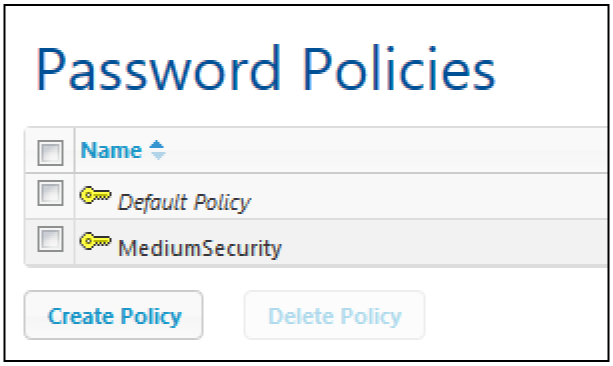
□	Name
□	*Default Policy*
□	MediumSecurity

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:

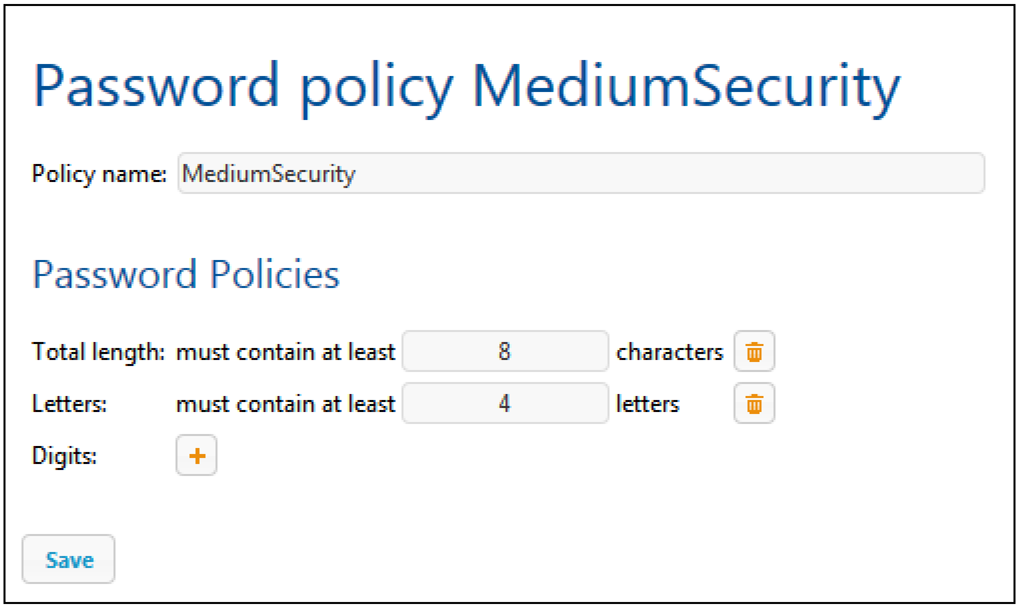
1. In the *Users and Roles* | *Password Policies* tab, click **Create Policy** (*see screenshot below*). This displays the Create Password Policy page.



Password Policies

□	Name
□	*Default Policy*
□	MediumSecurity

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

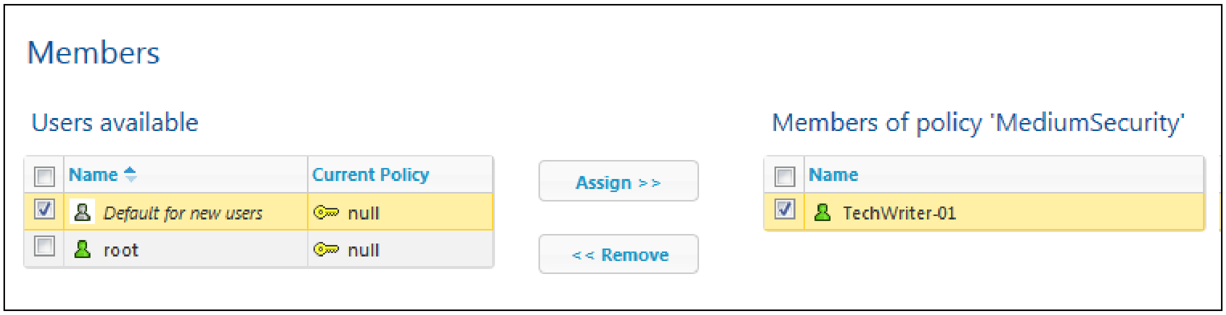
Digits: +

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

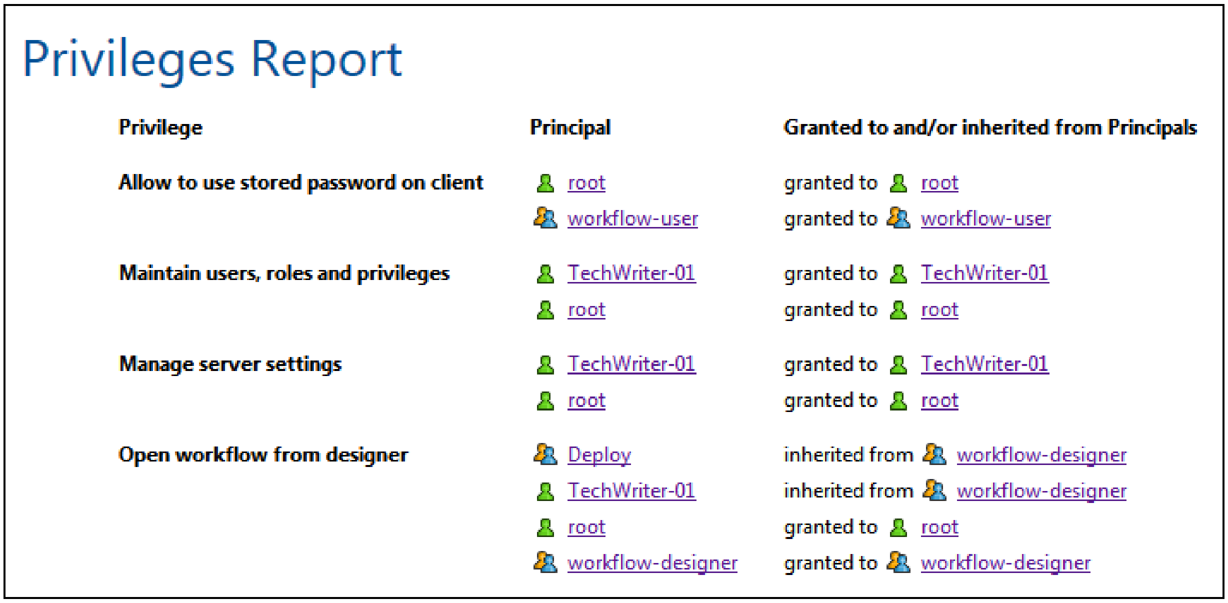
All available users are listed on the left. All users that are members of the policy are listed on the right. Select the user that you want to assign as a member from the list on the left, and click **Assign**. To remove an assigned user, select it in the list on the right, and click **Remove**. The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

5.2.4 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.

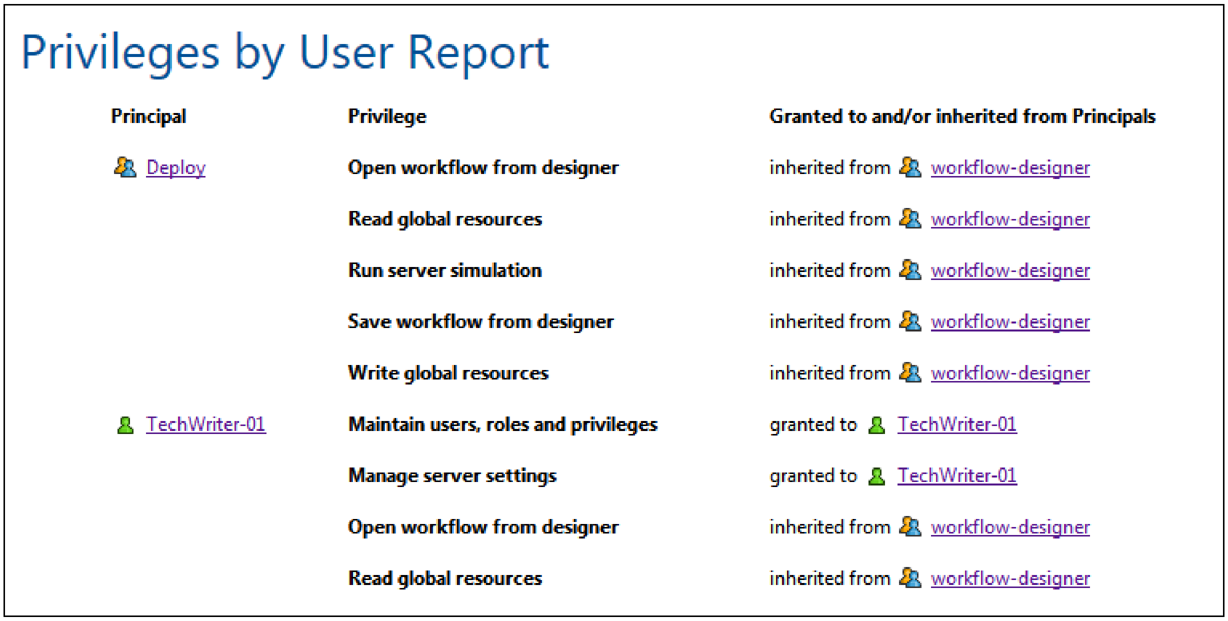


Privileges Report

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.

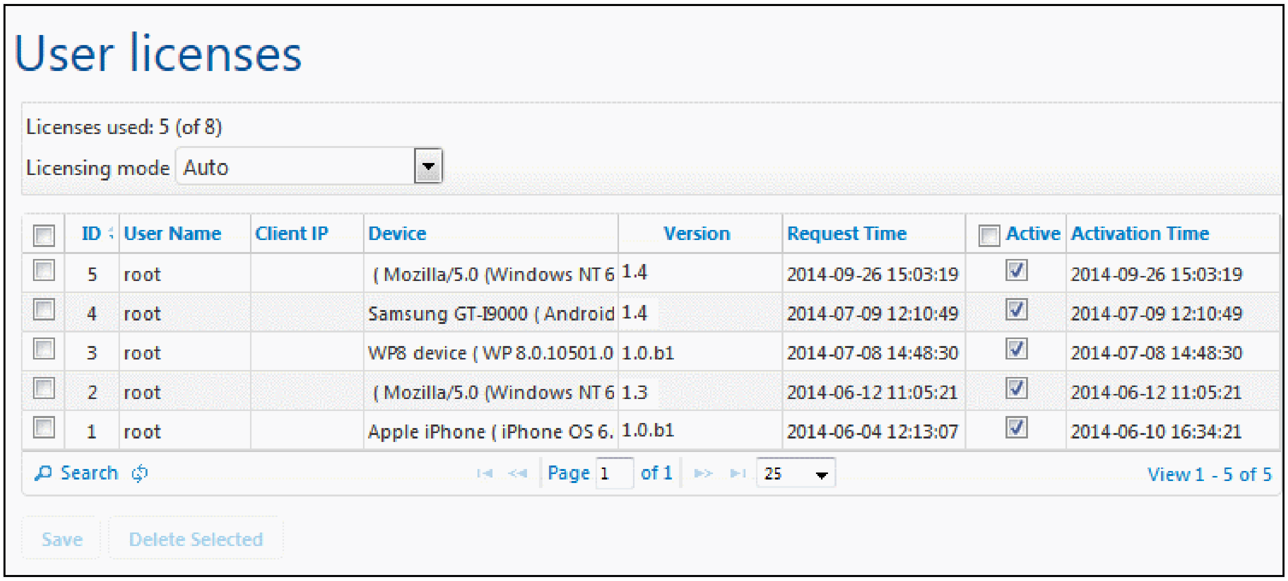


Privileges by User Report

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	Client IP	Device	Version	Request Time	□ Active	Activation Time
□	5	root		(Mozilla/5.0 (Windows NT 6	1.4	2014-09-26 15:03:19	☑	2014-09-26 15:03:19
□	4	root		Samsung GT-I9000 (Android	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	1.3	2014-06-12 11:05:21	☑	2014-06-12 11:05:21
□	1	root		Apple iPhone (iPhone OS 6.	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

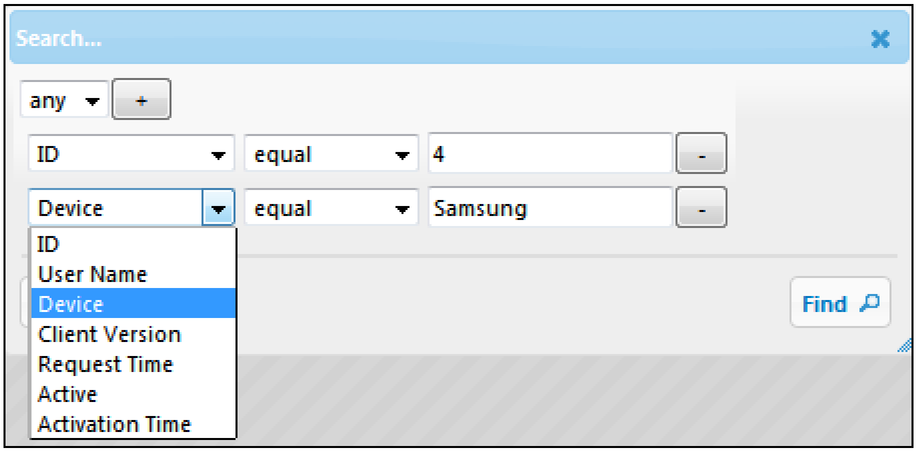
Given below is a description of the fields of the user license tab.

- *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.
- *Client IP:* IP address of the client device.

- *Device:* The mobile device or browser that requested the license.
- *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.
- *Request Time:* The time when the client requested a license.
- *Active:* The *Active* check box is used to activate/deactivate a license. Click **Save** to finalize the change.
- *Activation Time:* The time when a license was activated.

Search

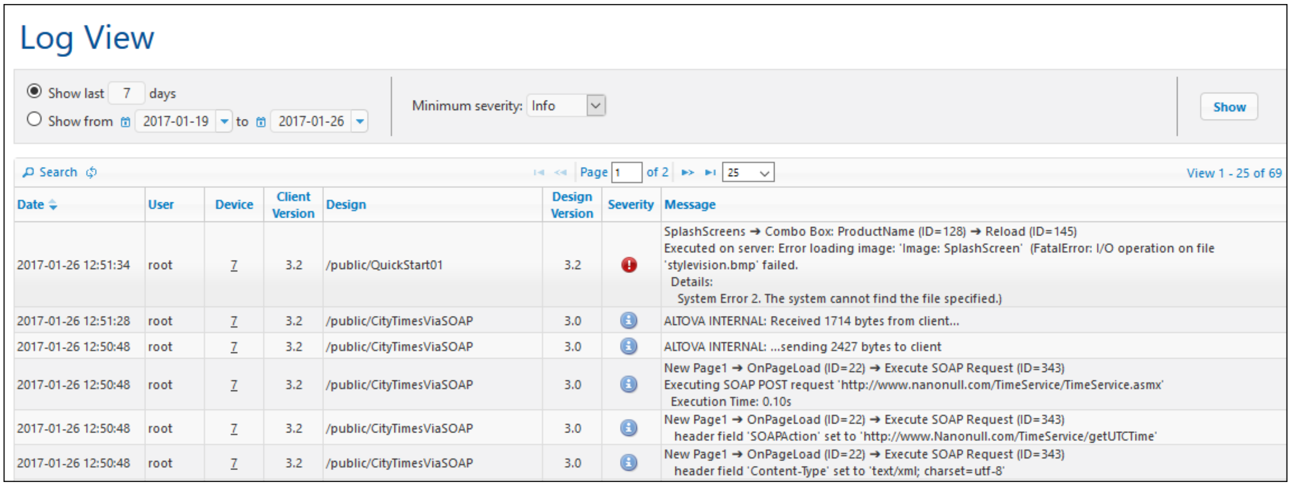
Click the **Search** button to open the Search dialog (*screenshot below*) and search by a combination of one or more user license fields.



- 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—including changes to server settings (who and when). Logs are shown according to the selected filters, which are 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 the Settings | Logging 117 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

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'

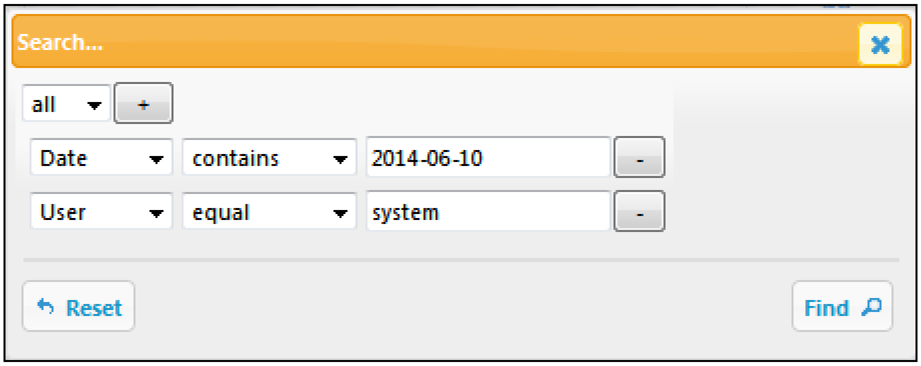
The view can be filtered by:

- *Date:* Ranges or specific dates can be set.
- *Minimum severity:* Error is the highest severity (only errors are shown); Warning 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 bottom 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*).



all

Date contains 2014-06-10

User equal system

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.

