

COMSOL Multiphysics

Model Manager Server Manual

Model Manager Server Manual

© 1998–2023 COMSOL

Protected by patents listed on www.comsol.com/patents, or see Help>About COMSOL Multiphysics on the File menu in the COMSOL Desktop for less detailed lists of U.S. Patents that may apply. Patents pending.

This Documentation and the Programs described herein are furnished under the COMSOL Software License Agreement (www.comsol.com/sla) and may be used or copied only under the terms of the license agreement.

COMSOL, the COMSOL logo, COMSOL Multiphysics, COMSOL Desktop, COMSOL Compiler, COMSOL Server, and LiveLink are either registered trademarks or trademarks of COMSOL AB. All other trademarks are the property of their respective owners, and COMSOL AB and its subsidiaries and products are not affiliated with, endorsed by, sponsored by, or supported by those trademark owners. For a list of such trademark owners, see www.comsol.com/trademarks.

Version: COMSOL 6.2

Contact Information

Visit the Contact COMSOL page at www.comsol.com/contact to submit general inquiries or search for an address and phone number. You can also visit the Worldwide Sales Offices page at www.comsol.com/contact/offices for address and contact information.

If you need to contact Support, an online request form is located on the COMSOL Access page at www.comsol.com/support/case. Other useful links include:

- Support Center: www.comsol.com/support
- Product Download: www.comsol.com/product-download
- Product Updates: www.comsol.com/product-update
- COMSOL Blog: www.comsol.com/blogs
- Discussion Forum: www.comsol.com/forum
- Events: www.comsol.com/events
- COMSOL Video Gallery: www.comsol.com/videos
- Support Knowledge Base: www.comsol.com/support/knowledgebase

Part number: CM020018

C o n t e n t s

Chapter 1: Introduction

About the Model Manager Server	8
What Can You Do with a Model Manager Server?	8
Where Can I Access the Documentation?	9
Overview of the Manual	11
Quick Start Guide	12

Chapter 2: Installation

Installing a Model Manager Server	16
Installation Planning	16
Installing in Windows	25
Installing in Linux and macOS	30
Modifying an Installation	31
Migrating From an Older Installation	32
Starting a Model Manager Server	34
Starting as a Windows Service	34
Starting Manually in Windows	35
Starting in Linux	35
Starting in macOS	37
Firewalls	37
Setting Up the Model Manager Server for Secure Connections	37
Running Behind a Reverse Proxy	38
Command Options	39
Security	40
Password Security	41

Chapter 3: Administration

Accessing the Web Interface	44
Overview of Web Pages	44
Current Database	45
Changing the Language.	45
Connectors	46
Adding a Connector With TLS	46
Common TLS Certificate Formats	49
The Connector Page	50
Login Configuration	52
Local Authentication	52
External Authentication	52
Proxy Authentication	57
Accounts	59
The My Account Page	59
The Accounts Page	60
Managed Server Components	62
Managed SQL Database Servers	62
Managed Resources Directories	68
Managed Search Index Servers.	71
Model Manager Server Databases	77
Database Configurations	77
Databases with Managed Server Components	81
Databases with External Server Components	82
Moving a Model Manager Server Database	87
Backup and Restore of a Model Manager Server Database	91
Backward Compatibility for Server Databases	93
Database Administration	94
Users	94
Groups	96

Database Permissions	97
Permission Templates	98
Asset Libraries.	100
Asset Types.	102
Primitive Attributes.	104
Composite Attributes	108

Chapter 4: Asset Management

Managing Simulation Projects Using Assets	114
Introduction to Asset Management	114
Adding a New Asset	116
The Asset Page	117
The Home Page	121
The Model Page	125
The File Page	128
Adding A New File	131
Customizing the Asset Management System	132
Adding New Types of Data Using Attributes	132
Adding New Types of Assets	135
Controlling Access to Assets Using Libraries	136
Searching on Asset Attributes	138
Full Text Search	138
Filters	139
Asset Search Syntax.	141
Example: A Project Asset Type	146
Defining Project Attributes	146
Adding the Project Asset Type	150
Adding Projects	151
Linking to Simulation Models in the Database	154
Opening Linked Simulation Models in the COMSOL Desktop	155
Viewing Older Versions	156
Finding Projects Using Attribute Filters	157

