Table of Contents

1 Altova MobileTogether Server Advanced Edition ............................................. 3

2 Introduction ...................................................................................................... 6
   2.1 MobileTogether Overview ......................................................................... 7
   2.2 Using MobileTogether Server ................................................................... 9

3 Setting Up MobileTogether Server .................................................................. 12
   3.1 Setup on Windows ................................................................................... 13
       3.1.1 Installation on Windows .................................................................. 14
       3.1.2 Licensing on Windows .................................................................... 16
   3.2 Setup on Linux .......................................................................................... 20
       3.2.1 Installation on Linux ....................................................................... 21
       3.2.2 Licensing on Linux ........................................................................ 24
       3.2.3 Notes about Environment ................................................................ 27
   3.3 Setup on macOS ....................................................................................... 29
       3.3.1 Installation on macOS ..................................................................... 30
       3.3.2 Licensing on macOS ....................................................................... 33
       3.3.3 Notes about Environment ................................................................ 36
   3.4 Setting Up SSL Encryption ........................................................................ 38

4 Server Procedures ............................................................................................ 44
   4.1 Start Altova LicenseServer ....................................................................... 45
   4.2 Start MobileTogether Server ..................................................................... 47
   4.3 Set Up SSL Encryption ............................................................................ 49
   4.4 Set Administrator and Mobile Client Ports .............................................. 54
   4.5 Users and Roles ....................................................................................... 58
   4.6 Available Privileges .................................................................................. 61
   4.7 Configure the Firewall ............................................................................. 64
   4.8 Configure Services .................................................................................... 65
       4.8.1 Timer Triggers .................................................................................. 67
Index
Chapter 1

Altova MobileTogether Server Advanced Edition
1 Altova MobileTogether Server Advanced Edition


- 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.

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

Also see: Demo videos about MobileTogether Server.

This documentation
This documentation is organized into the following sections:

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

**Current version: 5.4**

*Last updated: 05 June 2019*
Chapter 2

Introduction
2 Introduction

This introduction:

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

MobileTogether consists of the following modules:

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

---

**System requirements**

- **MobileTogether Designer**

| Windows          | Windows 7 SP1 with Platform Update, Windows 8, Windows 10 |
## MobileTogether Server

<table>
<thead>
<tr>
<th>Windows Server</th>
<th>Windows Server 2008 R2 SP1 with Platform Update or newer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td>Windows 7 SP1 with Platform Update, Windows 8, Windows 10</td>
</tr>
<tr>
<td>Windows Server</td>
<td>Windows Server 2008 R2 SP1 with Platform Update or newer</td>
</tr>
<tr>
<td>Linux</td>
<td>CentOS 6 or newer, RedHat 6 or newer, Debian 8 or newer, Ubuntu 14.04 or newer</td>
</tr>
<tr>
<td>macOS</td>
<td>macOS 10.12 or newer</td>
</tr>
</tbody>
</table>

## MobileTogether Client

<table>
<thead>
<tr>
<th>iOS</th>
<th>9 and higher for Apple mobile devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android</td>
<td>4.1 and higher for Android mobile devices</td>
</tr>
<tr>
<td>Windows RT, Metro</td>
<td>Windows 10; Windows RT for Windows touch-enabled PCs and tablet computers</td>
</tr>
<tr>
<td>HTML</td>
<td>HTML browsers for any other mobile devices</td>
</tr>
</tbody>
</table>
2.2 **Using MobileTogether Server**

To set up MobileTogether Server for use with MobileTogether clients:

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

The steps in more detail:

1. **Install MobileTogether Server**
   
   MobileTogether Server runs on Windows, Linux, and macOS systems. Before installing a new version of MobileTogether Server, de-install any previous version. See [Installation on Windows](#), [Installation on Linux](#), and [Installation on macOS](#).

2. **License MobileTogether Server**
   
   In order to license MobileTogether Server, it must be able to connect to a LicenseServer on your network. Start MobileTogether Server, register MobileTogether Server with LicenseServer, and assign a license to MobileTogether Server from LicenseServer. See [Licensing on Windows](#), [Licensing on Linux](#), and [Licensing on macOS](#).

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

4. **Define basic settings**
   
   Basic settings include administrator and client ports, and other communication settings and security settings.

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

   - **Administrator access:** Administrator access is via the Web UI and is used to carry out administrative tasks. Administrative actions include defining communication settings, security settings, and managing user accounts.
   - **End-user access:** End user access is via a mobile device and is used to download MobileTogether solutions to the client. Access to solutions on the server is determined by the user account the client logs in with.
6. Deploy MobileTogether solutions to MobileTogether Server

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

7. Configure MobileTogether Client apps to access MobileTogether Server

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

Server IP address and network firewall settings

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

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

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

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

**Tip:** Port 80 is usually open on most firewalls by default. So, if you are having difficulties with firewall settings and if port 80 is not already bound to some other service, you could specify port 80 as the MobileTogether Server port for client communication.
Chapter 3

Setting Up MobileTogether Server
3 Setting Up MobileTogether Server

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

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

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

**Installation on Windows**
- System requirements
- Installing MobileTogether Server
- Altova LicenseServer
- LicenseServer versions
- Trial license
- Application folder location

**Licensing on Windows**
- Start ServiceController
- Start LicenseServer
- Start MobileTogether Server
- Register MobileTogether Server
- Assign a license
3.1.1 Installation on Windows

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

▼ System requirements

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

▼ Windows Server
Windows Server 2008 R2 SP1 with Platform Update or newer

▼ Installing MobileTogether Server

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

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

▼ Altova LicenseServer

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

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

▼ LicenseServer versions

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

Current version: 3.2

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

▼ Application folder location
The application will be installed in the following folder:

<table>
<thead>
<tr>
<th>Windows 7, 8, 10</th>
<th>C:\Program Files\Altova\</th>
</tr>
</thead>
<tbody>
<tr>
<td>32 bit Version on 64-bit OS</td>
<td>C:\Program Files (x86)\Altova\</td>
</tr>
</tbody>
</table>
3.1.2 Licensing on Windows

MobileTogether Server 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.

The steps to carry out are given below in brief. For detailed information, see the LicenseServer user manual at the Altova website.