Copying log messages for locating errors in the design

If the server logs show an error, you can hover over the error message to display a **Copy** button that enables you to copy the error message to the clipboard. Now if you open the design of the solution in MobileTogether Designer, you can paste the error message in the Messages Pane. The server log message that you copied to the clipboard will be pasted and will contain links that take you to the source of the error in the design.

5.5 Cache

A cache is a data file that is generated from a page source of a design (typically an XML file or a database) at a given time. A cache is defined in MobileTogether Designer, and saved from there to MobileTogether Server. The data in the cache comes from the page source. The frequency and times of cache updates are defined in the properties of the cache.

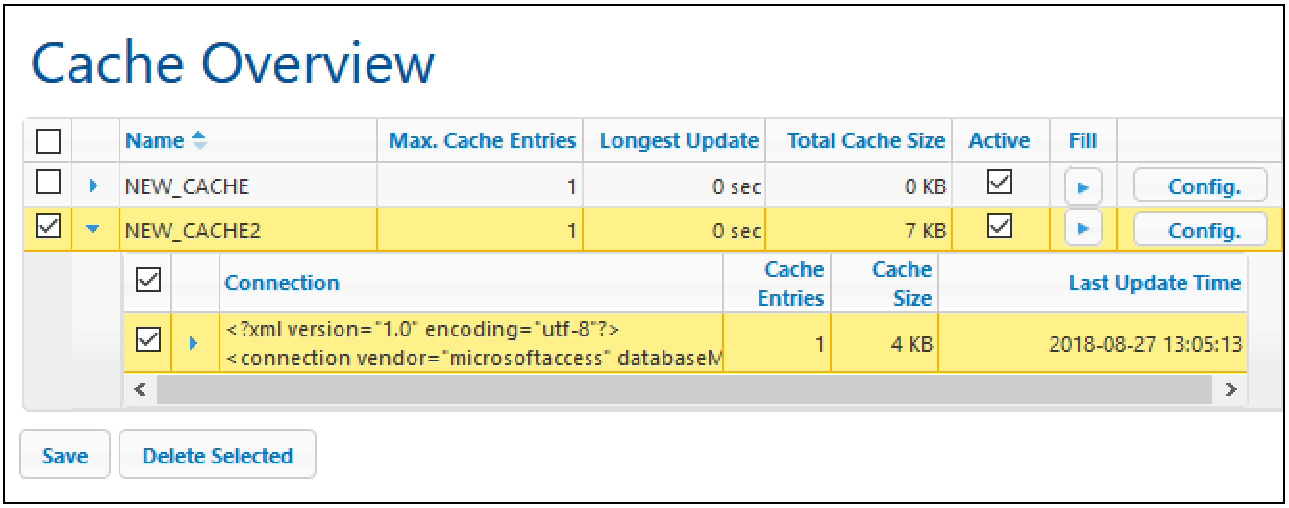
The **Cache** tab displays information about the caches that are currently available on the server, enables you to modify properties of individual caches, and also to activate/deactivate caches and delete caches.

Note: The initial creation of a cache is done in MobileTogether Designer; it cannot be done in MobileTogether Server. See the MobileTogether Designer documentation for information about creating caches.

Note: Server settings for caches (cache directory, log limit, etc) are available in the Settings | Cache 124 tab.

Details displayed and available actions

Caches that are currently available on the server are listed by their names together with information about the cache (*see screenshot below*). You can expand/collapse a cache listing. When expanded, the page sources to which a cache is connected are listed. (Note that a cache can be connected to multiple page sources if its data structure is compatible with that of other page sources.)



Cache Overview

□		Name	Max. Cache Entries	Longest Update	Total Cache Size	Active	Fill	
□	▸	NEW_CACHE	1	0 sec	0 KB	☑	▸	Config.
☑	▾	NEW_CACHE2	1	0 sec	7 KB	☑	▸	Config.

☑		Connection	Cache Entries	Cache Size	Last Update Time
☑	▸	<?xml version="1.0" encoding="utf-8"?> <connection vendor="microsoftaccess" databaseM	1	4 KB	2018-08-27 13:05:13

Details displayed
The following cache information is displayed:

- *Name:* Names are given at the time a cache is defined in MobileTogether Designer and cannot be changed in MobileTogether Server.
- *Max. Cache Entries:* If the cache contains data from a DB page source that is filtered using query parameters, then multiple entries for the cache can be saved simultaneously, up to the maximum number specified here. The number displayed here specifies how many cache entries will be stored before the first cache entry is deleted and the latest cache entry is appended. It is only for this kind of page source that the maximum value is greater than one. This number can be edited in the screen that appears when you click the cache's **Config** button (*see screenshot above*).

- *Longest Update:* Each cache can be updated multiple times. This column displays the time taken for the longest update.
- *Total Cache Size:* The cache size (for all cache entries) that is allocated to the cache on disk (or other medium). Cache size is allocated automatically.
- *Active:* Shows whether the cache is currently active or not.

Available actions
The following actions can be carried out:

- *Activate/deactivate a cache:* Check/uncheck the box in the Active column to activate/deactivate, respectively. 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.
- *Delete one or more caches:* Select the caches you want to delete 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.
- *Modify maximum cache entries:* Click **Config** and modify the number in the screen that appears, then click **Save**. Note that this option is available only for page sources that have been filtered using query parameters.
- *Modify cache update frequency and update times:* Click **Config** and modify the update frequency, then click **Save**.
- *View logs of cache entries:* Expand a cache entry to see its log.
- *Fill a cache:* Click the **Go** button in the *Fill* column of a cache to fill the cache manually.

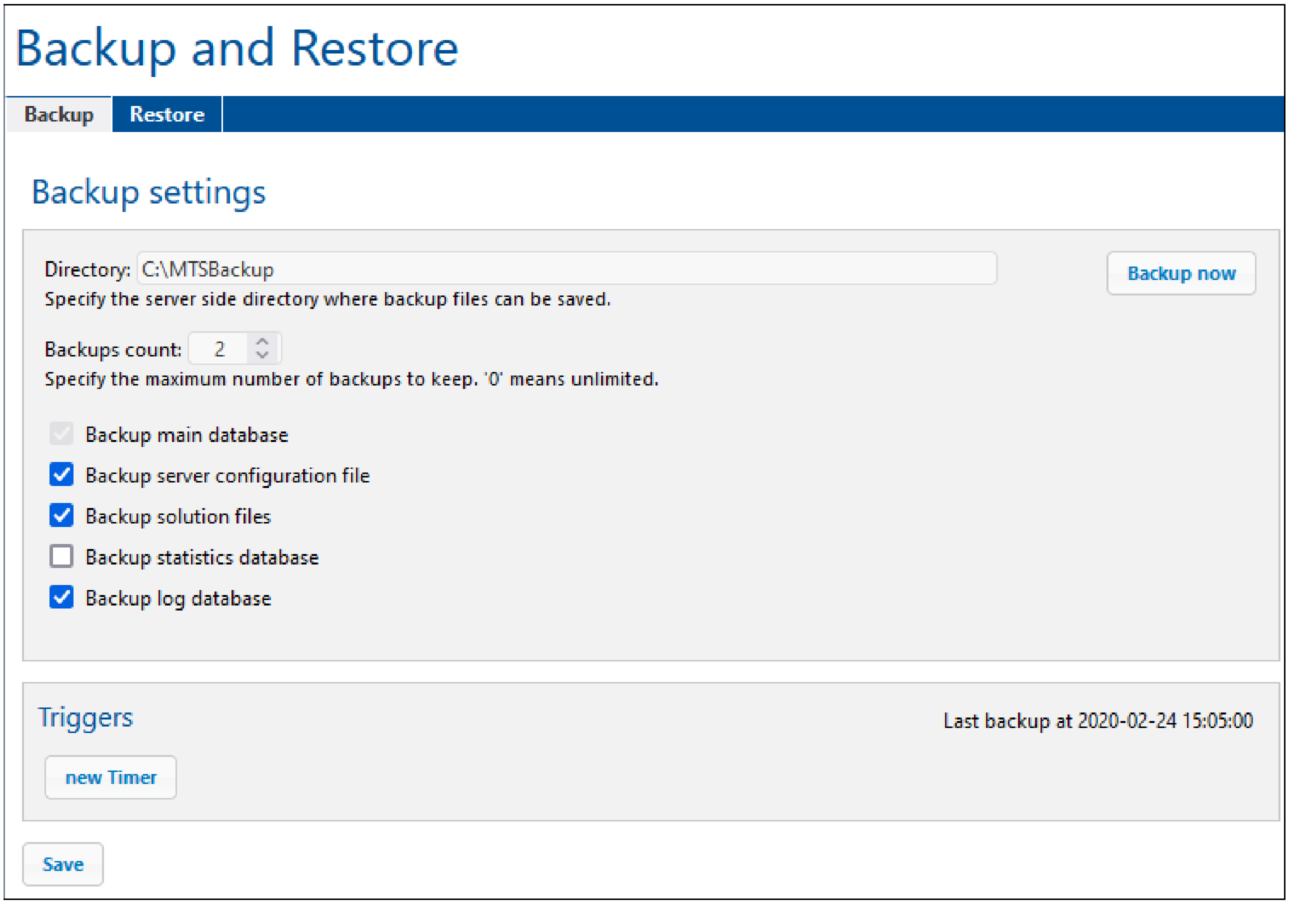
5.6 Backup and Restore

The **Backup and Restore** tab contains two sub-tabs, *Backup* and *Restore*, which provide settings and controls that enable you to back up and restore the following MobileTogether Server files: (i) the server database file (always backed up), (ii) the server configuration file, (iii) solution files, (iv) the statistics database, and (v) the server log database. You can set time triggers for regular backups, and you can back up immediately. Each backup is saved in a separate folder, which is named by the date and time of the backup

You can restore any or all of the backed up files from any backup (folder) at any time.

Backup settings

The *Backup* tab (*screenshot below*) provide settings and controls to configure and execute backups. After you have configured the settings, click **Save** before carrying out a backup.



Directory: C:\MTSBackup
Specify the server side directory where backup files can be saved.

Backups count: 2
Specify the maximum number of backups to keep. '0' means unlimited.

☑ Backup main database
☑ Backup server configuration file
☑ Backup solution files
☐ Backup statistics database
☑ Backup log database

Last backup at 2020-02-24 15:05:00

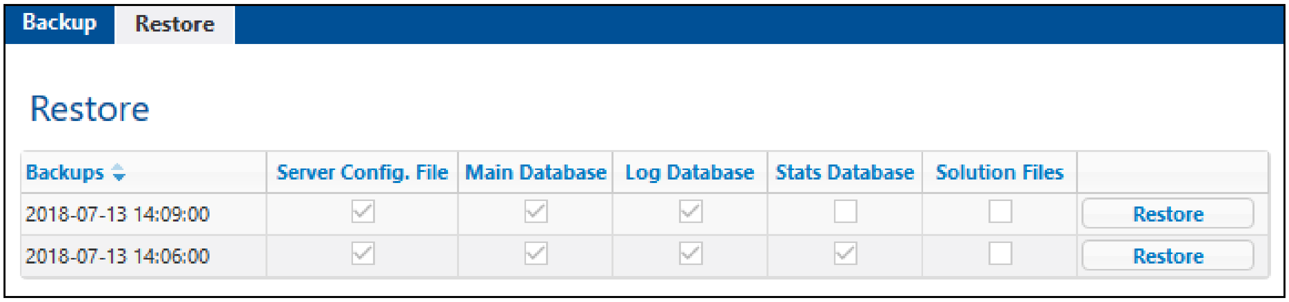
The following backup settings can be configured:

- The folder on the server that will contain the backed up files. (For Linux, you can set, for example, `/var/opt/Altova/MobileTogetherServer/MTSBackup` or `/tmp/MTSBackup`. For macOS, you can set `/var/Altova/MobileTogetherServer/MTSBackup` or `/tmp/MTSBackup`.

- The number of backups to keep. After this number has been reached, the oldest backup will be deleted. To keep an unlimited number of backups, select `0`. Also check the sizes of backup folders to help determine an optimal number of backups to keep.
- The server database file (`mobiletogether.db`) is always backed up. To back up additional files, check its type. In the screenshot above, for example, the server configuration file is additionally backed up. Note that when you restore a backup, all the files in that backup will be restored.
- You can set a Timer trigger for one or more backups. To do this, click **New Timer**, and set the time for your backup/s. You can temporarily disable a trigger, as well as create multiple triggers.
- After configuring a backup, click **Save** to save the settings.
- To back up immediately with the currently saved settings, click **Backup now**.

Restore

The *Restore* tab (*screenshot below*) displays the currently saved backups. For each backup, the files that were backed up are shown with a check mark (*see screenshot below*).



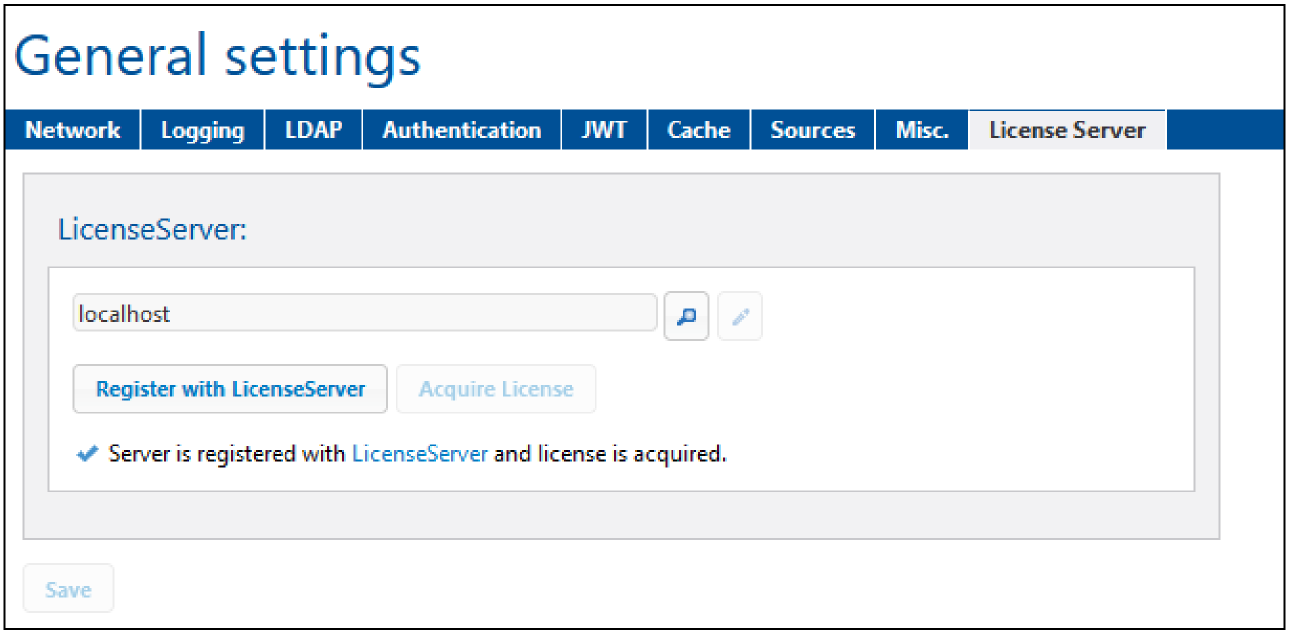
Restore

Backups	Server Config. File	Main Database	Log Database	Stats Database	Solution Files	
2018-07-13 14:09:00	☑	☑	☑	□	□	Restore
2018-07-13 14:06:00	☑	☑	☑	☑	□	Restore

To restore a particular backup, click that backup's **Restore** button. A dialog appears in which you can select the files you want to restore. Click **Restore** in this dialog to restore the selected files. Other files on the server will not be touched.

5.7 Settings

The **Settings** tab enables you to configure various aspects of the way MobileTogether Server functions. The settings are organized into a number of tabs; the **Cache** tab, which contains cache settings is shown in the screenshot below. If you wish to modify a setting, go to its tab, and modify the setting as required. Click the **Save** button at the bottom of the tab for the modified setting to take effect.



LicenseServer:

localhost

✓ Server is registered with LicenseServer and license is acquired.