Introduction

Read this guide to learn how to install and administer a *Model Manager server*, a web server that hosts Model Manager databases. The Model Manager server can either be configured to use internal database components managed by the server, or external database components provided and managed by your organization. The guide also contains details on the asset management system included with a Model Manager server — a web-based tool you can use to manage simulation projects involving models built in the COMSOL Multiphysics® software. See the *Model Manager Reference Manual* for details on how to access a Model Manager server database from within the COMSOL Desktop® modeling environment.

In this chapter:

- [About the Model Manager Server](#)
- [Overview of the Manual](#)

About the Model Manager Server

In this section:

- [What Can You Do with a Model Manager Server?](#)
- [Where Can I Access the Documentation?](#)

What Can You Do with a Model Manager Server?

A Model Manager database can be shared by multiple users by hosting the database in a Model Manager server. You access the database by connecting to the Model Manager server from the COMSOL Multiphysics software via your organization's internal network. This enables you and your coworkers to collaborate on version-controlled simulation models and data files — all while staying within the COMSOL Desktop modeling environment. Read the *Model Manager Reference Manual* to learn more on what you can do with a Model Manager database accessed via a Model Manager server.

A Model Manager server database uses three separate server components:

- A *relational database management system* (RDBMS) for the version control management of simulation models and data files in a *SQL database*.
- A *resources directory* for storing binary and text data whose size is deemed too large to put directly inside a SQL database. This data can, for example, be built, computed, and plotted data generated by models, or auxiliary data such as CAD and interpolation data used as input by models.
- An *enterprise search platform* — the Apache Solr™ enterprise search platform — used for the search functionality of the Model Manager.

An installation of Model Manager server optionally includes the PostgreSQL® relational database management system and the Apache Solr™ enterprise search platform. Installing these *managed server components* enables you to configure a Model Manager database without installing any additional software. Such a database can be fully managed by the Model Manager server itself, including starting and stopping component subprocesses and handling data backups. A managed server database using default configuration settings is automatically created the first time the Model Manager server is started.

You can also configure a Model Manager server database using an external relational database management system provided by your organization — the supported database platforms are Microsoft® SQL Server®, MySQL®, Oracle® Database, and PostgreSQL®. Similarly, you can configure the database using an external Apache Solr™ installation.

From the Model Manager server web interface, administrators can configure the server to use secure connections using transport layer security (TLS) provided via HTTPS, set up authentication mechanisms so that users can log in via the COMSOL Multiphysics software or via a web browser, and perform other administrative tasks for the server.

A Model Manager server also comes with a web-based *asset management system* that enables simulation engineers working in COMSOL Multiphysics to easily collaborate on simulation projects with people in your organization who may not have access to the COMSOL Multiphysics software. Via the asset management system, users can link models and data files to various documents, presentations, project notes, slides, and other supplementary files and metadata on so-called *assets* — all while keeping everything in the same database that stores the models and data files. Users can also browse and search the version-controlled models and data files themselves, as well as upload new versions of data files. This enables, for example, engineers working with other tools and software to update data via the web interface so that this data is immediately available to simulation engineers working in the COMSOL Desktop. Or, these same simulation engineers can publish results in the form of animations, images, plots, reports and other output via the asset management system simply by exporting such output to the Model Manager server from the COMSOL Desktop.

Where Can I Access the Documentation?

A number of online resources have more information about COMSOL, including licensing and technical information. The electronic documentation, topic-based (or context-based) help, and the Application Libraries are all accessed through the COMSOL Desktop.