**Start ServiceController**

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

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

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

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

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

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.
Register MobileTogether Server

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

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

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

```
MobileTogetherServer licenseserver localhost
```

Alternatively, you can register MobileTogether Server from the Settings tab of MobileTogether Server's Web UI. Essentially: (i) Start MobileTogether Server via ServiceController (see previous point); (ii) Enter your password to access the Setup page; (iii) Select the LicenseServer name or address, and click Register with LicenseServer.

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

Assign a license

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

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

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

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

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

**MobileTogether Server licenses**

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

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

**Installation on Linux**
- System requirements
- Uninstall old versions of Altova server products
- Download the Linux package
- Install MobileTogether Server
- Altova LicenseServer
- LicenseServer versions

**Licensing on Linux**
- Start LicenseServer
- Start MobileTogether Server
- Register MobileTogether Server
- Assign a license

**Notes about Environment**
3.2.1 Installation on Linux

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

- System requirements
  - Linux
    - CentOS 6 or newer
    - RedHat 6 or newer
    - Debian 8 or newer
    - Ubuntu 14.04 or newer

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

<table>
<thead>
<tr>
<th>Required by</th>
<th>CentOS, RedHat</th>
<th>Debian</th>
<th>Ubuntu</th>
</tr>
</thead>
<tbody>
<tr>
<td>LicenseServer</td>
<td>krb5-libs</td>
<td>libgssapi-krb5-2</td>
<td>libgssapi-krb5-2</td>
</tr>
<tr>
<td>MobileTogether</td>
<td>qt4, krb5-libs, qt-x11</td>
<td>libqtgui4, libgssapi-krb5-2</td>
<td>libqtgui4, libgssapi-krb5-2, libldap-2.4 (also see note)</td>
</tr>
<tr>
<td>Server Advanced</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edition</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** 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.

**Note:** On newly installed Ubuntu server 18.04.1 LTS, it was found that the universe repository needed to be enabled to apt (using the command: `sudo add-apt-repository universe`). Installation of `libqtgui4` was possible after this update.

- Uninstall old versions of Altova server products

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

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

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

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

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

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

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

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Package extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debian</td>
<td>.deb</td>
</tr>
<tr>
<td>Ubuntu</td>
<td>.deb</td>
</tr>
<tr>
<td>CentOS</td>
<td>.rpm</td>
</tr>
<tr>
<td>RedHat</td>
<td>.rpm</td>
</tr>
</tbody>
</table>

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-5.4-debian.deb`
- [Ubuntu]: `sudo dpkg --install mobiletogetherserver-5.4-ubuntu.deb`
- [CentOS]: `sudo rpm -ivh mobiletogetherserver-5.4-1.x86_64.rpm`
- [RedHat]: `sudo rpm -ivh mobiletogetherserver-5.4-1.x86_64.rpm`

The MobileTogether Server package will be installed in the folder:

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

Altova LicenseServer

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

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

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

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

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

Current version: 3.2
Licensing on Linux

MobileTogether Server 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.

The steps to carry out are given below in brief. For detailed information, see the LicenseServer user manual at the Altova website.

**Start LicenseServer**

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

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

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

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

**Start MobileTogether Server**

Start MobileTogether Server as a daemon with the following command:

- `< Debian 8>`
  - `sudo /etc/init.d/mobiletogetherserver start`
  - `sudo systemctl start mobiletogetherserver`
- `< CentOS 7>`
  - `sudo initctl start mobiletogetherserver`
  - `sudo systemctl start mobiletogetherserver`
- `< Ubuntu 15>`
  - `sudo initctl start mobiletogetherserver`
  - `sudo systemctl start mobiletogetherserver`
- `[RedHat]`
  - `sudo initctl start mobiletogetherserver`
To set up and configure MobileTogether Server, open its Web UI (Setup) page by entering the URL of the Web UI page in the address bar of an Internet browser: http://<serverIPAddressOrName>:8085.

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

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

```
sudo /opt/Altova/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. Essentially: (i) Start MobileTogether Server via ServiceController (see previous point); (ii) Enter your password to access the Setup page; (iii) Select the LicenseServer name or address, and click Register with LicenseServer.

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

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

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

If you are using a computer server with a large number of CPU cores but only have a low volume to process, you may also create a virtual machine that is allocated a smaller number of cores, and purchase a license for that number. Such a deployment, of course, will have less processing speed than if all available cores on the server were utilized.
Note: Each Altova server product license can be used for only one client machine—the machine on which the Altova server product is installed—at a time, even if the license has unused licensing capacity. For example, if a 10-core license is used for a client machine that has 6 CPU cores, then the remaining 4 cores of licensing capacity cannot be used simultaneously for another client machine.

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

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

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’s Solutions Working Directory**. 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 will need to be adjusted to match your
environment:

Shell script (unit file)

```bash
#- 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:
export JAVA_HOME=/usr/local/jdk1.7.0_17
export CLASSPATH=/usr/local/jdbc/oracle/ojdbc6.jar:/usr/local/jdbc/oracle/
odb.jar:/usr/local/jdbc/oracle/xmlparserv2.jar:/usr/local/jdbc/postgresql/
jdbc/iseries/lib/jt400.jar:/usr/local/jdbc/mysql/mysql-connector-java-
5.1.16-bin.jar:/usr/local/jdbc/sqlite/sqlitejdbc-v056.jar:/usr/local/jdbc/
Informix_JDBC_Driver/lib/ifxjdbc.jar:/usr/local/jdbc/sybase/jconn7/
jconn4.jar:/usr/local/jdbc/db2/db2jcc.jar:/usr/local/jdbc/db2/
db2jcc_license_cu.jar:/:;
```
3.3 Setup on macOS

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

Installation on macOS

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

Licensing on macOS

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

Notes about Environment
3.3.1 **Installation on macOS**

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

- **System requirements**
  - **macOS**
    - macOS 10.12 or newer
    - Java for macOS (latest version)

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

- **Uninstall old versions of Altova server products**
  Before uninstalling MobileTogether Server, stop the service with the following command:
  ```bash```
  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 command:
  ```bash```
  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:
  ```bash```
  sudo launchctl unload /Library/LaunchDaemons/com.altova.LicenseServer.plist
  ```
  To check whether the service has been stopped, open the Activity Monitor in Finder and make sure that LicenseServer is not in the list. Then proceed to uninstall in the same way as described above for MobileTogether Server.

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

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

The MobileTogether Server package will be installed in the folder:

```
/usr/local/Altova/MobileTogetherServer (application binaries)
/var/Altova/MobileTogetherServer (data files: database and logs)
```

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

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

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

### Altova LicenseServer

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

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

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

The LicenseServer package will be installed in the folder:

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

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

### LicenseServer versions

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

_Current version: 3.2_
3.3.2 Licensing on macOS

MobileTogether Server 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.

The steps to carry out are given below in brief. For detailed information, see the LicenseServer user manual at the Altova website.

**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:

```bash
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:

```bash
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:

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

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

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

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

- Double-click the MobileTogether Server 5.4 icon in the Applications folder of the Finder
- Enter the URL of the Web UI page in the address bar of an Internet browser:
  ```plaintext
  http://<serverIPAddressOrName>:8085
  ```

**Firewall Note**

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

**Register MobileTogether Server**

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

```
sudo /usr/local/Altova/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 via ServiceController; (ii) Enter your password to access the Setup page; (iii) Select the LicenseServer name or address, and click Register with LicenseServer.

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

**Assign a license**

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

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

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

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

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

*MobileTogether Server licenses*

MobileTogether Server licenses are based on the number of CPU cores on the MobileTogether Server machine. Core licenses allow an unlimited number of MobileTogether Server
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.
### 3.3.3 Notes about Environment

**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's Solutions Working Directory*. 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
Setting Up MobileTogether Server

○ **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:

```
<?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/MobileTogetherServer5.4/bin/MobileTogetherServer</string>
      <string>debug</string>
    </array>
    <key>KeepAlive</key>
    <true/>
    <key>UserName</key>
    <string>_altovamobiletogetherserver</string>
    <key>EnvironmentVariables</key>
    <dict>
      <key>CLASSPATH</key>
```
```
3.4 Setting Up SSL Encryption

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

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

The steps to do this are listed below.

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

1. Generate a private key

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

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

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

2. Certificate Signing Requests (CSRs)

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

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

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

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

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

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

Allow time for obtaining a certificate

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

4. Receive public key from CA

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

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

5. Save public key to file

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

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

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

6. Save CA’s intermediate certificates to file

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

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

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

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

11. Enable MobileTogether Clients to use SSL
In MobileTogether Client apps that communicate with an SSL-enabled MobileTogether Server, enable SSL communication by checking the SSL Encryption checkbox. See the MobileTogether Client documentation for information about how to find this check box.
Chapter 4

Server Procedures
4 Server Procedures

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

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

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 Server 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.

![Screenshot of ServiceController]

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

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

▼ On Linux

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

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

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

On macOS

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

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

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

  sudo launchctl unload /Library/LaunchDaemons/
  com.altova.LicenseServer.plist
4.2 Start MobileTogether Server

In order to run MobileTogether Server, MobileTogether Server must be started as a service. 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.

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

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

**On Linux**

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

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

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

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

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

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

```bash
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:

```plaintext
--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.

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

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

8. Install SSL certificate on MobileTogether Server

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

• private.key: Contains the private key certificate
• publickey.cer: Contains the public key certificate and the CA's intermediate certificates (primary and secondary)
To install the SSL certificates on MobileTogether Server, do the following:

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

```
SSL certificates:

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

Private Key:
Browse... No file selected.

Certificate:
Browse... No file selected.

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.

Let’s Encrypt Certificates
```

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

10. Test SSL communication

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

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

11. Enable MobileTogether Clients to use SSL

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

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

Set the administrator ports

The administrator ports provide access for the following purposes:

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

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

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.

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

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

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

<table>
<thead>
<tr>
<th>Linux</th>
<th>/var/opt/Altova/MobileTogetherServer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mac</td>
<td>/var/Altova/MobileTogetherServer</td>
</tr>
<tr>
<td>Windows 7, 8, 10</td>
<td>C:\ProgramData\Altova\MobileTogetherServer</td>
</tr>
</tbody>
</table>

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.