The subsections of this section describe the settings in each tab:

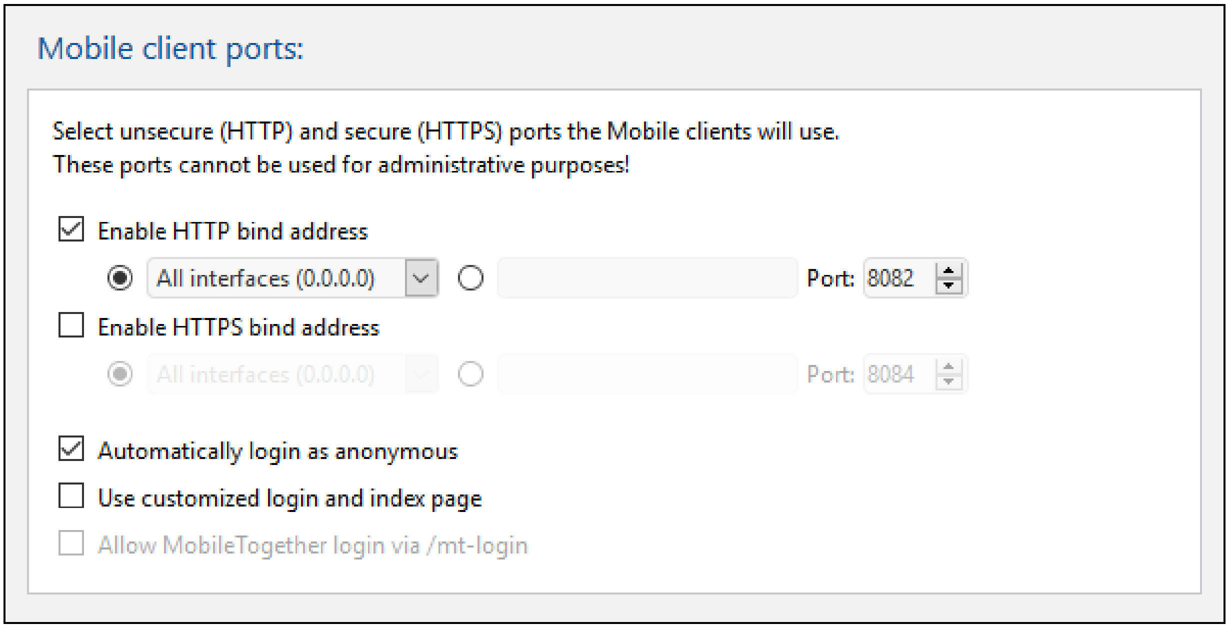
- Network[110]: Settings for mobile client ports, administrator ports, and SSL certificates
- Logging[117]: Logging settings
- LDAP[119]: Settings for Directory service login
- Authentication[120]: Settings to enable authentication of users coming from another MobileTogether Server; saves user a second MobileTogether Server login
- JWT[123]: JSON Web Token (JWT) authentication settings
- Cache[124]: cache settings
- Misc[126]: Settings for server statistics, server simulations, workflow execution on server, session timeouts, backups on upgrade, and email sending,
- Sources[124]: Server-side data folder location, and management of server-side database connections
- LicenseServer[130]: Registration and licensing with Altova LicenseServer
- Non-WebUI Settings[130]: Other settings that can be edited in the MobileTogether Server configuration file.

5.7.1 Network

The **Network** tab enables you to configure network settings that define how the server can be accessed (i) by client devices, and (ii) by administrators. Client access enables devices to connect to the server and use solutions that have been deployed to the server, Administrator access enables the server to be configured and managed. If you modify any setting, click **Save** at the bottom of the tab for the modified setting to take effect.

▼ 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 47. You can specify 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 field of the second radio button. If you are using a dual-stack server running both IPv4 and IPv6, use a double colon **`::`** as the bind address; this allows both protocols on all network interfaces. Only ports with numbers from **`1`** to **`65535`** may be used.



Mobile client ports:

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

☑ Enable HTTP bind address

● All interfaces (0.0.0.0) ○ Port: 8082

☐ Enable HTTPS bind address

● All interfaces (0.0.0.0) ○ 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 89. 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 89 for anonymous 89.

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	`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*

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Cache-Control" content="no-store"/>
    <title>Customized Login</title>
  </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>
          <!-- The user to login -->
          <tr>
            <td>
              <label for="username">Username:</label>
            </td>
            <td>
              <input type="text" name="username" id="username" size="30"/>
            </td>
          </tr>
          <!-- The password of the user -->
          <tr>
            <td>
              <label for="password">Password:</label>
            </td>
            <td>
              <input type="password" name="password" id="password" size="30"/>
            </td>
          </tr>
        </table>
        <!-- The Active Directory domain details -->
        <h2>Active Directory Login:</h2>
        <table>
          <tr>
            <td>
              <label for="providernameprefix">Domain prefix:</label>
            </td>
            <td>
              <input type="text" name="providernameprefix"
id="providernameprefix" value=""/>
```

```
            </td>
          </tr>
          <tr>
            <td>
              <label for="providernamesuffix">Domain suffix:</label>
            </td>
            <td>
              <input type="text" name="providernamesuffix"
id="providernamesuffix" value=""/>
            </td>
          </tr>
        </table>
        <!-- The Sign-In button -->
        <input type="submit" value="Sign in"/>
        <!-- The page to redirect to after a successful login. -->
        <input type="hidden" name="from_page" value="/index"/>
      </form>
    </div>
  </body>
</html>
```

⊟ *index.html*