If you are reading the documentation as a PDF file on your computer, the [blue links](#) do not work to open an application or content referenced in a different guide. However, if you are using the Help system in COMSOL Multiphysics, these links work to open other modules, application examples, and documentation sets.

CONTACTING COMSOL BY EMAIL

For general product information, contact COMSOL at info@comsol.com.

COMSOL ACCESS AND TECHNICAL SUPPORT

To receive technical support from COMSOL for the COMSOL products, please contact your local COMSOL representative or send your questions to support@comsol.com. An automatic notification and a case number will be sent to you by email. You can also access technical support, software updates, license information, and other resources by registering for a COMSOL Access account.

COMSOL ONLINE RESOURCES

COMSOL website	www.comsol.com
Contact COMSOL	www.comsol.com/contact
COMSOL Access	www.comsol.com/access
Support Center	www.comsol.com/support
Product Download	www.comsol.com/product-download
Product Updates	www.comsol.com/product-update
COMSOL Blog	www.comsol.com/blogs
Discussion Forum	www.comsol.com/forum
Events	www.comsol.com/events
COMSOL Application Gallery	www.comsol.com/models
COMSOL Video Gallery	www.comsol.com/videos
Support Knowledge Base	www.comsol.com/support/knowledgebase

Overview of the Manual

This *Model Manager Server Manual* contains information that helps you install, configure, and administer a Model Manager server. It also contains information about the server's asset management system — a web-based tool you can use to manage and organize simulation projects involving models built in the COMSOL Multiphysics software. The information in this guide is specific to this functionality. Instructions on how to use the Model Manager in general are included with the *Model Manager Reference Manual*.

TABLE OF CONTENTS AND INDEX

To help you navigate through this guide, see the [Contents](#) section and [Index](#).

QUICKLY GETTING STARTED

Use the [Quick Start Guide](#) to get an overview of the necessary steps for getting started with a Model Manager server.

INSTALLING A MODEL MANAGER SERVER

The [Installation](#) chapter guides you through the installation and startup of a Model Manager server.

ADMINISTRATING A MODEL MANAGER SERVER

The [Administration](#) chapter contains details on how you configure and administer a Model Manager server via its web interface.

USING ASSETS TO MANAGE SIMULATION PROJECTS

In the [Asset Management](#) chapter, you will learn how assets can be used to connect simulation engineers working in COMSOL Multiphysics with people in your organization who may not have access to the COMSOL Multiphysics software.

Quick Start Guide

Getting started with a Model Manager server involves the following steps:

- 1 Install the Model Manager server software on a server computer in your organization's network — see [Installing a Model Manager Server](#). The section [Installation Planning](#) contains general guidelines and background information for server setups, data storage strategies, and handling of backups.
- 2 Ensure that the Model Manager server software is started automatically with the server computer for the highest availability — see [Starting a Model Manager Server](#).
- 3 Review the security guidelines for the Model Manager server software — see [Security](#). For example, unless the network is completely trusted, it is recommended to either place the Model Manager server behind a reverse proxy with HTTPS support or add a connector with TLS enabled — see below.
- 4 Log in to the Model Manager server in a web browser using the temporary password provided during installation — see [Accessing the Web Interface](#). Change your temporary password when prompted.
- 5 The Model Manager server is configured with default administrative settings. To review and update these settings, select **System** in the **Administration** menu (⚙️) to open the system administration area.
- 6 Set the authentication mechanism used when users log in. In the **System** navigation sidebar, under **Login**, you can choose between:
 - a **Accounts**. Manually create accounts for the users that should have access to the Model Manager server. The account passwords are stored locally in the Model Manager server [Preference Directory](#) — see [Local Authentication](#).
 - b **External Authentication**. Delegate the account authentication to an external authentication provider such as Windows® authentication or an LDAP server — see [External Authentication](#).
 - c **Proxy Authentication**. Delegate the account authentication to a trusted reverse proxy placed in front of the Model Manager server — see [Proxy Authentication](#).
- 7 Unless you have placed the Model Manager server behind a reverse proxy with HTTPS support, you are strongly recommended to configure the server to use secure connections. Under **Configuration>Connectors**, add a connector with TLS enabled — see [Connectors](#). You will need to have a TLS certificate on hand.