```html
<!DOCTYPE html>
<html>
```

© 2019 Altova GmbH
<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>
<tr>
<td>
<label for="username">Username:</label>
</td>
<td>
<input type="text" name="username" id="username" size="30"/>
</td>
</tr>
<tr>
<td>
<label for="password">Password:</label>
</td>
<td>
<input type="password" name="password" id="password" size="30"/>
</td>
</tr>
</table>
<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>
</form>
</div>
</body>
<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>
</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 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.

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:

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>root</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>anonymous</strong></td>
<td>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.</td>
</tr>
</tbody>
</table>

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 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* tab.

<table>
<thead>
<tr>
<th>Privileges</th>
</tr>
</thead>
<tbody>
<tr>
<td>🚀 Maintain users, roles and privileges</td>
</tr>
<tr>
<td>✔️ Set own password</td>
</tr>
<tr>
<td>✔️ Override security</td>
</tr>
<tr>
<td>✔️ Allow to use stored password on client (do not require authentication on application start)</td>
</tr>
<tr>
<td>✔️ View unfiltered log</td>
</tr>
<tr>
<td>✔️ View cache overview</td>
</tr>
<tr>
<td>✔️ View user licenses overview</td>
</tr>
<tr>
<td>✔️ Read users and roles</td>
</tr>
<tr>
<td>✔️ Manage server settings</td>
</tr>
<tr>
<td>☑️ Trace workflow (Enables detailed workflow execution logging to files (including working XML files) when the “Logging to File” option is enabled)</td>
</tr>
<tr>
<td>✔️ Read statistics (Enables reading server statistics)</td>
</tr>
<tr>
<td>✔️ Read database structures</td>
</tr>
<tr>
<td>✔️ Read global resources</td>
</tr>
<tr>
<td>✔️ Write global resources</td>
</tr>
<tr>
<td>✔️ Open workflow from designer</td>
</tr>
<tr>
<td>✔️ Save workflow from designer</td>
</tr>
<tr>
<td>✔️ Run server simulation</td>
</tr>
</tbody>
</table>

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.

- **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. See the description of the Statistics setting 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 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 | Save to MobileTogether server.

- **Run server simulation**
  Allows the user having this privilege to run a simulation from the browser (and preview the result). Note the Back browser button returns you to the container view.
4.7 Configure the Firewall

Server IP address and network firewall settings

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

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

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

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

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

A server service is a set of MobileTogether actions that is deployed to MobileTogether Server Advanced Edition as a solution (.mtd file). 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.

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).

You can create the following types of triggers:

- **Timer triggers**, 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**, which enable you to trigger a service by checking for changes to a file or directory on the server.
• **HTTP triggers**, 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.

<table>
<thead>
<tr>
<th>Trigger</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Trigger now]</td>
<td>Runs the service immediately. Disabled for filesystem triggers: redundancy because any file/dir change would trigger the service.</td>
</tr>
<tr>
<td>Duplicate trigger.</td>
<td></td>
</tr>
<tr>
<td>Delete trigger.</td>
<td></td>
</tr>
<tr>
<td>Undo a previous delete action.</td>
<td></td>
</tr>
</tbody>
</table>

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**.

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 illustrates how to define the settings of a timer trigger.

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 illustrates how to define the settings of a file system trigger.

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 illustrates how to define the settings of a HTTP trigger.

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.

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 5.4).
2. Access the MobileTogether Server administrator interface in a web browser by typing this URL: http://<serverIPAddressOrName>:8085/.
3. Enter your login information and go to the Workflows tab.
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, go to the Settings tab and, in the Statistics pane of the Misc tab, 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, 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**

<table>
<thead>
<tr>
<th>Platform</th>
<th>Folder Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>/var/opt/Altova/MobileTogetherServer</td>
</tr>
<tr>
<td>Mac</td>
<td>/var/Altova/MobileTogetherServer</td>
</tr>
<tr>
<td>Windows 7, 8, 10</td>
<td>C:\ProgramData\Altova\MobileTogetherServer</td>
</tr>
</tbody>
</table>

**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**).

![Statistics solution interface](image)

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 server 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 server 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.

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

**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:

<table>
<thead>
<tr>
<th></th>
<th>IE 8.0 +</th>
<th>Firefox 3.5+</th>
<th>Safari 4.0+</th>
<th>Chrome 4.0+</th>
<th>Opera 10.5+</th>
<th>iPhone 2.0+</th>
<th>Android 2.0+</th>
</tr>
</thead>
</table>

**Updating server settings on client devices**

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

These procedures are described below.

Backing up MobileTogether Server

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

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

<table>
<thead>
<tr>
<th>OS</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Linux</strong></td>
<td>/var/opt/Altova/MobileTogetherServer</td>
</tr>
<tr>
<td><strong>Mac</strong></td>
<td>/var/Altova/MobileTogetherServer</td>
</tr>
<tr>
<td><strong>Windows 7, 8, 10</strong></td>
<td>C:\ProgramData\Altova\MobileTogetherServer</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>File/Folder</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cache</td>
<td>Default directory for caches of solutions. If a cache is not available, it will be recreated automatically at runtime.</td>
</tr>
<tr>
<td>logs</td>
<td>Default directory for log files that are created when the Logging to file option is enabled and for general MobileTogether Server logs.</td>
</tr>
<tr>
<td>SolutionFiles</td>
<td>Default directory for XML or image files referenced from deployed solutions.</td>
</tr>
<tr>
<td>cert.pem</td>
<td>PEM file with certificate needed for secure (SSL) communication.</td>
</tr>
<tr>
<td>key.pem</td>
<td>PEM file with private key needed for secure (SSL) communication.</td>
</tr>
<tr>
<td>mobiletogether.db</td>
<td>This is the main database file (SQLite) that stores the MobileTogether Server object system, user data, deployed solutions, files, and others.</td>
</tr>
<tr>
<td>mobiletogetherlog.db</td>
<td>This is the database file (SQLite) that stores the MobileTogether Server logs.</td>
</tr>
<tr>
<td>mobiletogetherserver.cfg</td>
<td>Stores global configuration settings of MobileTogether Server (port number, solutions directory, etc)</td>
</tr>
</tbody>
</table>
### Server Procedures

#### How to Back Up and Restore MobileTogether Server

<table>
<thead>
<tr>
<th>mobiletogetherserver.licsid</th>
<th>File with ID of registered <strong>LicenceServer</strong> client.</th>
</tr>
</thead>
<tbody>
<tr>
<td>mobiletogetherserver.licsvr</td>
<td>Contains the address of the LicenseServer and failover if one is specified.</td>
</tr>
</tbody>
</table>

**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 tab.

### Restoring MobileTogether Server

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

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

### Updating client connections to the server

If you have moved MobileTogether Server to another machine (with new settings, such as a new IP address), client device settings to connect to MobileTogether Server must be updated. See Information for Clients for details.
4.12 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.
Chapter 5

Web UI Reference
5 Web UI Reference

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

Accessing the Web UI of MobileTogether Server

The MobileTogether Server Web UI is accessed as follows:

▼ On Windows

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

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

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 via one or more domains has been defined, then a Login combo box is available in which you can: (i) select from among the defined domains, or (ii) choose to login directly (not via a domain).
On Mac OS X

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

http://<serverIPAddressOrName>:8085/

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

Web UI tabs

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

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

The Workflows tab  (screenshot below) provides an interface for managing the container structure of the root folder of MobileTogether Server and the access rights (permissions) for each container. Containers are folders that contain sub-containers and/or solutions (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.

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:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Container</td>
<td>Creates a container in the current folder. Click a container to open it</td>
</tr>
<tr>
<td>Save</td>
<td>Saves changes such as a modification of descriptions</td>
</tr>
<tr>
<td>Move or Rename Selected Objects</td>
<td>If one object is selected, opens the Rename and Move dialog, in which you can (i) rename and/or (ii) move the object to a new container that you select. If multiple objects are selected, opens the Move dialog, in which you can select the container to move to.</td>
</tr>
<tr>
<td>Delete Selected Objects</td>
<td>Deletes the selected container or file</td>
</tr>
<tr>
<td>Lock Selected</td>
<td>A locked solution cannot be overwritten by a new deployment. If attempted, an error message is displayed in MobileTogether Designer</td>
</tr>
</tbody>
</table>
Unlock Selected | Unlocks a locked solution
---|---
Permissions | Sets which users/roles can access individual containers, and their level of access
Search | Searches for the term submitted. Check Recursive to search in descendant containers

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

The container `/public/`

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

The display of a container's contents

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

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

• **Run in Browser:** The server URL where the solution file is deployed. Click to run the solution. If the solution defines server services, click the **Service config** button in this column to access the service's configuration interface. (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 page is displayed that shows the automated tests of that solution (*screenshot below*).

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

1. In the **Active** column, check the test runs that you want to make active. These test runs will be played back on the client when the user starts a solution. If multiple test runs are selected, then all the selected test runs will be played back. If any one of a solution's test runs has been activated, then, on the Workflows page, the wheel in the design's **Automated Test** column is displayed in red.
2. Set the 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).
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.
Rules for inheritance of permissions

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

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

**Container**

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

**Workflow**

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

**Security**

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

**Editing the permissions of a container**

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

   ![Permissions for public/contacts](image1)

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](image2)

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

4. Change the permissions as required. The Inherit option causes permissions to be inherited from the parent container.
5.2 Users and Roles

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

### Administration: Users

<table>
<thead>
<tr>
<th>Workflows</th>
<th>Users and Roles</th>
<th>User licenses</th>
<th>Log</th>
<th>Cache</th>
<th>Settings</th>
</tr>
</thead>
</table>

#### Users

- **TechWriter-01**
- **anonymous**
- **root**

- **Create User**
- **Import Domain Users**
- **Delete Selected 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:

<table>
<thead>
<tr>
<th>root</th>
<th>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.</th>
</tr>
</thead>
<tbody>
<tr>
<td>anonymous</td>
<td>anonymous is an account for anonymous users that access services exposed via the HTTP service interface. It cannot be used for logging in</td>
</tr>
</tbody>
</table>
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.

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

About Roles

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

The following roles are predefined:

- **authenticated** is automatically assigned to every user except anonymous. So 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.

![Users Table]

**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:

<table>
<thead>
<tr>
<th>User</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>root</td>
<td>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.</td>
</tr>
<tr>
<td>anonymous</td>
<td>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.</td>
</tr>
</tbody>
</table>

**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.

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

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

The following roles are predefined:
- authenticated is automatically assigned to every user except anonymous. So 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.

![Create User Page](image)

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

### Importing a domain user

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