```
<html>
  <head>
    <meta http-equiv="Cache-Control" content="no-store" />
    <title>Custom Index</title>
  </head>
  <body>
    <img alt="Logo" src="/index/static/logo.png"></img>
    <hr/>
    <p><a href="/do_logout">Logout</a></p>
    <h1>MobileTogether Custom Login</h1>
    <p><a href='/run?d=/public/About'>Start the About app</a></p>
    <p><a href='/run?d=/public/DateCalc'>Start the Date Calculator app</a></p>
    <p><a href='/run?d=/public/WorldPopulation'>Start the World Population
Statistics app</a></p>
  </body>
</html>
```

Note: If the user is a domain user, the login credentials will have a form something like this: `domainPrefix@domainSuffix`. For example: If the domain user is `someUserName@somedomain.altova.com`, the domain prefix is `someUserName`, and the domain suffix is `@somedomain.altova.com`.

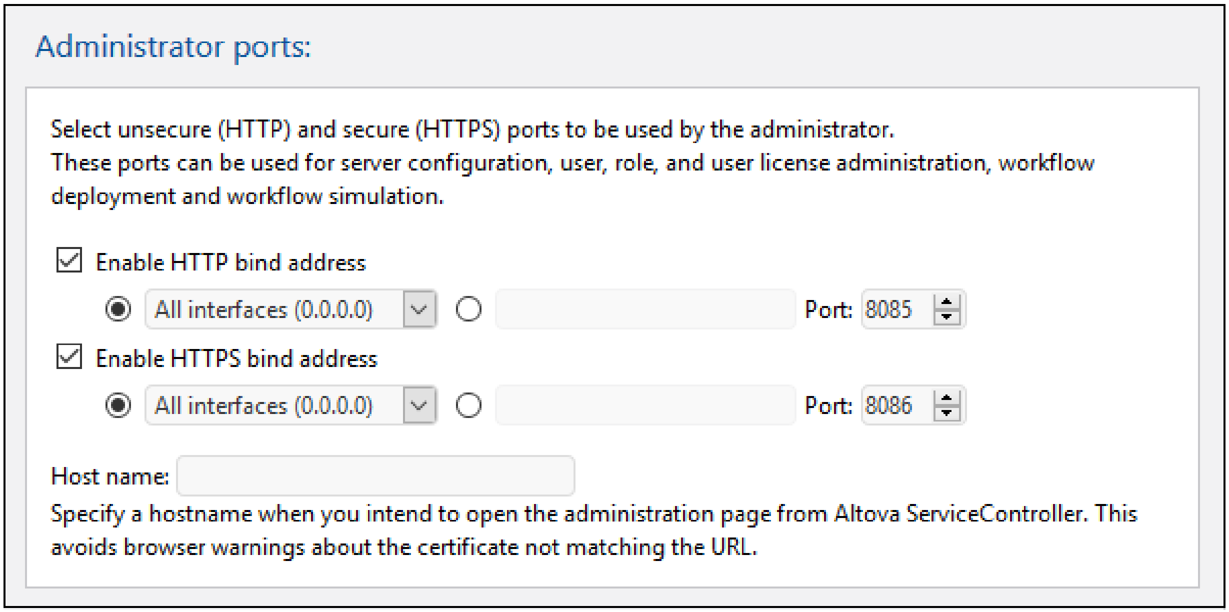
Allow MobileTogether login via /mt-login
This option specifies that the login will be via the default login page and first page—and not via the

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 [87].
- 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.

☑ Enable HTTP bind address

● All interfaces (0.0.0.0) ○ Port: 8085

☑ Enable HTTPS bind address

● All interfaces (0.0.0.0) ○ Port: 8086

Host name:

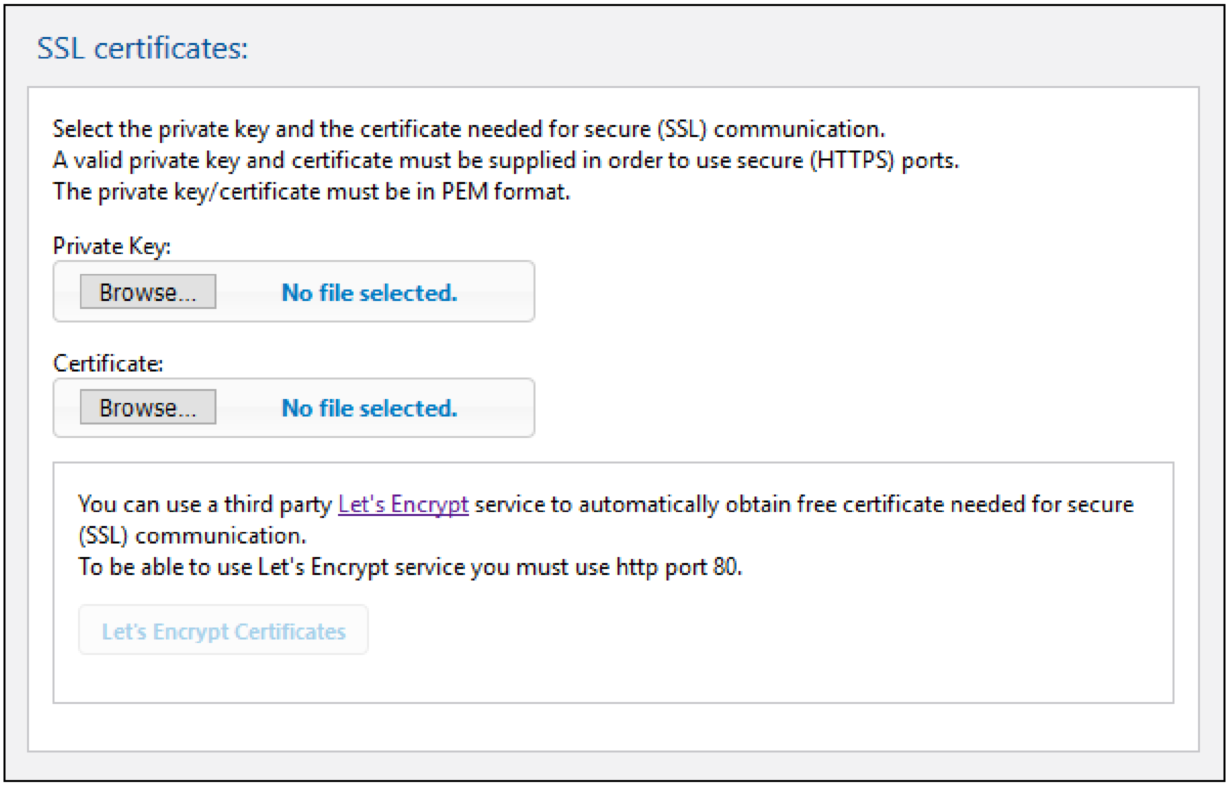
Specify a hostname when you intend to open the administration page from Altova ServiceController. This avoids browser warnings about the certificate not matching the URL.

The `HTTP` port is the unsecure port; `HTTPS` is the secure port. To use `HTTPS`, you need to set up SSL Encryption [47]. 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.

You can specify 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 field of the second radio button. If you are using a dual-stack server running both IPv4 and IPv6, use a double colon `::` as the bind address; this allows both protocols on all network interfaces. Only ports with numbers from **`1`** to **`65535`** may be used.

▾ 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 [47] 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:

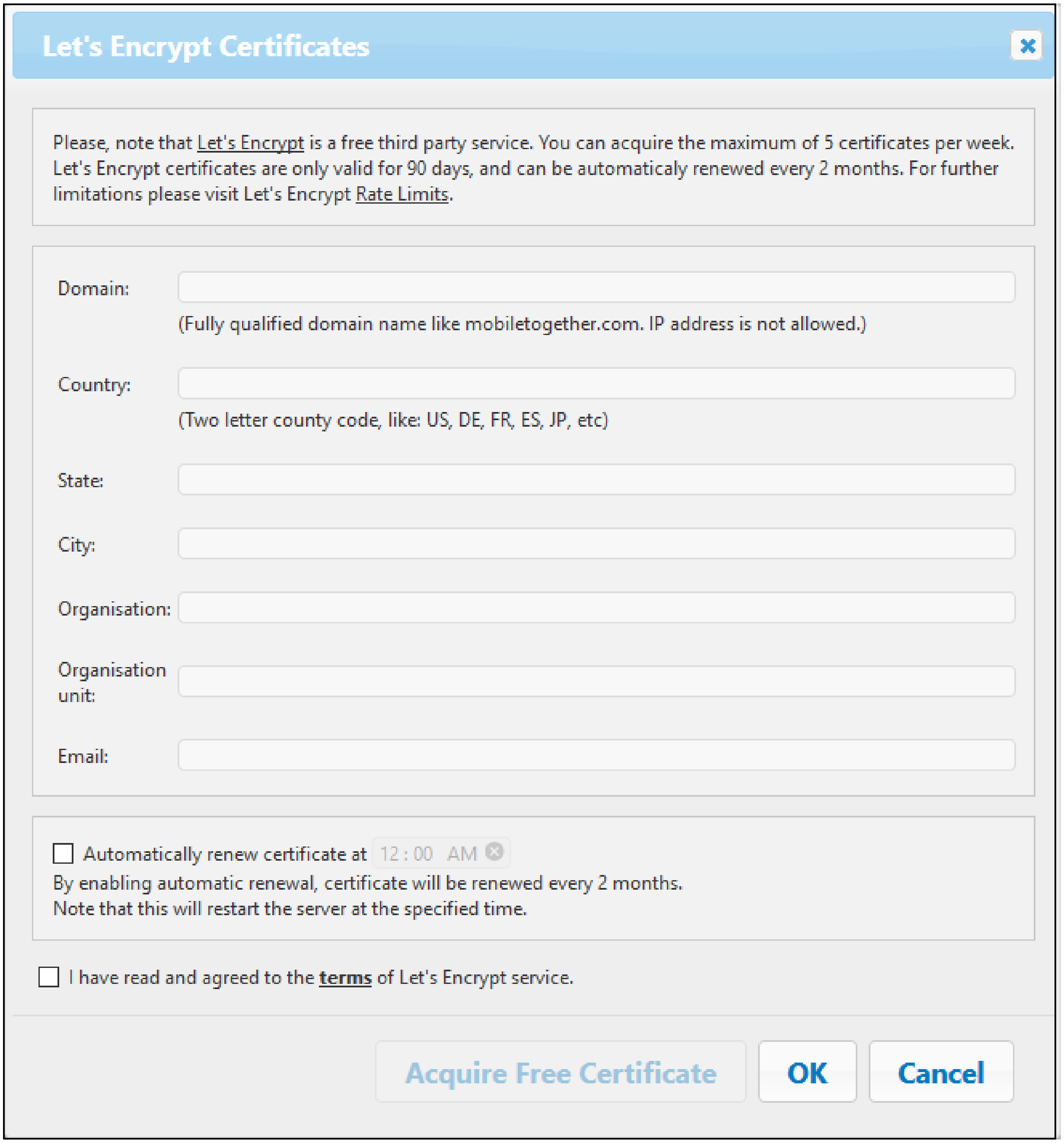
Certificate:

You can use a third party Let's Encrypt service to automatically obtain free certificate needed for secure (SSL) communication.
To be able to use Let's Encrypt service you must use http port 80.

To use the free certificates of the **Let's Encrypt** service, the following requirements must be met:

- MobileTogether Server must be visible from the outside on port **80**. This is because Let's Encrypt will send a request to your domain in order to validate your identity.
- Use of a valid domain name, like altova.com, is needed for registration. IP addresses are not allowed.
- Your domain host must support Certification Authority Authorization (CAA) records.

After you set the HTTP client port to **80**, the **Let's Encrypt Certificates** button *(see screenshot above)* will be enabled. Click it to open the Let's Encrypt Certificates dialog *(screenshot below)*, in which you enter details for obtaining a Let's Encrypt certificate.



Please, note that Let's Encrypt is a free third party service. You can acquire the maximum of 5 certificates per week. Let's Encrypt certificates are only valid for 90 days, and can be automaticaly renewed every 2 months. For further limitations please visit Let's Encrypt Rate Limits.

Domain:
(Fully qualified domain name like mobiletogether.com. IP address is not allowed.)

Country:
(Two letter county code, like: US, DE, FR, ES, JP, etc)

State:

City:

Organisation:

Organisation unit:

Email:

□ Automatically renew certificate at 12 : 00 AM
By enabling automatic renewal, certificate will be renewed every 2 months.
Note that this will restart the server at the specified time.

□ I have read and agreed to the **terms** of Let's Encrypt service.

In the dialog, enter the required details and check the *Agreement of Terms* statement. Let's Encrypt certificates are valid for 90 days, so MobileTogether Server offers the option of automatically renewing the certificate every two months (on the same date as the date you acquire the certificate). Check this option if you want to avail of it. Since the server will be unavailable for a few seconds while a certificate renewal is implemented, you can also select the time of renewal on the renewal date. After you have completed the dialog, click **OK**.

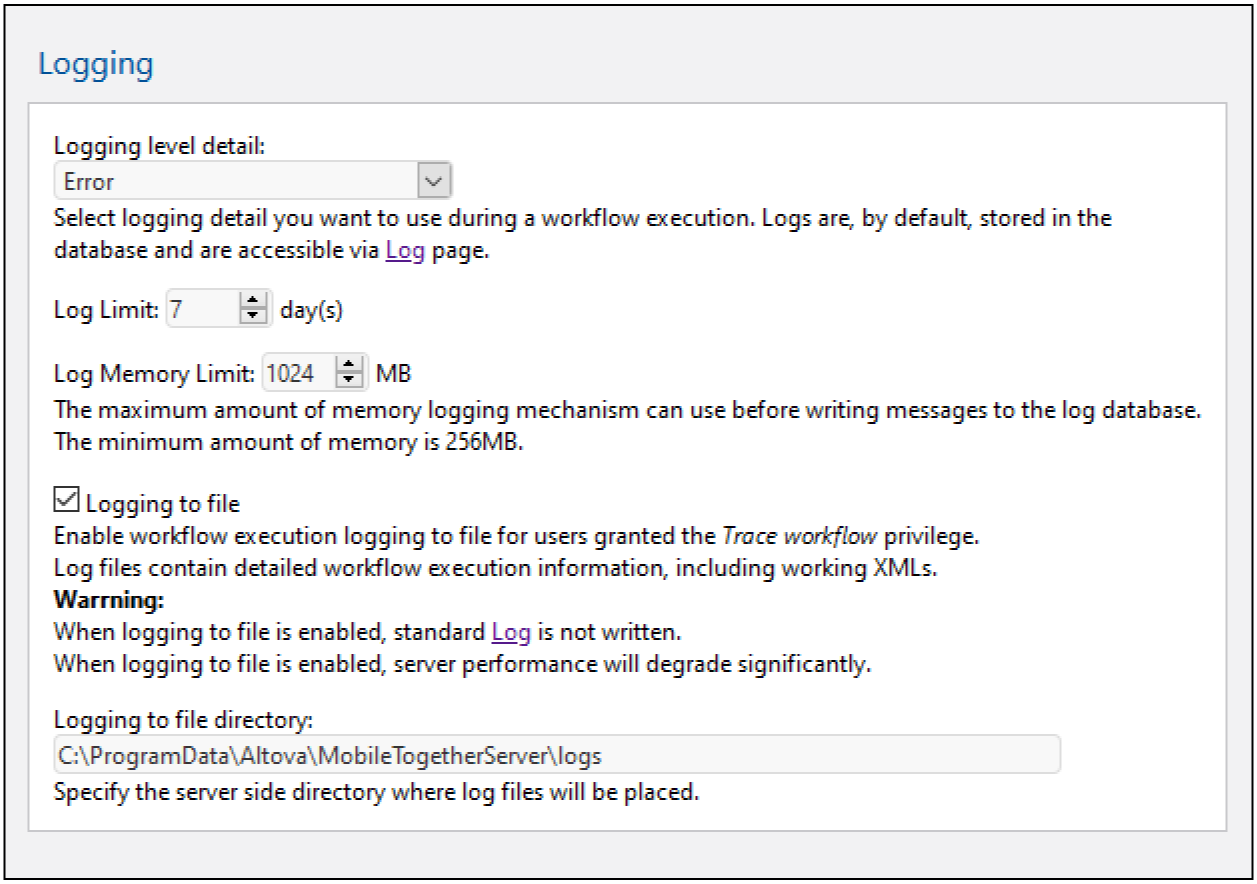
Click **Save** at the bottom of the *Network* tab for the settings to be saved and take effect.

5.7.2 Logging

The **Logging** tab provides settings for the logging features of MobileTogether Server. If you modify any setting, click **Save** at the bottom of the tab for the modified setting to take effect.

Logging

Logs contain reports of workflow activity, and they are displayed in the Log 104 tab of the Web UI. The settings in the Logging 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.
Warrning:
When logging to file is enabled, standard Log is not written.
When logging to file is enabled, server performance will degrade significantly.

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

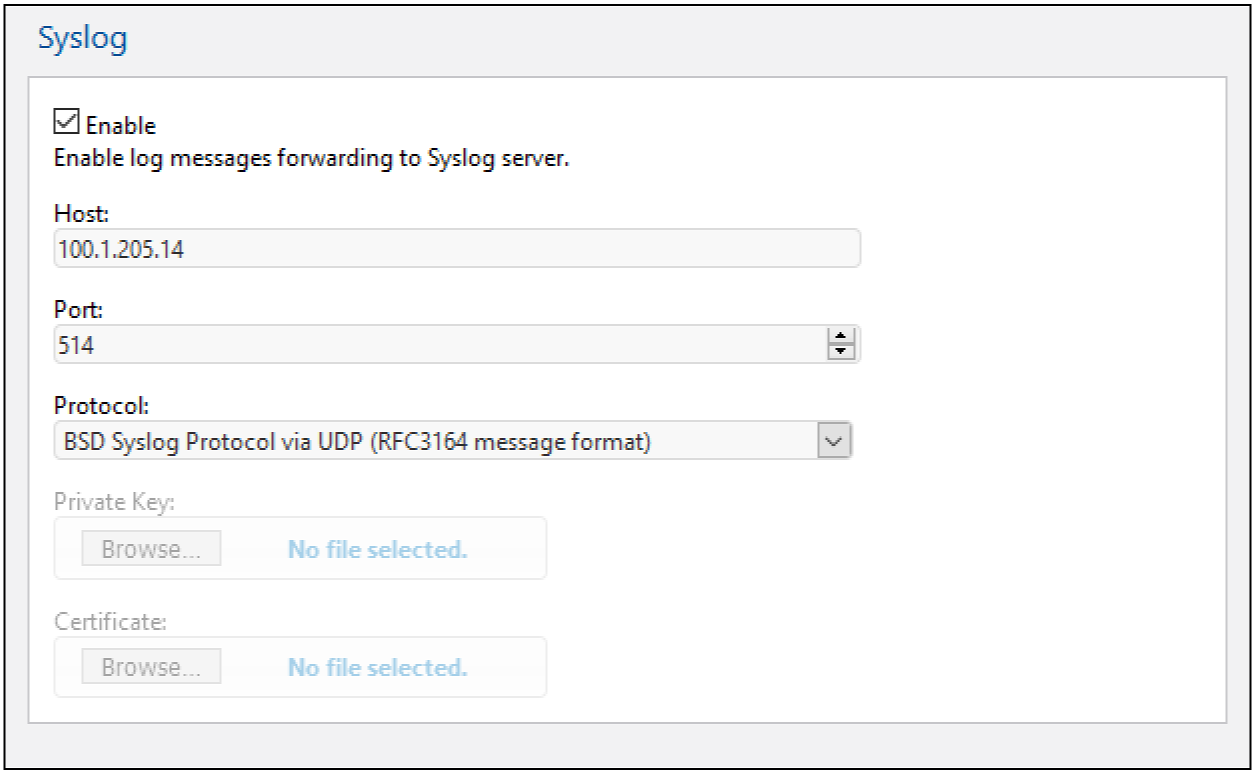
of the limit value will be: (i) the amount of memory on the machine, and (ii) the logging level detail. The lowest Log Memory Limit value you are allowed to enter is `256 MB`.

- *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.

Syslog

Syslog is a standard protocol used to forward system log or event messages to a specific server, called a syslog server. Developers often use a syslog server to collect logs from various machines in a central location for further analysis. Syslog is the default logging mechanism in Linux. On Windows, a syslog server will need to be installed. (Note that Windows Event Log cannot be use for this purpose.)

The Syslog settings pane *(screenshot below)* enables you to define the settings for the syslog server.



Syslog

☑ Enable
Enable log messages forwarding to Syslog server.

Host:
100.1.205.14

Port:
514

Protocol:
BSD Syslog Protocol via UDP (RFC3164 message format)

Private Key:
Browse... No file selected.

Certificate:
Browse... No file selected.

- The *Host* and *Port* settings are those of the Syslog server. Also see *Notes* below.
- *Protocol:* You can select from among the following protocols: BSD (UDP or TCP) or IETF (TCP or TLS/SSL). The BSD protocol provides the non-blocking UDP communication (which is, however, not a required communication property, because MobileTogether Server does not use any blocking). The IETF syslog protocol is recommended because it sends more information and supports secure communication. Additional information (about user, solution, and client-version) can be accessed in syslog server via macros: `${.SDATA.meta.user}`, `${.SDATA.meta.session}`, and `${.SDATA.meta.version}`.

Notes

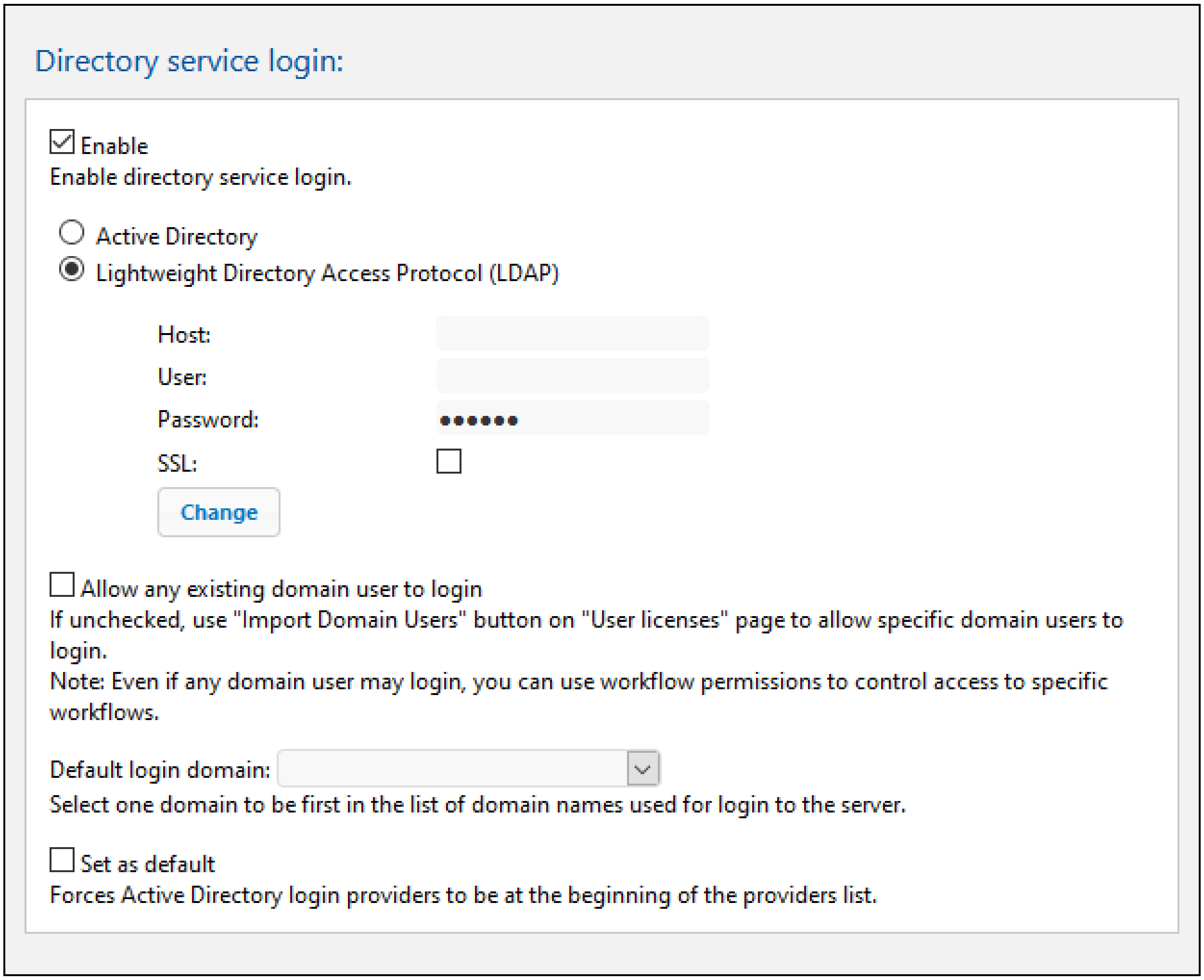
Linux usually comes with **`rsyslog`** pre-installed. Note, however, that **`syslog-ng`** is a more powerful option. The default ports (if not configured differently) are:

- BSD: 514 (for UDP and TCP)
- IETF: 601 via TCP
- IETF: 6514 via TLS

There are some additional syslog server settings that can be made via the config file 130. These settings are listed in the Config File Settings 130.

5.7.3 LDAP

If **Directory Service Login** is enabled, users can log in to the server with their domain-specific user names and passwords (*see screenshot below*). After enabling Directory Service Login, you can choose whether to use Active Directory Login or Lightweight Directory Access Protocol (LDAP) for login via directory services. *Active Directory* login is used by Microsoft Active Directory. The *LDAP* option can be used with any other directory service provider that supports LDAP.



Directory service login:

☑ Enable

Enable directory service login.

○ Active Directory

● Lightweight Directory Access Protocol (LDAP)

Host:	
User:	
Password:	••••••
SSL:	□

Change

□ 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.

Default login domain:

Select one domain to be first in the list of 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.

LDAP settings
Click **Change** to configure and edit your LDAP settings.

- Enter the name or IP address of the host (which is the machine hosting MobileTogether Server), and the user name and password.
- User names must be in the form of a User-Principal-Name (UPN) or a Distinguished Name (DN). UPNs work only for Active Directory. For other LDAP servers, you must use a Distinguished Name.
- If you want to use a secure connection to the LDAP server (if LDAP server supports it), select the *SSL* check box. For information about using certificates, see Set Up SSL Encryption 47 and Network Settings 110. (Note: On Windows, SSL errors are reported in **Windows Event Viewer | Windows Logs | System**, where *Source* = *Schannel*.)

Note: If the wrong password is entered four times, then Active Directory Login will lock you out. If this happens, call your IT department to unlock your account.

Directory service login settings
If *Allow any existing domain user to login* is checked, then all existing domain users can log in. If not, you can specify which domain users may log in by using the Import Domain Users feature 89. Then go to the Users and Roles | Users 89 tab to import specific users as MobileTogether Server users. An allowed domain user can then be assigned roles or privileges in the usual way 87. After these settings have been made, the allowed users can use their domain-specific login information to log in to MobileTogether Server.

- *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 Domain Users** button of the Users 89 tab.
- *Default login domain:* From the available domains, select the domain that will be listed first. This domain will be the default domain.
- *Set as default:* If set, then Directory Service Login providers are listed at the beginning of the dropdown list of domains.

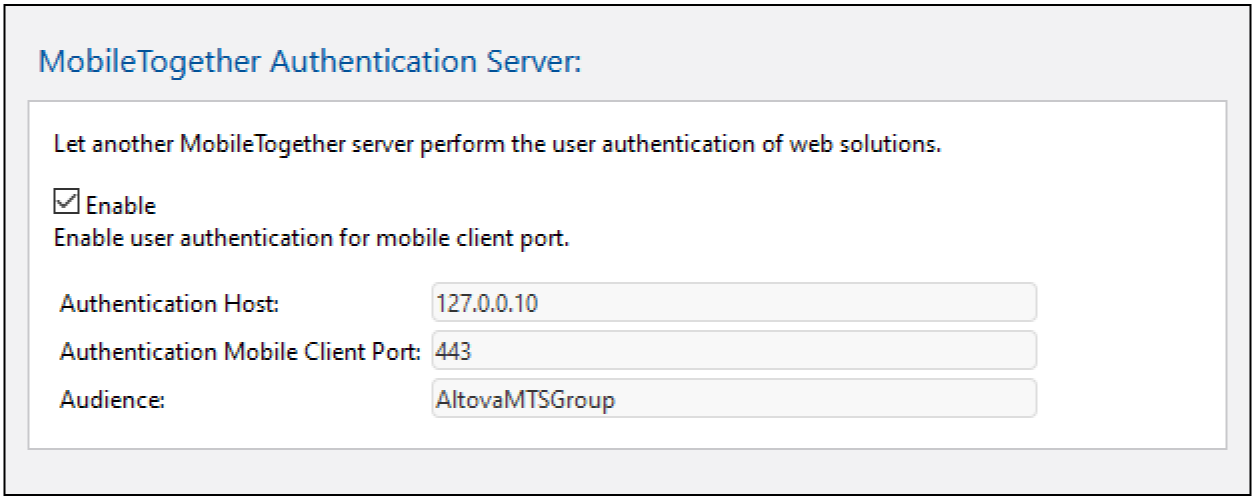
Click **Save** at the bottom of the Settings pane to make the new settings take effect.

5.7.4 Authentication

The Authentication settings enable a user who comes from a solution on another MobileTogether Server and who has been authenticated on that server (the Authentication Server) to start a solution on this MobileTogether Server (the Solution Server) without needing to go through a login on a second MobileTogether Server. There are two settings on this page: (i) Properties of the Authentication Server; (ii) Properties of the simulated Solution Server.

Authentication Server properties

If you want to allow authentication to be securely carried over from one MobileTogether Server to the current one, select *Enable*, and enter the properties of the Authentication Server. This is the server from which the calling solution will access the current server.



MobileTogether Authentication Server:

Let another MobileTogether server perform the user authentication of web solutions.

☑ Enable
Enable user authentication for mobile client port.

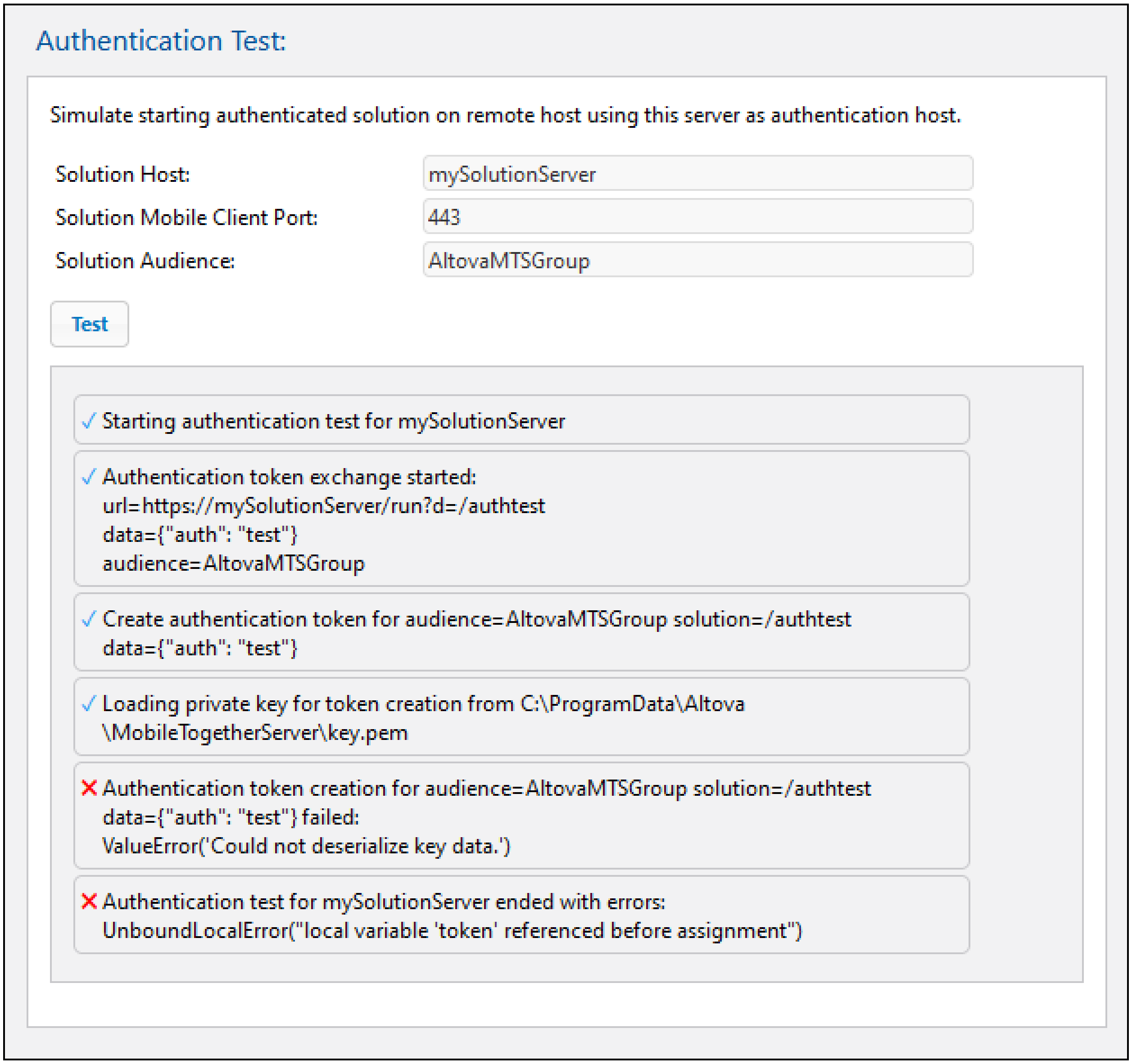
Authentication Host: 127.0.0.10
Authentication Mobile Client Port: 443
Audience: AltovaMTSGroup

Properties of the Authentication Server:

- *Authentication Host:* This is the IP address of the machine hosting the MobileTogether Server where the authentication has been carried out.
- *Authentication Mobile Client Port:* This is the port through which client devices connect to the server. Note that**:** (i) both servers must use SSL encryption 47 (HTTPS connections), and (ii) that both solutions (the calling and called) must be run for anonymous users. *Also see Network Settings* 110.
- *Audience:* This is a string that defines the Audience of the solutions for which authentication is allowed. It must be the same as the Audience string that is specified in the Solution Execution action of the calling solution. By ensuring that the Audience strings match, you prepare the current server to receive the calling solution as authenticated.

Authentication Test settings