- 8 Decide if you want to use the default managed server components automatically set up for a Model Manager server, use custom managed server components, or use external server components. The latter can include, for example, a SQL database server administered by your organization.
 - a If you want to use the automatically set up managed server components, review them under **Managed Components** — see [Managed Server Components](#). Preferably update the backup directory locations of the default managed PostgreSQL[®] server and the default managed resources directory to locations on a different server disk — see [Moving a Model Manager Server Database](#). The new backup directory locations are used the next time the server is restarted.

A default Model Manager server database has been automatically configured with these managed server components — see [The Default Managed Database](#).
 - b If you want to use managed server components but leave the automatically set up default components unmodified as references, add new managed server components under **Managed Components** — see [Managed Server Components](#). Also add a new Model Manager server database under **Configuration>Databases** using these managed server components — see [Databases with Managed Server Components](#). Set the new database as the current default — see [The Database Configuration Page](#).
 - c If you want to use external server components, add a new Model Manager server database under **Configuration>Databases**. Configure a custom SQL database, custom resources directory, and custom search indexes — see [Databases with External Server Components](#). It is also possible to use a mixture of custom managed server components and external server components. Set the new database as the current default — see [The Database Configuration Page](#).
- 9 Launch COMSOL Multiphysics and verify that you can connect to the Model Manager server database from the COMSOL Desktop environment — select **File>Open From>Add Database>Connect to Server Database** and enter the address of the server and your account credentials. See [Connecting to a Server Database](#) in the *Model Manager Reference Manual*.

Installation

In this chapter you will learn how to install, start, and secure a Model Manager server. You are strongly recommended to read the general guidelines and background information for the server setup in the introductory section on planning a Model Manager server installation.

In this chapter:

- [Installing a Model Manager Server](#)
- [Starting a Model Manager Server](#)
- [Security](#)

Installing a Model Manager Server

A Model Manager server installation is similar to a COMSOL Multiphysics® or COMSOL Server™ installation. For detailed information on installation, license files, and license management, see the *COMSOL Multiphysics Installation Guide*.



Before starting a Model Manager server you need to start a license manager. The FlexNet® license manager can be installed together with a Model Manager server. A license server is not needed when running a trial license. For more information on the FlexNet® license manager, see the *COMSOL Multiphysics Installation Guide*. You can download the document from www.comsol.com/iog.

In this section:

- [Installation Planning](#)
- [Installing in Windows](#)
- [Installing in Linux and macOS](#)
- [Modifying an Installation](#)
- [Migrating From an Older Installation](#)

Installation Planning

A Model Manager server is typically installed on a server computer in your organization's internal network. Computers running the COMSOL Multiphysics software connect to the server computer over the network in order to access the Model Manager server database. Users can also access the Model Manager server web interface via a web browser — either to perform administrative tasks for the server or use the web-based asset management system.

This section contains general guidelines and background information to help you with planning your server setup. This includes a description of the various directories used by the server — the [Installation Directory](#) for the Model Manager server program, the [Preference Directory](#) for the Model Manager server settings and log files, and the [Data Directories](#) for the Model Manager server database — with particular emphasis on recommendations for access permissions, disk requirements, and backup strategies for

these directories. The section also contains details on [Server Processes and Memory Footprint](#) for the Model Manager server and some [Examples of Server Setups](#).

INSTALLATION DIRECTORY

The Model Manager server is supported on the Windows[®], Linux[®], and macOS operating systems. The default installation directory for each operating system is:

- Windows[®]: C:\Program Files\COMSOL\COMSOL62\ModelManagerServer
- Linux[®]: /usr/local/comsol62/modelmanagerserver
- macOS: /Applications/COMSOL62/ModelManagerServer

You can choose another directory when installing the server. The installation directory may either be on a physical disk or on a network disk.