```
 | Roles available | Roles assigned to the user 'Tech-01'
 | Name  | All | Authenticated
 | John  | Yes | Yes
 | Smith | No  | No

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

Deleting a user

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>TechWriter.01</td>
<td>checked</td>
</tr>
<tr>
<td>anonymous</td>
<td>unchecked</td>
</tr>
<tr>
<td>root</td>
<td>unchecked</td>
</tr>
</tbody>
</table>

[Create User]  [Delete Selected Users]
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.

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.
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.

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 |**
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).

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

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

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

- Importing a domain role

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

![Roles](image)

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.

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

Creating a password policy

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

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

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

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.

<table>
<thead>
<tr>
<th>Privilege</th>
<th>Principal</th>
<th>Granted to and/or inherited from Principals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allow to use stored password on client</td>
<td><code>root</code></td>
<td>granted to <code>root</code></td>
</tr>
<tr>
<td></td>
<td><code>workflow-user</code></td>
<td>granted to <code>workflow-user</code></td>
</tr>
<tr>
<td>Maintain users, roles and privileges</td>
<td><code>TechWriter-01</code></td>
<td>granted to <code>TechWriter-01</code></td>
</tr>
<tr>
<td></td>
<td><code>root</code></td>
<td>granted to <code>root</code></td>
</tr>
<tr>
<td>Manage server settings</td>
<td><code>TechWriter-01</code></td>
<td>granted to <code>TechWriter-01</code></td>
</tr>
<tr>
<td></td>
<td><code>root</code></td>
<td>granted to <code>root</code></td>
</tr>
<tr>
<td>Open workflow from designer</td>
<td><code>Deploy</code></td>
<td>inherited from <code>workflow-designer</code></td>
</tr>
<tr>
<td></td>
<td><code>TechWriter-01</code></td>
<td>inherited from <code>workflow-designer</code></td>
</tr>
<tr>
<td></td>
<td><code>root</code></td>
<td>granted to <code>root</code></td>
</tr>
<tr>
<td></td>
<td><code>workflow-designer</code></td>
<td>granted to <code>workflow-designer</code></td>
</tr>
</tbody>
</table>

- 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.

<table>
<thead>
<tr>
<th>Principal</th>
<th>Privilege</th>
<th>Granted to and/or inherited from Principals</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Deploy</code></td>
<td>Open workflow from designer</td>
<td>inherited from <code>workflow-designer</code></td>
</tr>
<tr>
<td></td>
<td>Read global resources</td>
<td>inherited from <code>workflow-designer</code></td>
</tr>
<tr>
<td></td>
<td>Run server simulation</td>
<td>inherited from <code>workflow-designer</code></td>
</tr>
<tr>
<td></td>
<td>Save workflow from designer</td>
<td>inherited from <code>workflow-designer</code></td>
</tr>
<tr>
<td></td>
<td>Write global resources</td>
<td>inherited from <code>workflow-designer</code></td>
</tr>
<tr>
<td><code>TechWriter-01</code></td>
<td>Maintain users, roles and privileges</td>
<td>granted to <code>TechWriter-01</code></td>
</tr>
<tr>
<td></td>
<td>Manage server settings</td>
<td>granted to <code>TechWriter-01</code></td>
</tr>
<tr>
<td></td>
<td>Open workflow from designer</td>
<td>inherited from <code>workflow-designer</code></td>
</tr>
<tr>
<td></td>
<td>Read global resources</td>
<td>inherited from <code>workflow-designer</code></td>
</tr>
</tbody>
</table>
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.

- 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.
- Client Version: The version of the MobileTogether Client app on the client device. Knowing
the client version can be important for debugging and troubleshooting errors that might occur on the client device.

- **Active**: The Active check box is used to activate/deactivate a license. Click **Save** to finalize the change.
- **Request Time, Activation Time**: The times when a license is requested and activated, respectively.

### Search

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

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

The Log tab shows the logged actions—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 In the Settings | Logging 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).

The view can be filtered by:

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

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

Searching for log messages

To access the Search dialog (screenshot below), click the Search button at the top or bottom left of the log table (see screenshot above).
For each search rule, select a field (such as Date or User), an operator (such as contains or equals), and the value to search for. Add a new search rule by clicking the Add Rule button. Delete a search rule by clicking its Delete Rule button. The all selector at the top specifies that the search condition is fulfilled only when all the search rules are individually fulfilled. The any selector specifies that the search condition is fulfilled if any one search rule is fulfilled. Click Find to start the search. Click Reset to remove the search filter.
5.5 Cache

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 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.)