The Authentication Test settings enable you to define the properties of a remote server you want to test for authenticated communication with your current server. The current server plays the role of Authentication Server, and the communication between it and the remote server simulates authenticated communication. After you enter the properties of the remote server, click **Test** (*screenshot below*). During communication, log messages of both servers will be displayed in the messages pane below the **Test** button (*see screenshot*). You can now examine these to debug communication or authentication issues.



Authentication Test:

Simulate starting authenticated solution on remote host using this server as authentication host.

Solution Host: mySolutionServer

Solution Mobile Client Port: 443

Solution Audience: AltovaMTSGroup

✓ Starting authentication test for mySolutionServer

✓ Authentication token exchange started:
url=https://mySolutionServer/run?d=/authtest
data={"auth": "test"}
audience=AltovaMTSGroup

✓ Create authentication token for audience=AltovaMTSGroup solution=/authtest
data={"auth": "test"}

✓ Loading private key for token creation from C:\ProgramData\Altova
\MobileTogetherServer\key.pem

✗ Authentication token creation for audience=AltovaMTSGroup solution=/authtest
data={"auth": "test"} failed:
ValueError('Could not deserialize key data.')

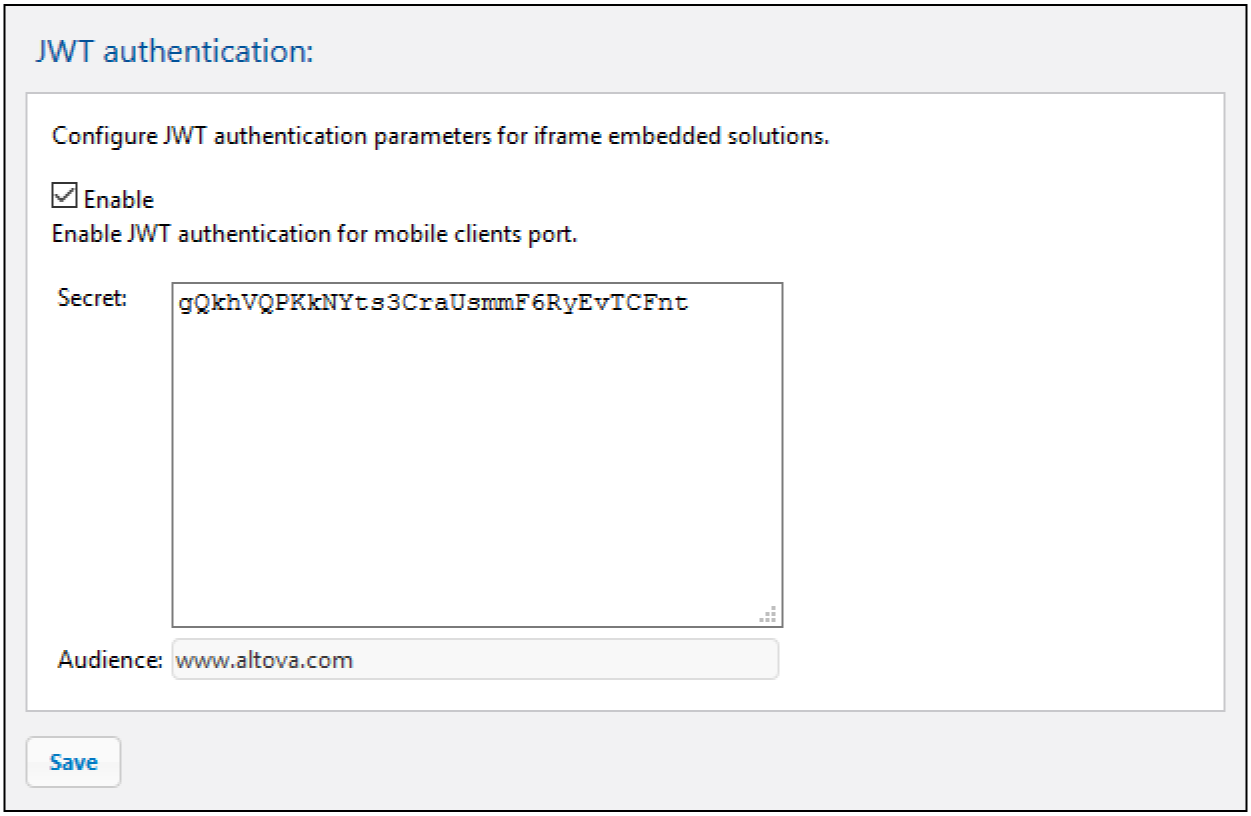
✗ Authentication test for mySolutionServer ended with errors:
UnboundLocalError("local variable 'token' referenced before assignment")

Properties of the remote Solution Server:

- *Solution Host:* This is the hostname of the remote machine hosting the MobileTogether Server solution and with which authenticated communication is to be tested.
- *Authentication Mobile Client Port:* This is the port through which client devices connect to the remote server.
- *Audience:* This is a string that defines the Audience of the solutions for which authentication is allowed.

5.7.5 JWT

The **JWT Authentication** setting *(screenshot below)* enables JSON Web Token (JWT) authentication of embedded webpage solutions. If a solution is embedded in a webpage and JWT authentication is enabled on the server, the solution will be loaded in the embedding webpage without the user having to log in to MobileTogether Server. For more information, see the description of embedded webpage solutions in the MobileTogether Designer documentation.


JWT authentication:

Configure JWT authentication parameters for iframe embedded solutions.

☑ Enable
Enable JWT authentication for mobile clients port.

Secret: gQkhVQPKkNYts3CraUsmmF6RyEvTCFnt

Audience: www.altova.com

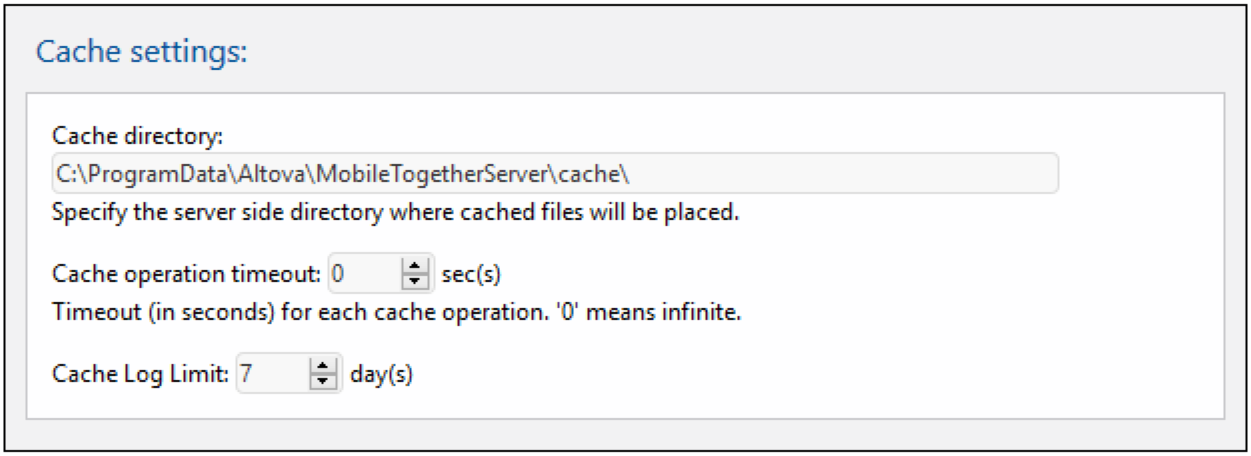
After enabling JWT authentications, there are two settings you must define:

- *Secret:* If you have used a symmetric key (shared secret) to create the JWT, then enter the shared secret key here. If you have used asymmetric encryption (public–private key encryption), then enter the public key here. With this information, the server will be able to verify the JWT that is sent with the first `GET` request from the embedded solution.
- *Audience:* Enter the same string as that you entered for the *Audience* claim when creating the JWT (*see the MobileTogether Designer documentation for more information*).

If you modify a setting, click **Save** at the bottom of the tab for the modified setting to take effect.

5.7.6 Cache

Cache Settings specify: (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 106 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)

If you modify a setting, click **Save** at the bottom of the tab for the modified setting to take effect.

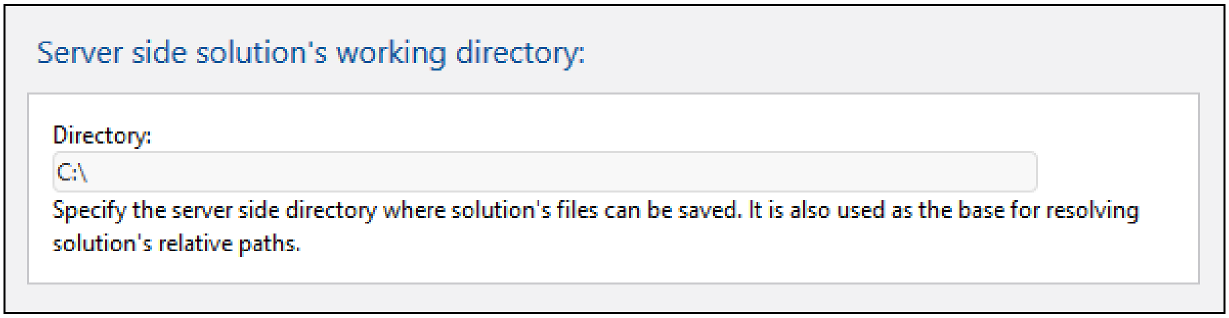
5.7.7 Sources

The **Sources** tab enables you to (i) specify the folder in which data files of solutions are stored, and (ii) define and manage a server file that contains connection information for databases. If you modify any setting, click **Save** at the bottom of the tab for the modified setting to take effect.

▼ 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:\

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.

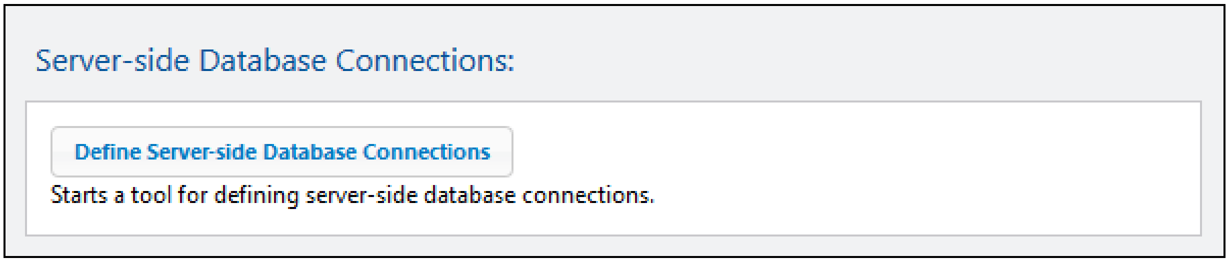
The Working Directory setting, in effect, restricts any read/write access to local files during execution of solutions. Only files inside the Working Directory or any of its sub-directories can be accessed by MobileTogether Server for the execution of solutions.

▼ Server-side database connections

This setting (*screenshot below*) enables you to save database (DB) connections to a DB-connections XML file on the server. Server-side DB connections can then be made and used by a solution's Read DB Structure action to read the data in a DB (*see the MobileTogether Designer user manual for details of how to define this action action*).

Note: This setting is displayed only if the following privileges have been enabled: *Read database structures* 59 and *Manage server settings* 59.

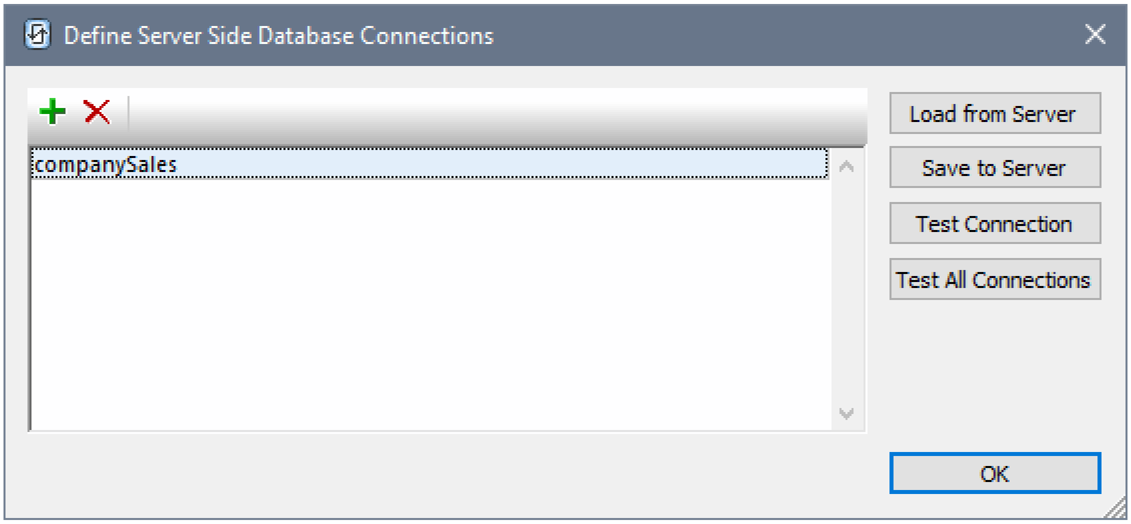
Note: This feature (server-side DB connections) is Windows-specific, and is therefore not available on a Linux-based or macOS-based MobileTogether Server.



Server-side Database Connections:

Starts a tool for defining server-side database connections.

To create a new server-side DB connection or to manage existing connections, click **Define Server-side DB Connections** *(see screenshot above)*. In the dialog that appears, select *Altova Define Server Side DB Connections*, and click **Open Link**. If you are prompted for credentials to access the server, enter these and click **OK**. The Define Server Side Database Connections dialog appears (*screenshot below*).



To create a server-side DB connection, do the following:

1. Click the **Add DB** button in the toolbar at top left (*see screenshot above*).
2. In the DB Connection Wizard that now appears, add a new DB connection by following the wizard's steps.
3. After the DB connection has been created, it appears in the dialog. You can modify the name if you want to by double-clicking the name and editing it. The screenshot above shows that a connection named **`companySales`** has been created.
4. Click **Save to Server** (*see screenshot above*) to save the connection to the server. The connection will be added to an XML file (located on the server) in which all defined DB connections are stored.

The following additional actions can be carried out from the Define Server Side Database Connections dialog:

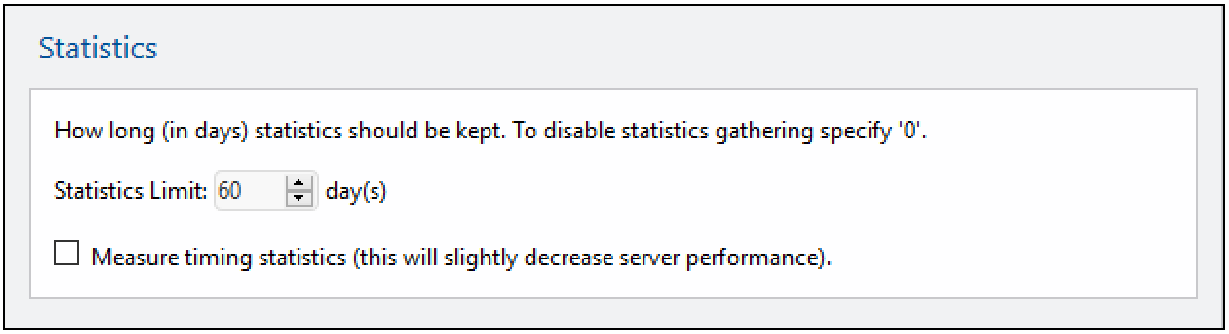
- Delete a connection by selecting it in the dialog and clicking the **Delete** button in the toolbar at top left. Click **Save to Server** to save the modification to the DB-connections file.
- Click **Load from Server** to load the connections that are currently stored in the DB-connections file into the dialog.
- Test a DB connection by selecting it and clicking **Test Connection**. The success or failure of the test is reported in a message window.
- Click **Test All Connections** to test all the connections currently in the window. A message window displays a list of all the connections together with the test result of each.

5.7.8 Misc

The **Miscellaneous (Misc)** tab provides settings for a wide range of server features. If you modify any of the settings displayed in this tab, click **Save** at the bottom of the tab for the modified setting to take effect.

▼ Statistics

Statistics relating to server usage are stored in an internal MobileTogether database. You can view these statistics by opening the **Statistics.mtd** file, which is located for new MobileTogether Server installations (version 4.0 and later) in the **admin** container. The *Statistics Limit* setting (*screenshot below*) enables you to specify the time period for which statistics are kept. The default setting is **0**, meaning that statistics are not tracked.



Statistics

How long (in days) statistics should be kept. To disable statistics gathering specify '0'.

Statistics Limit: 60 day(s)

□ Measure timing statistics (this will slightly decrease server performance).

Note the following points:

- The **admin** container is automatically created only for new installations of MobileTogether Server. If you are updating your version of MobileTogether Server, you will need to explicitly deploy the **Statistics.mtd** workflow to the server. You can deploy it to any container you like, but we recommend that you create an **admin** container, and deploy it there.
- If you need to explicitly deploy the **Statistics.mtd** design, it is available in the **SolutionFiles** folder of the AppData folder of your MobileTogether Server installation *(see table below)*.
- When you run the solution, it will read data from the internal MobileTogether Server statistics database and provide you with an interface, in which you can filter and select options, to view graphs of the statistics in the database.
- The solution shows four main categories of data: (i) the number of users that connect to the server; (ii) the number of different devices that connect to the server; (iii) the number of requests sent to the server; (iv) the number of solution starts that occur on the server (each solution can be started multiple times, and each start counts as an independent start). Additional filtering is also possible (for example, for specific solutions or devices). Note that only solution execution is tracked; administration requests are not tracked.
- For a user to be able to read statistics, the *Read statistics* 78 privilege must be checked for that user.

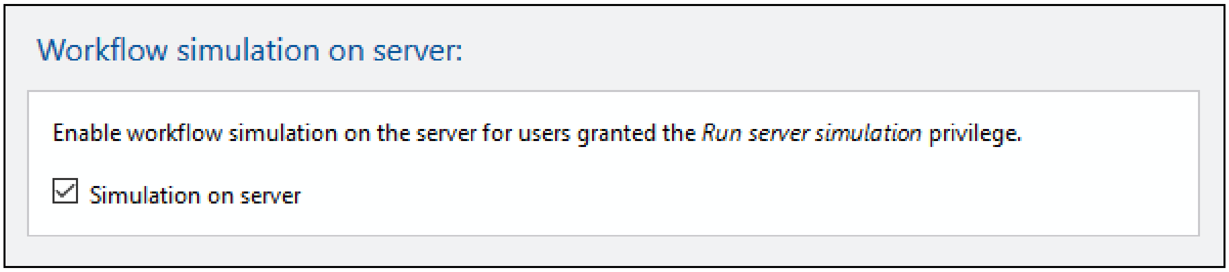
Location of the MobileTogether Server AppData folder on various operating systems

Linux	`/var/opt/Altova/MobileTogetherServer`
Mac	`/var/Altova/MobileTogetherServer`
Windows	`C:\ProgramData\Altova\MobileTogetherServer`

For detailed information about setting up and using the **Statistics** solution, see Solution Usage Statistics 68.

▼ 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.



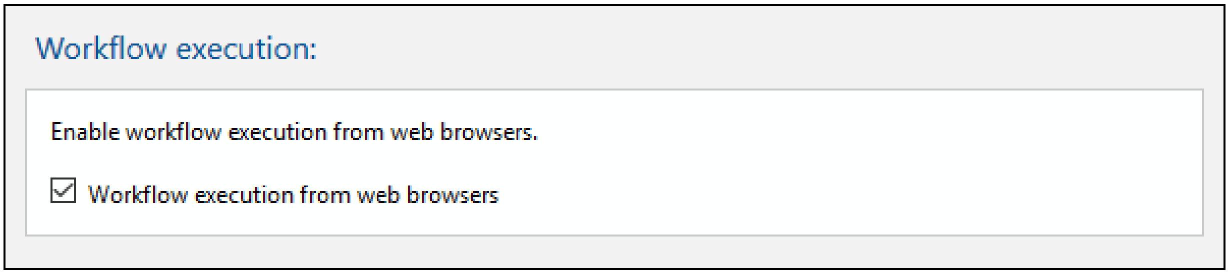
Workflow simulation on server:

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.



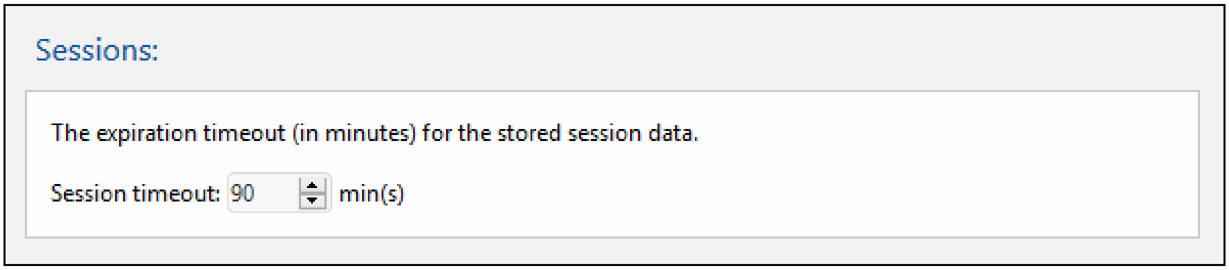
Workflow execution:

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.



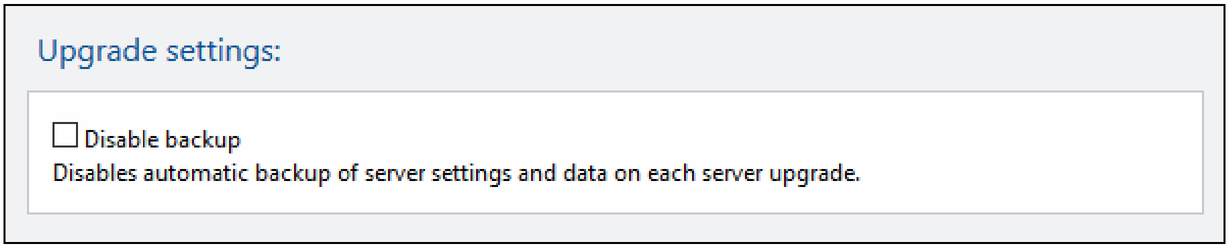
Sessions:

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

Session timeout: 90 min(s)

▼ 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 73 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 73.



Upgrade settings:

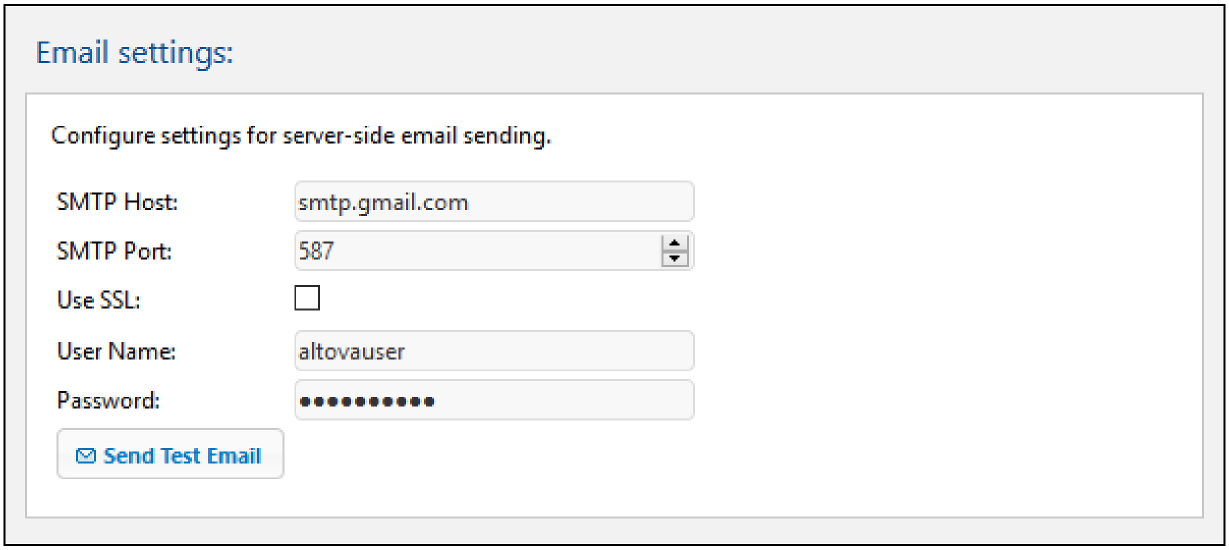
□ Disable backup

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[73] for information about how to do this.

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:

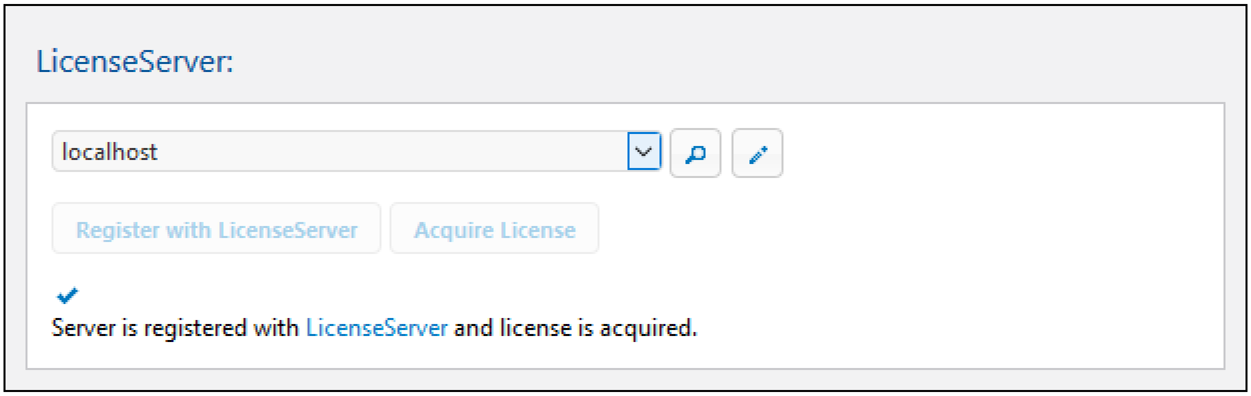
Configure settings for server-side email sending.

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.
- *Use SSL:* Specifies whether SSL is used or not.
- *User Name and Password:* The user name and password of an email account that is registered with the email service provider.

5.7.9 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[13] for information about licensing. If you modify any setting, click **Save** (at the bottom of the tab) for the modified setting to take effect.



✓ 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 **Manually Enter 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.

5.7.10 Config File Settings

Some settings cannot be made in the WebUI *(see previous sections)*, mainly because they do not need to be changed or should be changed only if you understand their effects. These settings are stored in a configuration file named **`mobiletogetherserver.cfg`**, which is located by default in the application data folder (*see below*). You can edit the `.cfg` configuration file in a text editor. This section contains information about important settings that are safe for you to add/edit in the configuration file.

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	`C:\ProgramData\Altova\MobileTogetherServer`

Server variables

You can define server variables in the configuration file, and the values of these server variables can be called in the design via the **`mt-server-variables`** Altova XPath extension function. At runtime, the server variable could be used to get a server-side value, such as the server's name. Actions in the solution can then be designed based on the value returned by the server variable. For example, a different set of actions can be set up according to whether the server is a test server or a live server—without having to build two separate solutions for the two servers.