It is recommended to not give the user account running the Model Manager server process any permissions other than *read* access to the installation directory.



The installation directory contains the program files for the Model Manager server and its managed server components. It does not contain any Model Manager server databases — see [Data Directories](#).

PREFERENCE DIRECTORY

Server settings and log files for a Model Manager server are stored in a *preference directory* with default location:

- Windows[®]: <user>\.comsol\v62modelmanagerserver
- Linux[®]: <user>/\.comsol/v62modelmanagerserver
- macOS: <user>/Library/Preferences/COMSOL/v62modelmanagerserver

with <user> being the home directory for the user account running the Model Manager server process. The location when running, for example, as a Windows[®] service using the predefined **NetworkService** user account is:

```
C:\Windows\ServiceProfiles\NetworkService\.comsol\v62modelmanagerserver
```

Replace **NetworkService** with **LocalService** in the directory path for the predefined **LocalService** account in Windows[®].

Two subfolders of particular interest in the preference directory are:

- `db` — containing accounts, database configurations, and other server preferences.
- `logs` — log files for the Model Manager server.



It is recommended that only the user account running the Model Manager server process is granted access permissions to the preference directory.

You can specify another directory path for the preference directory using the `-prefdir` command option — see [Command Options](#) for further details.



The Model Manager server preference directory must be located on a physical disk. Using a network disk is not supported and will result in the server being unable to start.

BACKUP OF PREFERENCE DIRECTORY

Add the subfolder `db` in the Model Manager server preference directory to your backup routines. The folder contains accounts, database configurations, and other server preferences that may be time consuming to recreate in case of a server disk crash or other failure. A periodic file system copy of the folder is sufficient.

When restoring a Model Manager server after a failure, copy the `db` folder from its backup location to its location in the preference directory before starting the server.

SERVER COMPONENTS

A Model Manager server uses three separate server components for the storage and search indexing of your data. A default installation of Model Manager server includes all three as managed server components, with the server handling the start and stop of component subprocesses and data backups. Once the Model Manager server is up and running, you can optionally replace some or all of these managed server components with external server components provided by your organization.

The Model Manager server uses a relational database management system for storing data in a SQL database. The installation includes the PostgreSQL[®] relational database management system as an optional managed server component. You can also configure the server to use an external relational database management system — see [External SQL Databases](#) for a list of supported SQL database servers. The external SQL

database server can either run on the same computer as the Model Manager server or on a different computer within the same network.

For the search functionality of the Model Manager, a Model Manager server uses the Apache Solr™ enterprise search platform. The installation includes this platform as an optional managed server component. You can also configure a Model Manager server to use an external Apache Solr™ installation provided by your organization — see [External Search Indexes](#). As for the SQL database server, the external Apache Solr™ server can either run on the same computer as the Model Manager server or on a different computer within the network.

Binary and text data that is too large to be stored inside the SQL database is stored directly on the file system inside a specific directory. This directory is registered with the Model Manager server as a server component. The backup of the files in the directory can either be managed by the Model Manager server or it can be handled by external backup routines provided by your organization.

The installation automatically adds a Model Manager database using managed server components configured with default settings — see also [The Default Managed Database](#). This includes, for example, data directories and backup directories located on the server computer’s file system. Depending on your expected user base and IT infrastructure, you may want to modify these default settings once the Model Manager server is up and running — either by updating the default database or by adding a new database.



When describing a Model Manager server and its server components there are inevitable terminology collisions for the words “database” and “server”. A Model Manager server uses a SQL database server as an independent server component. A Model Manager database uses a SQL database in such a SQL database server to store a *subset* of its data. A SQL database server can contain multiple SQL databases, each used by a different Model Manager database.