**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.
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, (iii) the statistics database, and (iv) 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.

The following backup settings can be configured:

- The folder on the server that will contain the backed up files.
- 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. 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.

<table>
<thead>
<tr>
<th>Backup Date/Time</th>
<th>Server Config File</th>
<th>Main Database</th>
<th>Log Database</th>
<th>Stats Database</th>
<th>Solution Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>2018-07-13 14:09:00</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2018-07-13 14:06:00</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
</tbody>
</table>

To restore a particular backup, click that backup's **Restore** button. Only the checked (backed up) files (*see screenshot above*) will be restored. 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.

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

- **Network**: Settings for mobile client ports, administrator ports, and SSL certificates
- **Logging**: Logging settings
- **LDAP**: Settings for Directory service login
- **JWT**: JSON Web Token (JWT) authentication settings
- **Cache**: cache settings
- **Misc**: Settings for server statistics, server simulations, workflow execution on server, session timeouts, backups on upgrade, and email sending
- **Sources**: Server-side data folder location, and management of server-side database connections
- **LicenseServer**: Registration and licensing with Altova LicenseServer
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**. 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.

  ![Mobile client ports](image)

- **Automatically login as anonymous**

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

- **Use customized login and index pages**

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

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

```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>
                <!-- The Active Directory domain details -->
                <h2>Active Directory Login:</h2>
                <table>
                    <tr>
                        <td>
                            <label for="providernameprefix">Domain prefix:</label>
                        </td>
                    </tr>
                </table>
            </table>
        </form>
    </div>
</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 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.
- 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.

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

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.

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

Logs contain reports of workflow activity, and they are displayed in the Log tab of the Web UI. The settings in this tab define logging parameters. If you modify any setting, click Save at the bottom of the tab for the modified setting to take effect.

- **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.
5.7.3 LDAP

If **Directory Service Login** is enabled, enables users to 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.

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.

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

**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.

<table>
<thead>
<tr>
<th>Directory service login:</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Directory service login" /></td>
</tr>
</tbody>
</table>

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**. Then go to the **Users and Roles | Users** tab to import specific users as MobileTogether Server users. An allowed domain user can then be assigned **roles or privileges in the usual way**. After these settings have been made, these 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 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.
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](#).

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.
5.7.5 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** 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 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.

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.

▼ Serve-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 corresponding privilege, Read database structures, has been enabled for the current user.

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

• The **Change Server** button enables you to select the MobileTogether Server on which the DB connections are to be saved.
5.7.7 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.

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 privilege must be checked for that user.

Location of the MobileTogether Server AppData folder on various operating systems

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>/var/opt/Altova/MobileTogetherServer</td>
</tr>
<tr>
<td>Mac</td>
<td>/var/Altova/MobileTogetherServer</td>
</tr>
</tbody>
</table>
Windows 7, 8, 10
C:\ProgramData\Altova\MobileTogetherServer

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

- **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 when a new MobileTogether Server version is installed. When you de-install an existing MobileTogether Server installation, these MobileTogether Server files and folders are still held in the system. Subsequently, when a new MobileTogether Server package is installed, this data is copied into a backup folder that is created in the MobileTogether Server application folder.

Upgrade settings:

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

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

▼ 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).

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

MobileTogether Server must be registered with an Altova LicenseServer on your network. The LicenseServer settings specify the LicenseServer machine to connect to, and enables you to register MobileTogether Server with LicenseServer. See the section, Setting Up MobileTogether Server for information about licensing. If you modify any setting, click Save (at the bottom of the tab) for the modified setting to take effect.

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

To enter a server address, click the Address button, and enter the server address.

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

This topic:

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

Default location of MobileTogether Server executable

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

- **Windows**
  <ProgramFilesFolder>\Altova\MobileTogetherServer\bin\MobileTogetherServer.exe

- **Linux**
  /opt/Altova/MobileTogetherServer/bin/mobiletogetherserver

- **Mac**
  /usr/local/Altova/MobileTogetherServer/bin/mobiletogetherserver

Usage and list of CLI commands

General command line syntax for `MobileTogetherServer` is:

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

Casing and slashes on the command line

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

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

where

<table>
<thead>
<tr>
<th>--h</th>
<th>--help</th>
<th>Displays the help text.</th>
</tr>
</thead>
<tbody>
<tr>
<td>--version</td>
<td>Displays the version of MobileTogether Server.</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>licenseserver</th>
<th>Register MobileTogether Server with LicenseServer on the local network.</th>
</tr>
</thead>
<tbody>
<tr>
<td>assignlicense</td>
<td>Upload a license to LicenseServer and assign it to MobileTogether Server on this machine.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>verifylicense</code></td>
<td>Check if the current MobileTogether Server is licensed; optionally, check if a given license key is assigned.</td>
</tr>
<tr>
<td><code>install</code></td>
<td>Installs MobileTogether Server as a service.</td>
</tr>
<tr>
<td><code>start</code></td>
<td>Starts MobileTogether Server as a service.</td>
</tr>
<tr>
<td><code>uninstall</code></td>
<td>Uninstalls MobileTogether Server as a service.</td>
</tr>
<tr>
<td><code>debug</code></td>
<td>Starts MobileTogether Server for debugging (not as a service).</td>
</tr>
<tr>
<td><code>resetpassword</code></td>
<td>Resets the password of the root user to the default value.</td>
</tr>
<tr>
<td><code>upgradedb</code></td>
<td>Upgrades the internal MobileTogether Server DB to the structure of the new version, and enters correct default values where necessary.</td>
</tr>
<tr>
<td><code>exportresourcestrings</code></td>
<td>Exports all application resource strings to an XML file.</td>
</tr>
<tr>
<td>`setdeflang</td>
<td>sdl`</td>
</tr>
<tr>
<td><code>help</code></td>
<td>Displays help for a specific command. For example: <code>help generate</code></td>
</tr>
<tr>
<td><code>version</code></td>
<td>Displays the version and exits.</td>
</tr>
</tbody>
</table>
6.1 assignlicense (Windows only)

On execution, the assignlicense command uploads the license file specified by the FILE argument to the registered LicenseServer, and assigns the license to MobileTogether Server on this machine. The FILE argument takes the filepath of the license file. The --test-only option allows you to upload to LicenseServer and validate the license, without assigning the license to MobileTogether Server. For details about licensing, see the LicenseServer documentation (https://www.altova.com/documentation).

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

Syntax

MobileTogetherServer assignlicense [options] FILE

Casing and slashes on the command line

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

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td>The path to the license file</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--t, --test-only</td>
<td>Values are true</td>
</tr>
<tr>
<td>--h, --help</td>
<td>Displays information about the command</td>
</tr>
</tbody>
</table>

Options are listed in their short forms (if available) and long forms. You can use one or two dashes for both short and long forms. An option that takes a value is written like this: --option=value.

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

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

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

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

Syntax

    mobiletogetherserver debug [options]

▼ Casing and slashes on the command line

    MobileTogetherServer on Windows
    mobiletogetherserver on Unix (Linux, Mac)

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--h, --help</td>
<td>Displays information about the command</td>
</tr>
</tbody>
</table>

Options are listed in their short forms (if available) and long forms. You can use one or two dashes for both short and long forms. An option that takes a value is written like this: --option=value.

Examples

Example of the debug command:

    mobiletogetherserver debug
6.3 exportresourcestrings

The **exportresourcestrings** command outputs an XML file containing the resource strings of the MobileTogether Server application. It takes two arguments:

- 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 path and name of the output XML file

How to create localizations is described [below](#).

**Syntax**

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

**Casing and slashes on the command line**

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

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LanguageCode</td>
<td>Specifies the language of resource strings in the exported XML file.</td>
</tr>
<tr>
<td></td>
<td>Supported languages are: <strong>en</strong>, <strong>de</strong>, <strong>es</strong>, <strong>fr</strong>, <strong>ja</strong></td>
</tr>
<tr>
<td>XMLOutputFile</td>
<td>Specifies the location and name of the exported XML file.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--h, --help</td>
<td>Displays information about the command</td>
</tr>
</tbody>
</table>

Options are listed in their short forms (if available) and long forms. You can use one or two dashes for both short and long forms. An option that takes a value is written like this: **--option=value**.

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

An example of the `exportresourcestrings` command:

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

This command creates a file called `Strings.xml` at `c:\` that contains all the resource strings of 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 four supported languages into the target language. The resource strings are the contents of the `<string>` elements in the XML file. Do not translate variables in curly brackets, such as `{option}` or `{product}`.
3. Contact Altova Support to generate a localized MobileTogether Server DLL file from your translated XML file.
4. After you receive your localized DLL file from Altova Support, save the DLL in the `C:\Program Files (x86)\Altova\<%APPFOLDER\bin` folder. Your DLL file will have a name of the form `MobileTogetherServer2019_lc.dll`. The `_lc` part of the name contains the language code. For example, in `MobileTogetherServer2019_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.
6.4 help

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

Syntax