```
[ServerVariables]
ServerName=Test
MyVariable=8087
```

Data transfer and listening

These settings are in the *Listen, ListenSSL, ListenAdmin,* and *ListenAdminSSL* sections of the configuration file *(see listings below)*.

Server timeout
The default timeout of the server is 10 seconds. If this is too low, you can use the `timeout` setting to set a higher timeout in seconds. If the `timeout` setting is missing or is `< 1`, then the default timeout of 10 seconds is used.

Size limit of data files transmitted to server
The server is set up by default to accept files that are up to 100 MB large. Larger files are rejected. If files larger than 100 MB are expected, you can increase the size limit by specifying the `max_request_body_size` setting in the *Listen* and *ListenSSL* sections of the configuration file. In the listing below, the file size has been increased so that the server can accept files of up to 200 MB. Note that the default value of `max_request_body_size` is 104857600 (100 MB)—even when the setting is not listed in the configuration file.

```
[Listen]
host=0.0.0.0
port=8087
active=1
ssl=0
admin=0
timeout=300
max_request_body_size=209715200

[ListenSSL]
host=0.0.0.0
port=8084
active=1
ssl=1
admin=0
timeout=300
max_request_body_size=209715200
```

Content Security Policy (CSP) settings

Since MobileTogether Server returns a Content Security Policy (CSP) HTTP header each time it receives an HTTP request, CSP is enabled for MobileTogether Server. Browsers that don't support CSP will still work with MobileTogether Server. They will ignore the CSP header and default to the standard same-origin policy for web content. You can customize the CSP HTTP headers that MobileTogether Server sends by using the settings given below, which are in the *Web* and *SSL* sections.

By default, resources (such as images, audio, or video) can be loaded in a web page only from the same location as that from which the web page is served (in our case, that would be MobileTogether Server). You would not normally need to change this default behavior. However, exceptions might arise when content is hosted on a different domain (as, for example, when a MobileTogether solution is embedded in an IFrame). For such situations, you can can customize the CSP directives listed below according to your needs.

For more information about CSP, see https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP and https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy

`[Web]`
`default_src = 'self'`. This is a fallback directive for the other CSP fetch directives below.
`frame_src = 'self'` *(CSP fetch directive.) This directive's value also sets the value of the X-Frame-Options header field (see note below).*
`image_src = 'self'` **`data:<URL>`** *(CSP fetch directive.)*
`media_src = 'self'` **`data:<URL>`** *(CSP fetch directive.)*
`object_src = 'none'` *(CSP fetch directive.)*
`script_src = 'self' 'unsafe-inline' 'unsafe-eval'` *(CSP fetch directive;* **`'self'`** *is default when no value is specified.)*
`style_src = 'self' 'unsafe-inline' 'unsafe-eval'` *(CSP fetch directive;* **`'self'`** *is default when no value is specified.)*
`nosniff = 0` or `1`. *See the note about X-Content-Type-Options below.*
`referrer_policy = strict-origin-when-cross-origin`. *This is the default value. See the note about Referrer Policy below.*

Note the following points:

- If a fetch directive is omitted, then the `default_src` directive is used as a fallback.
- If you want to specify that a fetch directive has more than one value, enter the values as a space-separated list. The listing above contains examples of space-separated values.
- The `X-Frame-Options` header can protect visitors from clickjacking attacks. A value of `DENY` indicates to the browser that your site may not be framed, while a value of `SAMEORIGIN` allows you to frame your own site. The value is automatically selected from the value of the `frame_src` setting. If `frame_src` is is set to `'self'` or `'deny'`, then `X-Frame-Options` is automatically set: respectively, to `SAMEORIGIN` or `DENY`. If `frame_src` is set to a URI, however, then `X-Frame-Options` is not set.
- The `X-Content-Type-Options` header prevents the browser from mime-sniffing the document's content type, thereby forcing the browser to use the document's declared MIME type. This header can take only one value, `nosniff`. In the configuration settings, the `nosniff` setting can take a value of `1` (the default, which switches the value on) and `0` (which causes the header to not be sent).
- Referrer Policy: When a user clicks a link on a web page, the destination site receives information about the originating page. This information is obtained from the referrer header that is stored in the request made to the destination page. The amount of information that is stored in the referrer header is controlled by the `Referrer-Policy` header. `Referrer-Policy` is set automatically to `strict-origin-when-cross-origin`. You can set this header to other values if you like.

Strict Transport Security setting
If this HTTP header is returned in the response, it tells the browser to remember the server and try to connect with HTTPS automatically next time. For more information, see https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security.

[SSL]
strict = `0` or `1`, or is not set

Set the value suitably:

- `0`: the header is sent with `max-age=0`
- `1`: the header is sent with `max-age=31536000` (which is 365 days)
- If no value is set, then the header is not sent in the response

Adding or overwriting HTTP(S) response headers

You can add headers to HTTP or HTTPS responses, or modify the existing headers of a response. To do this, set up an HTTP or HTTPS section in the configuration file and add the new headers you want (*see listings below*). Since the settings in these sections are processed subsequently to the processing of other headers, any already existing header will be modified by the settings in these sections. In each section below, the first setting is a new setting, whereas the second setting overwrites a header that is already defined.