[Moving a Model Manager Server Database](#)

DATA DIRECTORIES

The three server components of a Model Manager database each uses its own data directory:

- The data directory for the SQL database. This is the main storage for the version control management of models and data files in repositories and branches, the version-controlled assets in asset libraries, and all user management and access control functionality. This data will inevitably grow over time, but, with a reasonably sized disk drive, it should still be fine to keep on the same computer as the Model Manager server installation. The size is also kept manageable by the fact that large binary and text data is stored *externally* to the SQL database, while the SQL database itself mainly stores descriptive, structural, and administrative *metadata* for this binary and text data.
- The data directory for the search index server. This data is automatically extracted from that which is stored in the SQL database and, as such, requires no backup. The size is typically much smaller than the other two data directories and there is generally no issue in placing it on the same computer as the Model Manager server installation.
- The so-called *resources directory* for storing large binary and text data associated with models, data files, and assets. Large here typically means above a few hundred kilobytes. This includes:
 - Binary data for geometries and meshes, computed solutions, and results plots associated with your models.
 - CAD data, interpolation functions, and other auxiliary data files used as input for your models.
 - Documentation files, presentations, videos, and other supplementary files linked to models via the asset management system.

The resources directory is expected to be the largest of the three and you may want to offload it to a dedicated file server whose file system can be mounted by the server computer running the Model Manager server — see [Examples of Server Setups](#).

The location of the data directories used by the default Model Manager server database added during installation depends on the computer account running the server — see [Default Data Directories](#).

When using managed server components, it is recommended to store the data directory for the SQL database server on a physical disk for best performance. For Linux, storing the data directory on a network disk using NFS mounted in hard mode

is supported. Other network disk configurations are not supported. Placing the data directory for the search index server on a network disk is not supported for any configuration.



Support for storing the data directories of managed server components on a network disk is as follows:

- SQL database server — only supported on Linux[®] using NFS mounted in hard mode.
- Search index server — never supported.
- Resources directory — always supported.

If placing the data directory of the SQL database server or the search index server on the same computer as the Model Manager server is not an option, the recommended approach is to use external server components. You can install a SQL database server on another computer and then connect to that server from the Model Manager server over the network. Connecting to an external SQL database server may also prove useful if the IT infrastructure of your organization already supports managing SQL databases, including handling their backups. You can also install and connect to an Apache Solr[™] enterprise search platform on another computer. Finally, the resources directory may be placed on an external file server as long as the Model Manager server can mount it as a network disk.



The computer account running the Model Manager server must have the necessary access permissions to connect to an external SQL database server, search index server, or file server. This is typically *not* the case for the predefined **LocalService** account in Windows[®].



Databases with External Server Components

Files stored in the resources data directory always pass through the Model Manager server when opened or saved from a COMSOL Multiphysics session — this to ensure data integrity and that necessary permissions are verified. There is therefore little to gain from placing a file server containing the resources data directory geographically near your users if the Model Manager server itself, for example, is not.

Default Data Directories

The location of the data directories used by the default Model Manager server database depends on the computer account set during installation. For the predefined **NetworkService** account in Windows[®], for example, the directories are located inside:

```
C:\Windows\ServiceProfiles\NetworkService\AppData\Local\COMSOL\
ModelManager\ManagedDatabases\default-managed
```

with subdirectories:

- **data** — the data directory of the managed SQL database server.
- **index** — the data directory of the managed search index server.
- **resources** — the managed resources directory.

Replace **NetworkService** with **LocalService** in the directory path for the predefined **LocalService** account in Windows[®].

For Linux[®], the directories are located inside:

```
<user>/ .comsolmodelmanagerdata/manageddatabases/default-managed
```

and for macOS:

```
<user>/Library/Application Support/COMSOL/ModelManager/
ManagedDatabases/default-managed
```



Moving a Model Manager Server Database

BACKUP OF DATA DIRECTORIES

A Model Manager server database using managed server components requires backup for the data directory of the SQL database server and for the resources data directory. The data directory of the search index server requires no backup. The default Model Manager database is already configured so that the Model Manager server handles this backup — see [Managed Backups](#) for further details.