```
mobiletogetherserver help [options] Command
```

▼ Casing and slashes on the command line

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

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>The name of the command for which you want help information</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--h, --help</td>
<td>Displays information about the command</td>
</tr>
</tbody>
</table>

Options are listed in their short forms (if available) and long forms. You can use one or two dashes for both short and long forms. An option that takes a value is written like this: `--option=value`. 

Examples

An example of the help command:

```
mobiletogetherserver help licenseserver
```

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

The --help option

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

```
mobiletogetherserver licenseserver --help
```
achieves the same result as does using the help command with an argument of licenseserver:

```
mobiletogetherserver help licenseserver
```

In both cases, help information about the licenseserver command is displayed.
6.5 install

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

Syntax

```
mobiletogetherserver install [options]
```

⚠️ Casing and slashes on the command line

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

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--h, --help</code></td>
<td>Displays information about the command</td>
</tr>
</tbody>
</table>

Options are listed in their short forms (if available) and long forms. You can use one or two dashes for both short and long forms. An option that takes a value is written like this: `--option=value`.

Examples

Example of the `install` command:

```
mobiletogetherserver install
```
6.6 licenseserver

On execution, the licenseserver command registers MobileTogether Server with the LicenseServer specified by the Server-Or-IP-Address argument. For the licenseserver command to be executed successfully, the two servers must be connected on the network and LicenseServer must be running. You must have administrator privileges to be able to register MobileTogether Server with LicenseServer.

Once MobileTogether Server has been successfully registered with LicenseServer, you will receive a message to this effect. The message will also display the URL of the LicenseServer. You can now go to LicenseServer to assign MobileTogether Server a license. For details about licensing, see the LicenseServer documentation (https://www.altova.com/documentation).

Syntax

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

Casing and slashes on the command line

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

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server-Or-IP-Address</td>
<td>Identifies the machine on the network on which Altova LicenseServer is installed and running. It can be the machine's name or its IP address.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-- j, --json</td>
<td>Values are true</td>
</tr>
<tr>
<td>--h, --help</td>
<td>Displays information about the command</td>
</tr>
</tbody>
</table>

Options are listed in their short forms (if available) and long forms. You can use one or two dashes for both short and long forms. An option that takes a value is written like this: **--option=value**.

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.
6.7 resetpassword

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.

Syntax

```
mobiletogetherserver resetpassword [options]
```

▼ Casing and slashes on the command line

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

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--datadir</td>
<td>Specifies the path of the database directory</td>
</tr>
<tr>
<td>--h, --help</td>
<td>Displays information about the command</td>
</tr>
</tbody>
</table>

Options are listed in their short forms (if available) and long forms. You can use one or two dashes for both short and long forms. An option that takes a value is written like this: `--option=value`.

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

Example

Examples of the resetpassword command:

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

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

Syntax

```
mobiletogetherserver setdeflang | sdl [options] LanguageCode
```

Casing and slashes on the command line

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

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LanguageCode</td>
<td>sets the default language of MobileTogether Server. Supported languages are: en, de, es, fr, ja</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--h, --help</td>
<td>Displays information about the command</td>
</tr>
</tbody>
</table>

Options are listed in their short forms (if available) and long forms. You can use one or two dashes for both short and long forms. An option that takes a value is written like this: `--option=value`.

Examples

An example of the `setdeflang` command:

```
mobiletogetherserver setdeflang DE
```

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

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

Note: If MobileTogether Server is not installed as a service, you can install it with the install command, which is not the same as starting the service. To uninstall MobileTogether Server as a service, use the uninstall command.

Syntax

   mobiletogetherserver start [options]

▼ Casing and slashes on the command line

   MobileTogetherServer on Windows
   mobiletogetherserver on Unix (Linux, Mac)

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--h, --help</td>
<td>Displays information about the command</td>
</tr>
</tbody>
</table>

Options are listed in their short forms (if available) and long forms. You can use one or two dashes for both short and long forms. An option that takes a value is written like this: --option=value.

Examples

Examples of the start command:

   mobiletogetherserver start
6.10 uninstall

The `uninstall` command uninstalls MobileTogether Server as a service on the server machine. To re-install MobileTogether Server as a service, use the `install` command.

Syntax

```
mobiletogetherserver uninstall [options]
```

Casing and slashes on the command line

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

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

* Use forward slashes on Linux and Mac, backslashes on Windows.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--h, --help</code></td>
<td>Displays information about the command</td>
</tr>
</tbody>
</table>

Options are listed in their short forms (if available) and long forms. You can use one or two dashes for both short and long forms. An option that takes a value is written like this: `--option=value`.

Examples

Examples of the uninstall command:

```
mobiletogetherserver uninstall
```

Casing and slashes on the command line

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

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

* Use forward slashes on Linux and Mac, backslashes on Windows.
6.11 upgradedb

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.

Syntax

    mobiletogetherserver upgradedb [options]

▼ Casing and slashes on the command line

    MobileTogetherServer on Windows
    mobiletogetherserver on Unix (Linux, Mac)

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--h, --help</td>
<td>Displays information about the command</td>
</tr>
</tbody>
</table>

Options are listed in their short forms (if available) and long forms. You can use one or two dashes for both short and long forms. An option that takes a value is written like this: --option=value.

Examples

Examples of the upgradedb command:

    mobiletogetherserver upgradedb
6.12 verifylicense (Windows only)

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. This command is supported only on Windows systems. It is not supported on Linux or Mac systems. For details about licensing, see the LicenseServer documentation (https://www.altova.com/documentation).

Syntax

```
MobileTogetherServer verifylicense [options]
```

Casing and slashes on the command line

MobileTogetherServer on Windows
mobiletogetherserver on Unix (Linux, Mac)

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--l, license-key</td>
<td>The license key to check: whether it has been assigned to the product</td>
</tr>
<tr>
<td>--h, --help</td>
<td>Displays information about the command</td>
</tr>
</tbody>
</table>

Options are listed in their short forms (if available) and long forms. You can use one or two dashes for both short and long forms. An option that takes a value is written like this: `--option=value`.

Examples

Examples of the `verifylicense` command:

```
mobiletogetherserver verifylicense
mobiletogetherserver verifylicense --license-key=ABCDEFGHIJKLMNOPQRSTUVWXYZ12-3456789
```

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

The `version` command displays the version number of MobileTogether Server and exits.

Syntax

```
mobiletogetherserver version [options]
```

▼ Casing and slashes on the command line

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

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--h, --help</td>
<td>Displays information about the command</td>
</tr>
</tbody>
</table>

Options are listed in their short forms (if available) and long forms. You can use one or two dashes for both short and long forms. An option that takes a value is written like this: `--option=value`.

Examples

Examples of the `version` command:

```
mobiletogetherserver version
```
Index

A
Active directory login, 112
Address of server, 112
Administrator ports, 54, 112
Altova LicenseServer,
   connection settings, 112
   registering with, 112
   starting, 45

B
Back up MobileTogether Server, 75
Browsers,
   enabling execution of solutions for, 112

C
Cache settings, 112
Caches,
   creating, 108
   settings of, 108
Client ports, 112
Client users list, 104
Command line usage, 132
Commands,
   exportresourcestrings, 137
   help, 139
   licenseserver, 142
   setdeflang, 145

E
Encryption, 49
exportresourcestrings command, 137

F
File system trigger settings, 68

H
help command, 139
Host settings, 112
HTTP and HTTPS ports,
   for mobile clients, 112
   for server administrators, 112
HTTP trigger settings, 69

I
Installation on Linux, 21
Installation on macOS, 30
Installation on Windows, 14

L
LicenseServer,
   connection settings, 112
   registering with, 112
   see Altova LicenseServer, 45
licenseserver command, 142
Licensing on Linux, 24
Licensing on macOS, 33
Licensing on Windows, 16
Linux,
   installation on, 21
   licensing on, 24
Localization, 137, 145
Log of server actions, 106
Log settings, 112

M
macOS,
macOS,
  installation on, 30
  licensing on, 33
Mobile client ports, 54, 112
Mobile clients,
  information for, 74
MobileTogether Server, 3
  automatic shutdown of unlicensed servers, 47
  installing, 12
  licensing, 12
  setting up, 12
  starting, 47
  using, 9
MobileTogether Server overview, 7

P

Password policies,
  assigning members to, 101
  creating, 101
Passwords,
  enabling domains for, 112
Permissions, 82
Ports,
  for mobile clients, http and https, 112
  for server administrators, http and https, 112
Privileges, 58
  descriptive list of, 61

R

Reports,
  of privileges, 103
  of privileges by user, 103
Restoring MobileTogether Server, 75
Roles, 58
  assigning members to, 97
  creating, 97
  defining privileges for, 97

S

Server actions,
  log of, 106
Server address, 112
Server administrator ports, 112
Server folders,
  management of, 82
  structure of, 82
Server session timeouts, 112
Server side DB connections, 112
Server side solution’s working directory, 112
Server statistics, 70
Services,
  configuration overview, 65
  file system trigger for, 68
  HTTP trigger for, 69
  timer trigger for, 67
  trigger management, 65
setdeflang command, 145
Settings, 112
Setup,
  on Linux, 20
  on macOS, 29
  on Windows, 13
Simulation settings, 112
Solutions directory on server, 112
SSL certificates, 112
SSL encryption, 38, 49
Statistics,
  of solution usage, 70

T

Timer trigger settings, 67
Triggers for server services,
  file system triggers settings, 68
  HTTP trigger settings, 69
  management of, 65
  timer trigger, 65
  timer trigger settings, 67

U

Unlicensed server shutdown, 47
User licenses,
  administration of, 104
User login,
   and domain-specific passwords, 112
   importing user domains for, 112

Users, 58
   assigning roles to, 92
   creating new, 92
   deleting, 92
   managing, 92

W

Web browsers,
   enabling execution of solutions for, 112

Windows,
   installation on, 14
   licensing on, 16

Workflows, 82

Working directory, 112