```
[HTTPHeader]
my-new-http-header = some-value
Referrer-Policy = origin
```

```
[HTTPSHeader]
my-new-https-header = some-value
Referrer-Policy = origin
```

Syslog Server settings

These settings are available in the **`[Log]`** group of settings and define the syslog server. Some of these settings can be specified via the Settings page of the Web UI[117]. However, others can only be specified in the config file as listed below.

```
[Log]
syslog_enabled = 0 or 1
syslog_protocol = BSD_UDP or BSD_TCP or IETF_TCP or IETF_TLS
syslog_host = <IP address>
syslog_port = <usually 514 or 601 or 6514>
syslog_key = <key for TLS communication>
syslog_cert = <cert for TLS communication>
```
`syslog_ca =` This setting is not availabe in the UI. It is the Root CA, which is usually already installed on a PC. If you are unable to install it, specify it with this setting.
`syslog_timeout =` This setting is not availabe in the UI. The default is 5 seconds. Note that, if this setting's value is too high and Syslog server is not available, then writing the log to the MobileTogether Server log database could be jeopardized.

6 Command Line

Location of executable

Given below are the default locations of the MobileTogether Server executable, which you can call to execute the commands described n this section:

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

Usage

The command line syntax is:

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

- `--help` (short form `--h`) displays the help text of the given command. If no command is named, then all commands of the executable are listed, each with a brief description of the command.
- `--version` displays the version number of MobileTogether Server.
- `<command>` is the command to execute. Commands are described in the sub-sections of this section *(see list below)*.
- `[options]` are the options of a command; they are listed and described with their respective commands.
- `[arguments]` are the arguments of a command; they are listed and described with their respective commands.

▼ Casing and slashes on the command line

`MobileTogetherServer` *on Windows*
`mobiletogetherserver` *on Windows and 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.

CLI commands

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

- addtorole 136: Adds a principal to a MobileTogether Server role.
- applicationid 137: Returns the application ID.
- assignlicense 138: Uploads a license to LicenseServer and assigns this license to MobileTogether Server.
- createcontainer 140: Creates a new container within the root or within an existing cointainer.
- createrole 141: Creates a new MobileTogether Server role.
- createuser 143: Creates a new MobileTogether Server user.

- debug[145]: Starts MobileTogether Server for debugging.
- deploy[146]: Deploys a MobileTogether package (.mtp file) to MobileTogether Server.
- exportresourcestrings[149]: Exports all application resource strings to an XML file.
- help[153]: Displays information about the command that is submitted in the argument (or about all commands if no argument is submitted).
- install[154]: Installs MobileTogether Server as a service.
- licenseserver[155]: Registers MobileTogether Server with a LicenseServer on the local network.
- resetpassword[157]: Resets the password of MobileTogether Server's administrator interface.
- setdeflang[158]: Sets the default language of MobileTogether Server.
- start[163]: Starts MobileTogether Server as a service.
- uninstall[164]: Uninstalls MobileTogether Server as a service.
- upgradedb[165]: Updates the internal MobileTogether Server database to that of the new MobileTogether Server version and inserts correct default values.
- verifylicense[166]: Checks if current MobileTogether Server is licensed and, optionally, whether it is licensed with the given license key.
- version[167]: Displays the version number of MobileTogether Server.

6.1 addtorole

Syntax and description

The **addtorole** command adds the submitted principal (user or role) to the specified role. See the topic Roles 94 for the relationship of a principal to a role.

```
mobiletogetherserver addtorole [options] Role Principal
```

- The *Role* argument is required and specifies the name of the MobileTogether Server role to which the submitted principal will be added.
- The *Principal* argument is required and submits the name of the principal that is to be added to the specified role.
- Use the **--h, --help** option to display information about the command.

▼ Casing and slashes on the command line

MobileTogetherServer *on Windows*
mobiletogetherserver *on Windows and 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.

Example

Examples of the **addtorole** command:

```
mobiletogetherserver addtorole Legal Tech-01
```

- The command adds the principal named **Tech-01** to the role named **Legal**.

Options

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

6.2 applicationid

Syntax and description

The **`applicationid`** command displays the ID of the MobileTogether application.

```
mobiletogetherserver applicationid [options]
```

▼ Casing and slashes on the command line

`MobileTogetherServer` *on Windows*
`mobiletogetherserver` *on Windows and 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.

Example

Example of the **`applicationid`** command:

```
mobiletogetherserver applicationid
```

- Returns the application id.

Options

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

6.3 assignlicense

Syntax and description

The **assignlicense** command uploads a license file to the Altova LicenseServer with which MobileTogether Server is registered (see the `licenseserver` command), and assigns the license to MobileTogether Server. It takes the path of a license file as its argument. The command also allows you to test the validity of a license.

```
mobiletogetherserver assignlicense [options] FILE
```

- The *FILE* argument takes the path of the license file.
- The `--test-only` option uploads the license file to LicenseServer and validates the license, but does not assign the license to MobileTogether Server.

For details about licensing, see the LicenseServer documentation (https://www.altova.com/manual/en/licenseserver/3.12/).

▼ Casing and slashes on the command line

MobileTogetherServer *on Windows*
mobiletogetherserver *on Windows and 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.

▼ Backslashes, spaces, and special characters on Windows systems

On Windows systems: When spaces or special characters occur in strings (for example in file or folder names, or company, person or product names), use quotes: for example, **"My File"**. Note, however, that a backslash followed by a double-quotation mark (for example, **"C:\My directory\"**) might not be read correctly. This is because the backslash character is also used to indicate the start of an escape sequence, and the escape sequence **\"** stands for the double-quotation mark character. If you want to escape this sequence of characters, use a preceding backslash, like this: **\\"**. To summarize: If you need to write a file path that contains spaces or an end backslash, write it like this: **"C:\My Directory\\"**.

Examples

Examples of the `assignlicense` command:

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

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

Options

Options are listed in short form (if available) and long form. You can use one or two dashes for both short and long forms. An option may or may not take a value. If it takes a value, it is written like this: **`--option=value`**. Values can be specified without quotes except in two cases: (i) when the value string contains spaces, or (ii) when explicitly stated in the description of the option that quotes are required. If an option takes a Boolean value and no value is specified, then the option's default value is `TRUE`. Use the **`--h, --help`** option to display information about the command.

▼ test-only [t]

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

Values are `true|false`. If `true`, then the license file is uploaded to LicenseServer and validated, but not assigned.

6.4 createcontainer

Syntax and description

The **`createcontainer`** command creates a container at the location named in the **`Path`** argument.

```
mobiletogetherserver createcontainer [options] Path
```

- The *`Path`* argument is that path to the container that is to be created, starting with the root container. For example: **`/public/contacts/personal`**. Note that all ancestor containers must exist. For example: In order to create the **`personal`** container given in the previous path, the container **`/public/contacts/`** must exist.
- Use the **`--h, --help`** option to display information about the command.

Casing and slashes on the command line

`MobileTogetherServer` *on Windows*
`mobiletogetherserver` *on Windows and 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.

Example

Examples of the **`createcontainer`** command:

```
mobiletogetherserver createcontainer /public/contacts/personal
```

- The command adds the **`personal`** container as a child of the the container **`/public/contacts`**.

Options

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

6.5 createrole

Syntax and description

The **createrole** command creates a MobileTogether Server role[94] and assigns this role the privileges specified with the **Privileges** argument.

```
mobiletogetherserver createrole [options] Role [Privileges]
```

- The *Role* argument is the name of the MobileTogether Server role to be created.
- The *Privileges* argument is a comma-separated list of the privileges to be granted to the role.
- Use the **--h, --help** option to display information about the command.

▼ List of privileges

Enter the privileges you want to assign to the role. For a description of these privileges, see the topic Roles[94].

- maintain-users
- set-own-password
- override-security
- allow-store-password
- view-log
- view-cache
- view-licenses
- read-users
- manage-settings
- trace-workflow
- read-statistics
- read-dbstructures
- read-globalresource
- write-globalresource
- open-workflow-from-designer
- save-workflow-from-designer
- run-server-simulation

▼ Casing and slashes on the command line

MobileTogetherServer *on Windows*
mobiletogetherserver *on Windows and 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.

Example

Examples of the **createrole** command:

```
mobiletogetherserver createrole MyNewRole maintain-users,set-own-password,override-security
```

- The command adds a new role named `MyNewRole`, which has the privileges `maintain-users`, `set-own-password`, `override-security`.

Options

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

6.6 createuser

Syntax and description

The **`createuser`** command creates a MobileTogether Server user 89 and assigns this user an initial password and privileges.

```
mobiletogetherserver createuser [options] User Password [Privileges]
```

- The *`User`* argument is the name of the MobileTogether Server user to be created.
- The *`Password`* argument sets the initial password of this user. This argument is mandatory.
- The *`Privileges`* argument is optional. It is a comma-separated list of the privileges granted to the user.
- Use the **`--h, --help`** option to display information about the command.

▼ List of privileges

Enter the privileges you want to assign to the role. For a description of these privileges, see the topic Roles 94.

- maintain-users
- set-own-password
- override-security
- allow-store-password
- view-log
- view-cache
- view-licenses
- read-users
- manage-settings
- trace-workflow
- read-statistics
- read-dbstructures
- read-globalresource
- write-globalresource
- open-workflow-from-designer
- save-workflow-from-designer
- run-server-simulation

▼ Casing and slashes on the command line

`MobileTogetherServer` *on Windows*
`mobiletogetherserver` *on Windows and 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.

Example

Examples of the **`createuser`** command:

```
mobiletogetherserver createuser NewUser NewUserPwd
mobiletogetherserver createuser NewUser NewUserPwd maintain-users,set-own-
password,override-security
mobiletogetherserver createuser --change_password_on_next_login=true NewUser NewUserPwd
mobiletogetherserver createuser --passwordpolicy=PolicyName NewUser NewUserPwd
```

- The first command creates a new user with an initial password; no privileges are assigned.
- The second command creates a new user with an initial password, and assigns this user three privileges.
- The third command creates a new user with an initial password. The initial password must be changed when the user logs in.
- The fourth command creates a new user with an initial password. The password policy 98 to be used for this user's password is specified.

Options

Options are listed in short form (if available) and long form. You can use one or two dashes for both short and long forms. An option may or may not take a value. If it takes a value, it is written like this: **--option=value**. Values can be specified without quotes except in two cases: (i) when the value string contains spaces, or (ii) when explicitly stated in the description of the option that quotes are required. If an option takes a Boolean value and no value is specified, then the option's default value is TRUE. Use the **--h, --help** option to display information about the command.

▼ change_password_on_next_login

--change_password_on_next_login = true|false
This option determines whether the user must change their password on the next login. The default—that is, if the option is not specified—is false.

▼ passwordpolicy

--passwordpolicy = *Policy*
Sets the password policy that must be followed for this user's password. For information about creating password policies, see the topic Password Policies 98.

6.7 debug

Syntax and description

The **debug** command starts MobileTogether Server for debugging—not as a service. To stop MobileTogether Server in this mode, press **Ctrl+C**.

```
mobiletogetherserver debug [options]
```

▼ Casing and slashes on the command line

MobileTogetherServer *on Windows*
mobiletogetherserver *on Windows and 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.

▼ Backslashes, spaces, and special characters on Windows systems

On Windows systems: When spaces or special characters occur in strings (for example in file or folder names, or company, person or product names), use quotes: for example, **`"My File"`**. Note, however, that a backslash followed by a double-quotation mark (for example, **`"C:\My directory\"`**) might not be read correctly. This is because the backslash character is also used to indicate the start of an escape sequence, and the escape sequence **`\"`** stands for the double-quotation mark character. If you want to escape this sequence of characters, use a preceding backslash, like this: **`\\"`**. To summarize: If you need to write a file path that contains spaces or an end backslash, write it like this: **`"C:\My Directory\\"`**.

Example

Example of the **debug** command:

```
mobiletogetherserver debug
```

6.8 deploy

Syntax and description

The **deploy** command deploys a MobileTogether package (**.mtp** file) to MobileTogether Server. When a MobileTogether package is created in MobileTogether Designer, the deployment path can be specified in the package, but does not need to be. The arguments and options of the **deploy** command described below take this into account.

```
mobiletogetherserver deploy [options] Package [Path]
```

- ***Package*** specifies the path to the MobileTogether package that you want to deploy.
- ***Path*** (optional) specifies the location on the server where you want to deploy the package. If this argument is specified on the command line and the package already contains a deployment path, then the path supplied on the command line is used and the deployment path in the package is ignored. If this argument is not supplied and the package contains no deployment path, then an error message about this is displayed.
- If a package having the same name already exists at the deployment location and you want to overwrite it, use the **force** option *(see below)*. If you do not use **force** in this situation, an error will be displayed, indicating that a package already exists at the specified deployment location.
- You can specify the input parameters that will be used in actions of the project's **OnServerDeploy** event (*see the MobileTogether Designer documentation*).

Note: The server must be stopped before this command is executed.

Casing and slashes on the command line

MobileTogetherServer *on Windows*
mobiletogetherserver *on Windows and 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.

Backslashes, spaces, and special characters on Windows systems

On Windows systems: When spaces or special characters occur in strings (for example in file or folder names, or company, person or product names), use quotes: for example, **"My File"**. Note, however, that a backslash followed by a double-quotation mark (for example, **"C:\My directory\"**) might not be read correctly. This is because the backslash character is also used to indicate the start of an escape sequence, and the escape sequence **\"** stands for the double-quotation mark character. If you want to escape this sequence of characters, use a preceding backslash, like this: **\\"**. To summarize: If you need to write a file path that contains spaces or an end backslash, write it like this: **"C:\My Directory\\"**.

Example

Examples of the **deploy** command:

```
mobiletogetherserver deploy "C:\temp\ParcelDelivery.mtp"
mobiletogetherserver deploy --force "C:\temp\ParcelDelivery.mtp"
mobiletogetherserver deploy "C:\temp\ParcelDelivery.mtp" "/public/ParcelDelivery"
mobiletogetherserver deploy --force "C:\temp\ParcelDelivery.mtp"
"/public/ParcelDelivery"
mobiletogetherserver deploy --force --force_solutionfile=datalib\cust-NY.sqlite --
force_solutionfile=datalib\cust-MA.sqlite "C:\temp\ParcelDelivery.mtp"
mobiletogetherserver deploy --force --input_parameters="P1=5089; MyP2='space separated
words'; SomeP3=JoinedWords" "C:\temp\ParcelDelivery.mtp"
```

Options

Options are listed in short form (if available) and long form. You can use one or two dashes for both short and long forms. An option may or may not take a value. If it takes a value, it is written like this: **`--option=value`**. Values can be specified without quotes except in two cases: (i) when the value string contains spaces, or (ii) when explicitly stated in the description of the option that quotes are required. If an option takes a Boolean value and no value is specified, then the option's default value is `TRUE`. Use the **`--h, --help`** option to display information about the command.

▼ datadir

`--datadir = `*`PathToDatabaseDirectory`*
Specifies the path of the database directory.

▼ force

`--force`
If this option is specified, then the MobileTogether package that is being deployed (by the **`deploy`** command) will overwrite any package of the same name that is at the location specified by the deployment path used by the **`deploy`** command. The deployment path is taken either from the package or is specified in the ***`Path`*** argument *(see above)*. If you do not use **`force`** in this situation, an error will be displayed, indicating that a package already exists at the specified deployment location.

▼ force_solutionfile

`--force_solutionfile = `*`PathToSSSFile`*
Specifies an already deployed server side solution file that you want to overwrite. The **`PathToSSSFile`** is the path to the server side solution file on the server and it is relative to the server side solution's working directory 124. To overwrite multiple solution files, specify this option as many times as required (see example above). If an existing solution file is not specified using this option, then it is not overwritten.

It is assumed that the server side solution files are available in the package. If a solution file exists in the package but is not present on the server, then it is written to the server.

▼ input_parameters

`--input_parameters = `*`Parameters`*
Defines the parameters of actions that are executed for the **`OnServerDeploy`** event. The entire parameter list must be enclosed in quotes. Parameter values must be enclosed in single quotes.

For example: **`--input_parameters="`**`P1=5089; P2='space separated words'; P3=JoinedWords`**`"`**

Also see the examples above.

6.9 exportresourcestrings

Syntax and description

The **`exportresourcestrings`** command outputs an XML file containing the resource strings of the MobileTogether Server application in the specified language. Available export languages are English (`en`), German (`de`), Spanish (`es`), French (`fr`), and Japanese (`ja`).