You can also let external backup software provided by your organization handle the backup of the two data directories. This may be useful when you want to save disk space on the computer running the Model Manager server and mounting an external file server — as described for [Managed Backups](#) — is not feasible. The simplest option is to periodically shut down the Model Manager server and do a file system copy of

both directories to an external location. It is important to copy both directories as they could otherwise become inconsistent with respect to each other.



To get a consistent backup for this option, the Model Manager server must be shut down before copying the data directories. Copying the data manually via a system file explorer while the server is running — with different files captured at different points in time — will lead to data corruption.

If shutting down the server is impractical, it is also possible to do a backup while the service is running as long as the external backup software, as well as the file system volume containing the data directories, supports *frozen file system snapshots*. See the official documentation for the PostgreSQL[®] relational database management system for further details on file system level backups.

If your external backup software do supports frozen file system snapshots, but the data directory of SQL database server and the resources data directory reside on different disk volumes, it is recommended to order your backups so that the SQL database server's data directory is backed up first.



Backup and Restore of a Model Manager Server Database

MANAGED BACKUPS

The backup directories for the managed server components used by the default Model Manager database are located on the same disk drive as the data directories for these components — see [Default Managed Backup Directories](#). This backup configuration makes it possible to recover data after an accidental deletion in the database but is still vulnerable to a server disk crash. You are therefore strongly recommended to do one of the following options — or a combination thereof — when using a managed Model Manager database:

- Place the managed backup directories on another disk drive or on a mounted file server that the computer account running the Model Manager server has read and write access to.
- Keep the current location for the managed backup directories but let external backup software take periodic backups of these backup directories themselves.

The best option typically depends on your organization's IT infrastructure.

Default Managed Backup Directories

The backup directories for the managed server components used by the default Model Manager server database are located on the same disk drive as the [Default Data Directories](#). For the predefined **NetworkService** account in Windows[®], for example, the directories are located inside:

```
C:\Windows\ServiceProfiles\NetworkService\AppData\Local\COMSOL\
ModelManager\ManagedDatabases\default-managed-backup
```

with subdirectories:

- `data_backup` — backup directory for the managed SQL database server.
- `resources_backup` — backup directory for the managed resources directory.

Replace `NetworkService` with `LocalService` in the directory path for the predefined **LocalService** account in Windows[®].

For Linux[®], the directories are located inside:

```
<user>/ .comsolmodelmanagerdata/manageddatabases
/default-managed-backup
```

and for macOS:

```
<user>/Library/Application Support/COMSOL/ModelManager/
ManagedDatabases/default-managed-backup
```

SERVER PROCESSES AND MEMORY FOOTPRINT

The memory footprint of the main Model Manager server process is expected to be small and will typically not be a deciding factor for the overall system requirements of the server computer. When using managed server components for a Model Manager server database, there are two groups of subprocesses started on the server computer:

- Managed SQL database server processes.
- Managed search index server processes.

The memory footprint of both of these two groups will scale with the size of the database. A recommendation is to start with at least a few gigabytes of RAM and then follow up over time as your database grows.

EXAMPLES OF SERVER SETUPS

A few examples of server setups for a Model Manager server installation, in order of increasing complexity, is summarized as follows:

- A Model Manager server with a fully managed Model Manager database. The main Model Manager server process, as well as subprocesses for the managed SQL database server and managed search index server, all run on a single server computer. The data directories are all located on the server computer's file system. The backup directories are located on a physical disk drive different from that of the data directories.

Apart from the location of the backup directories, this is the setup used by the default managed database that is added automatically when installing a Model Manager server. It is also the recommended setup when getting started with a Model Manager server.