```
mobiletogetherserver exportresourcestrings [options] LanguageCode XMLOutputFile
```

- The ***`LanguageCode`*** argument gives the language of the resource strings in the output XML file; this is the *export language*. Allowed export languages (with their language codes in parentheses) are: English (`en`), German, (`de`), Spanish (`es`), French (`fr`), and Japanese (`ja`).
- The ***`XMLOutputFile`*** argument specifies the path and name of the output XML file.

How to create localizations is described below.

▼ Casing and slashes on the command line

`MobileTogetherServer` *on Windows*
`mobiletogetherserver` *on Windows and 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.

▼ Backslashes, spaces, and special characters on Windows systems

On Windows systems: When spaces or special characters occur in strings (for example in file or folder names, or company, person or product names), use quotes: for example, **`"My File"`**. Note, however, that a backslash followed by a double-quotation mark (for example, **`"C:\My directory\"`**) might not be read correctly. This is because the backslash character is also used to indicate the start of an escape sequence, and the escape sequence **`\"`** stands for the double-quotation mark character. If you want to escape this sequence of characters, use a preceding backslash, like this: **`\\"`**. To summarize: If you need to write a file path that contains spaces or an end backslash, write it like this: **`"C:\My Directory\\"`**.

Examples

Examples of the **`exportresourcestrings`** command:

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

- The command above creates a file called `Strings.xml` at `c:\` that contains the resource strings of MobileTogether Server in German.

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\MobileTogetherServer\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 five 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\MobileTogetherServer\bin` folder. Your DLL file will have a name of the form `MobileTogetherServer2024_lc.dll`. The `_lc` part of the name contains the language code. For example, in `MobileTogetherServer2024_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 grant

Syntax and description

The **`grant`** command sets what permissions a principal (user or role) has for a specific container. The server must be stopped before this command is executed.

```
mobiletogetherserver grant [options] Principal Path Container Workflow Security
```

- All five arguments are mandatory.
- ***`Principal`*** specifies the user or role for which permissions are being assigned. The principal must already be defined on the server.
- ***`Path`*** specifies the path to the container for which permissions are being assigned. The path to the container must be an absolute path starting at the directory root.
- ***`Container`*** specifies the container's permission level (`read-write | read | inherit | none`).
- ***`Workflow`*** specifies the workflow's permission level (`read-write-use | read-use | inherit | none`).
- ***`Security`*** specifies the level of access the principal has to the container's security settings (`read-write | read | inherit | none`).

Note: For a description of the values of permissions, see Workflows | Permissions [78].

▼ Casing and slashes on the command line

`MobileTogetherServer` *on Windows*
`mobiletogetherserver` *on Windows and 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.

▼ Backslashes, spaces, and special characters on Windows systems

On Windows systems: When spaces or special characters occur in strings (for example in file or folder names, or company, person or product names), use quotes: for example, **`"My File"`**. Note, however, that a backslash followed by a double-quotation mark (for example, **`"C:\My directory\"`**) might not be read correctly. This is because the backslash character is also used to indicate the start of an escape sequence, and the escape sequence **`\"`** stands for the double-quotation mark character. If you want to escape this sequence of characters, use a preceding backslash, like this: **`\\"`**. To summarize: If you need to write a file path that contains spaces or an end backslash, write it like this: **`"C:\My Directory\\"`**.

Example

Examples of the **`grant`** command:

```
mobiletogetherserver grant tech-01 /public/contact read-write read-write-use read-write
mobiletogetherserver grant tech-02 /public/contact inherit inherit inherit
mobiletogetherserver grant tech-03 /public/contact read read-use none
```

Options

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

6.11 help

Syntax and description

The **help** command takes a single argument (`Command`), which is the name of the command for which help is required. It displays the command's syntax, its options, and other relevant information. If the `Command` argument is not specified, then all commands of the executable are listed, with each having a brief text description.

```
mobiletogetherserver help Command
```

▼ Casing and slashes on the command line

MobileTogetherServer *on Windows*
mobiletogetherserver *on Windows and 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.

Example

Example of the **help** command to display information about the **licenserver** command:

```
mobiletogetherserver help licenseserver
```

The --help option

Help information about a command is also available by using the `--help` option of the command for which help information is required. The two commands below produce the same results:

```
mobiletogetherserver licenseserver --help
```

The command above uses the `--help` option of the **licenseserver** command.

```
mobiletogetherserver help licenseserver
```

The **help** command takes `licenseserver` as its argument.

Both commands display help information about the **licenseserver** command.

6.12 install

Syntax and description

The **`install`** command installs MobileTogether Server as a service on the server machine.

```
mobiletogetherserver install [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.
- Use the **`--h, --help`** option to display information about the command.

▼ Casing and slashes on the command line

`MobileTogetherServer` *on Windows*
`mobiletogetherserver` *on Windows and 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.

Example

Example of the **`install`** command:

```
mobiletogetherserver install
```

6.13 licenseserver

Syntax and description

The **`licenseserver`** command registers MobileTogether Server with the Altova LicenseServer specified by the *`Server-Or-IP-Address`* argument. For the `licenseserver` command to be executed successfully, the two servers (MobileTogether Server and LicenseServer) must be on the same network and LicenseServer must be running. You must also have administrator privileges in order to register MobileTogether Server with LicenseServer.

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

- The *`Server-Or-IP-Address`* argument takes the name or IP address of the LicenseServer machine.

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 about licensing, see the LicenseServer documentation (https://www.altova.com/manual/en/licenseserver/3.12/).

▼ Casing and slashes on the command line

`MobileTogetherServer` *on Windows*
`mobiletogetherserver` *on Windows and 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.

▼ Backslashes, spaces, and special characters on Windows systems

On Windows systems: When spaces or special characters occur in strings (for example in file or folder names, or company, person or product names), use quotes: for example, **`"My File"`**. Note, however, that a backslash followed by a double-quotation mark (for example, **`"C:\My directory\"`**) might not be read correctly. This is because the backslash character is also used to indicate the start of an escape sequence, and the escape sequence **`\"`** stands for the double-quotation mark character. If you want to escape this sequence of characters, use a preceding backslash, like this: **`\\"`**. To summarize: If you need to write a file path that contains spaces or an end backslash, write it like this: **`"C:\My Directory\\"`**.

Examples

Examples of the **`licenseserver`** command:

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

The commands above specify, respectively, the machine named `DOC.altova.com`, and the user's machine (`localhost` and `127.0.0.1`) as the machine running Altova LicenseServer. In each case, the command registers MobileTogether Server with the LicenseServer on the machine specified. The last command calls the server-executable to execute the command.

Options

Options are listed in short form (if available) and long form. You can use one or two dashes for both short and long forms. An option may or may not take a value. If it takes a value, it is written like this: **`--option=value`**. Values can be specified without quotes except in two cases: (i) when the value string contains spaces, or (ii) when explicitly stated in the description of the option that quotes are required. If an option takes a Boolean value and no value is specified, then the option's default value is `TRUE`. Use the **`--h, --help`** option to display information about the command.

▼ json [j]

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

Values are `true|false`. If `true`, prints the result of the registration attempt as a machine-parsable JSON object.

6.14 resetpassword

Syntax and description

The **`resetpassword`** command resets the password of the **`root`** user to the default value (**`root`**), and grants the root user all privileges. The running instance of MobileTogether Server must be stopped before performing this operation.

```
mobiletogetherserver resetpassword [options]
```

▼ Casing and slashes on the command line

`MobileTogetherServer` *on Windows*
`mobiletogetherserver` *on Windows and 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.

▼ Backslashes, spaces, and special characters on Windows systems

On Windows systems: When spaces or special characters occur in strings (for example in file or folder names, or company, person or product names), use quotes: for example, **`"My File"`**. Note, however, that a backslash followed by a double-quotation mark (for example, **`"C:\My directory\"`**) might not be read correctly. This is because the backslash character is also used to indicate the start of an escape sequence, and the escape sequence **`\"`** stands for the double-quotation mark character. If you want to escape this sequence of characters, use a preceding backslash, like this: **`\\"`**. To summarize: If you need to write a file path that contains spaces or an end backslash, write it like this: **`"C:\My Directory\\"`**.

Example

Example of the **`resetpassword`** command:

```
mobiletogetherserver resetpassword --datadir=C:
\ProgramData\Altova\MobileTogetherServer\mobiletogether.db
```

Options

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

▼ datadir

`--datadir = ` *`PathToDatabaseDirectory`*
Specifies the path of the database directory.

6.15 setdeflang

Syntax and description

The **`setdeflang`** command (short form is **`sdl`**) sets the default language of MobileTogether Server. Available languages are English (`en`), German (`de`), Spanish (`es`), French (`fr`), and Japanese (`ja`). The command takes a mandatory *`LanguageCode`* argument.

```
mobiletogetherserver setdeflang [options] LanguageCode
```

- The *`LanguageCode`* argument is required and sets the default language of MobileTogether Server. The respective values to use are: `en`, `de`, `es`, `fr`, `ja`.
- Use the **`--h, --help`** option to display information about the command.

▼ Casing and slashes on the command line

`MobileTogetherServer` *on Windows*
`mobiletogetherserver` *on Windows and 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.

Examples

Examples of the **`setdeflang`** (**`sdl`**) command:

```
mobiletogetherserver sdl de
mobiletogetherserver setdeflang es
```

- The first command sets the default language of MobileTogether Server to German.
- The second command sets the default language of MobileTogether Server to Spanish.

Options

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

6.16 setpassword

Syntax and description

The **`setpassword`** command sets or resets the password of any user. The server must be stopped before this command is executed.

```
mobiletogetherserver setpassword [options] User Password
```

- Both arguments are mandatory.
- ***`User`*** specifies the user for which the password is being assigned. The user must already be defined on the server.
- ***`Password`*** specifies the new password to assign to the user named in the previous argument.

▼ Casing and slashes on the command line

`MobileTogetherServer` *on Windows*
`mobiletogetherserver` *on Windows and 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.

▼ Backslashes, spaces, and special characters on Windows systems

On Windows systems: When spaces or special characters occur in strings (for example in file or folder names, or company, person or product names), use quotes: for example, **`"My File"`**. Note, however, that a backslash followed by a double-quotation mark (for example, **`"C:\My directory\"`**) might not be read correctly. This is because the backslash character is also used to indicate the start of an escape sequence, and the escape sequence **`\"`** stands for the double-quotation mark character. If you want to escape this sequence of characters, use a preceding backslash, like this: **`\\"`**. To summarize: If you need to write a file path that contains spaces or an end backslash, write it like this: **`"C:\My Directory\\"`**.

Example

Examples of the **`setpassword`** command:

```
mobiletogetherserver setpassword "tech-01" myNewPassword
mobiletogetherserver setpassword tech01 myNewPassword
```

Options

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

▼ datadir

`--datadir =` *`PathToDatabaseDirectory`*
Specifies the path of the database directory.

6.17 setsmtp

Syntax and description

The **setsmtp** command enables you to configure the email server's settings. The arguments of the command are equivalent to the values set in the Misc tab of the Settings page[126]. The server must be stopped before this command is executed.

```
mobiletogetherserver setsmtp [options] --host=StringValue --port=StringValue --
ssl=true|false
```

- The **--host**, **--port**, and **--ssl** arguments are mandatory.
- ***host*** and ***port*** specify the SMTP host name and SMTP port of your ISP's SMTP server. These details are provided to you by your ISP.
- ***ssl*** specifies whether SSL is used or not.

▼ Casing and slashes on the command line

MobileTogetherServer *on Windows*
mobiletogetherserver *on Windows and 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.

▼ Backslashes, spaces, and special characters on Windows systems

On Windows systems: When spaces or special characters occur in strings (for example in file or folder names, or company, person or product names), use quotes: for example, **"My File"**. Note, however, that a backslash followed by a double-quotation mark (for example, **"C:\My directory\"**) might not be read correctly. This is because the backslash character is also used to indicate the start of an escape sequence, and the escape sequence **\"** stands for the double-quotation mark character. If you want to escape this sequence of characters, use a preceding backslash, like this: **\\"**. To summarize: If you need to write a file path that contains spaces or an end backslash, write it like this: **"C:\My Directory\\"**.

Example

Examples of the **setsmtp** command:

```
mobiletogetherserver setsmtp --host=mySMTPServer --port=25 --ssl=false
mobiletogetherserver setsmtp --host=mySMTPServer --port=25 --ssl=false --user=AltovaMT
--password=MyPassword
```

Options

Options are listed in short form (if available) and long form. You can use one or two dashes for both short and long forms. An option may or may not take a value. If it takes a value, it is written like this: **--option=value**. Values can be specified without quotes except in two cases: (i) when the value string contains spaces, or (ii) when explicitly stated in the description of the option that quotes are required. If an option takes a Boolean value and no value is specified, then the option's default value is `TRUE`. Use the **`--h, --help`** option to display information about the command.

▼ change_password_on_next_login

`--change_password_on_next_login =` `true|false`
This option determines whether the user must change their password on the next login. The default—that is, if the option is not specified—is `false`.

▼ datadir

`--datadir =` *`PathToDatabaseDirectory`*
Specifies the path of the database directory.

▼ password

`--password =` *`StringValue`*
Sets the password for accessing this user's email account.

▼ passwordpolicy

`--passwordpolicy =` *`Policy`*
Sets the password policy that must be followed for this user's password. For information about creating password policies, see the topic Password Policies 98.

▼ user

`--user =` *`StringValue`*
Specifies the user name of an email account that is registered with the email service provider.

6.18 start

Syntax and description

The **start** command starts MobileTogether Server as a service on the server machine.

```
mobiletogetherserver start [options]
```

- If MobileTogether Server is not installed as a service, install it first with the **install** command (before starting it).
- To uninstall MobileTogether Server as a service, use the **uninstall** command.
- Use the **--h, --help** option to display information about the command.

Casing and slashes on the command line

MobileTogetherServer *on Windows*
mobiletogetherserver *on Windows and 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.

Backslashes, spaces, and special characters on Windows systems

On Windows systems: When spaces or special characters occur in strings (for example in file or folder names, or company, person or product names), use quotes: for example, **"My File"**. Note, however, that a backslash followed by a double-quotation mark (for example, **"C:\My directory\"**) might not be read correctly. This is because the backslash character is also used to indicate the start of an escape sequence, and the escape sequence **\"** stands for the double-quotation mark character. If you want to escape this sequence of characters, use a preceding backslash, like this: **\\"**. To summarize: If you need to write a file path that contains spaces or an end backslash, write it like this: **"C:\My Directory\\"**.

Example

Example of the **start** command:

```
mobiletogetherserver start
```

6.19 uninstall

Syntax and description

The **uninstall** command uninstalls MobileTogether Server as a service on the server machine.

```
mobiletogetherserver uninstall [options]
```

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

▼ Casing and slashes on the command line

MobileTogetherServer *on Windows*
mobiletogetherserver *on Windows and 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.

Example

Example of the **uninstall** command:

```
mobiletogetherserver uninstall
```

6.20 upgradedb

Syntax and 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. All solutions that exist in the old database will be available in the upgraded database.

If an error occurs during the upgrade, the upgrade is stopped and is rolled back.

The **`--nosamples`** option enables you to install a new database so that it has none of the Altova sample solutions. This is useful if you want to install a clean MobileTogether Server that would have only the solutions that you want to deploy there.

```
mobiletogetherserver upgradedb [options]
```

▼ Casing and slashes on the command line

`MobileTogetherServer` *on Windows*
`mobiletogetherserver` *on Windows and 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.

Example

Example of the **`upgradedb`** command:

```
mobiletogetherserver upgradedb --nosamples
```

Options

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

▼ datadir

`--datadir =` *`PathToDatabaseDirectory`*
Specifies the path of the database directory.

▼ nosamples

`--nosamples`
If specified, a new database is created that contains none of the Altova sample solutions (which would be deployed in a standard installation or if the **`upgradedb`** command is used without this option).

6.21 verifylicense

Syntax and description

The **verifylicense** command 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.

```
mobiletogetherserver verifylicense [options]
```

- To check whether a specific license is assigned to MobileTogether Server, supply the license key as the value of the `--license-key` option.

For details about licensing, see the LicenseServer documentation (https://www.altova.com/manual/en/licenseserver/3.12/).

▼ Casing and slashes on the command line

MobileTogetherServer *on Windows*
mobiletogetherserver *on Windows and 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.

Examples

Example of the **verifylicense** command:

```
mobiletogetherserver verifylicense
mobiletogetherserver verifylicense --license-key=ABCD123-ABCD123-ABCD123-ABCD123-
ABCD123-ABCD123
```

- The first command checks whether MobileTogether Server is licensed.
- The second command checks whether MobileTogether Server is licensed with the license key specified with the `--license-key` option.

Options

Options are listed in short form (if available) and long form. You can use one or two dashes for both short and long forms. An option may or may not take a value. If it takes a value, it is written like this: **`--option=value`**. Values can be specified without quotes except in two cases: (i) when the value string contains spaces, or (ii) when explicitly stated in the description of the option that quotes are required. If an option takes a Boolean value and no value is specified, then the option's default value is `TRUE`. Use the **`--h, --help`** option to display information about the command.

▼ license-key [l]

`--l, --license-key = ` *`Value`*
Checks whether MobileTogether Server is licensed with the license key specified as the value of this option.

6.22 version

Syntax and description

The **`version`** command displays the version number of MobileTogether Server.

```
mobiletogetherserver version
```

▼ Casing and slashes on the command line

`MobileTogetherServer` *on Windows*
`mobiletogetherserver` *on Windows and 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.

Example

Example of the `version` command:

```
mobiletogetherserver version
```

Index