- Same as the first option except that the backup directories are located on a separate file server mounted by the server computer.
- Same as the first option, except that the managed resources directory is located on a separate file server and the backup directories are located on a second separate file server.
- A Model Manager server with a partially managed Model Manager server database. The main Model Manager server process and the managed search index server subprocess run on the main server computer. The Model Manager server connects to an external SQL database server running on a second server computer. The data directory for the managed search index server is located on the main server computer, the data directory for the external SQL database server is located on the second server computer, and the managed resources directory is located on a separate file server. The backup directory for the managed resources directory is located on a second separate file server, while the backup of the external SQL database server is handled by your IT department. This option gives the most flexibility to, for example, scale up memory and disk size over time.

Installing in Windows

Start the Model Manager server installation by using the media that you have received or by using an internet download. In the installer, after selecting your preferred language, choose **New COMSOL Model Manager Server 6.2 Installation**.

LICENSE

In the next step, **License**, select the format of your license in the **License format** list under **License information**. For the **License file** option, write the file path to the license file that you have received from your COMSOL representative in the **License file** field. Click **Browse** to browse to and choose another file path. You can also use the **Port number** and **Hostname** option or the **Three-server redundancy** option if your license administrator has given you these details.

At this point, the installer detects the type of license used. The following instructions assume that the license used during installation corresponds to a Model Manager server license.

PRODUCTS

In the next step, **Products**, select the installation location and which software products and components you want to install. All products and components are selected by default.

- **Model Manager Server** — installs the Model Manager server software components.
- **Managed PostgreSQL®** — installs software components for a PostgreSQL® database system managed by the Model Manager server. Clear the check box if you only want to use an external relational database management system provided by your organization.
- **Managed Apache Solr™** — installs software components for an Apache Solr™ search platform managed by the Model Manager server. Clear the check box if you only want to use an external Apache Solr™ installation provided by your organization.
- **License Manager** — installs the COMSOL License Manager. You only need to install this on the computer where you would like to run the license manager.

OPTIONS

In the **Options** step:

- Select the **Create COMSOL Model Manager Server 6.2 folder on Windows Start Menu** check box to install **Start** menu shortcuts (Windows® operating system only).
- Select the **Add Windows Firewall rules for COMSOL programs** check box to add Windows Firewall rules.

- Select the **Check for updates after installation** check box to enable checking for updates after installation.
- In the **Java runtime environment** list, select **Built-in** to use the default Java runtime included with the Model Manager server installation. Select **Custom** if you prefer to use another Java runtime that you have licensed and installed.

LICENSE MANAGER

The **License Manager** step appears if the installer installed the FlexNet® license manager and your computer has been designated to run the license server. If this step does not appear, you can manage the license server using LMTTOOLS. This step contains the following options:

- The **Install license manager as a Windows service** check box is selected by default; if you clear the check box, the license manager will not be available as a Windows® service.
- The **Path to the debug log file** field contains a file path to the location of the license manager debug log file (default: C:\comsol.log). Click **Browse** to browse to and choose another file path.
- The **Service name** field shows the service name, LMCOMSOL, for information only.
- Under **Additional license manager options**, you can select any of the following check boxes:
 - Select the **Allow the lmdown command to be executed only from this computer** check box to restrict the execution of the lmdown command, which you can use to shut down the license manager, to this computer only.
 - Select the **Disable the lmdown command** check box to make the lmdown command unavailable.
 - Select the **Disable the lremove command** check box to make the lremove command, which you can use to remove a user's license, unavailable.

SERVER

The **Server** step makes it possible to set up the Model Manager server as a Windows® service. You can also configure an initial root administrator account, which you can use to log in to the Model Manager server web interface to continue the server setup once the installation has finished.

Basic Settings

In the **Default Model Manager Server port** field, write the port number that the server will use. The default is 8181.

Windows Service Settings

There are two ways to install the Model Manager server in Windows®. If you select the **Install Model Manager Server as a Windows service** check box (the default), the Model Manager server is then installed as a Windows® service. Otherwise, the Model Manager server is installed as a regular executable.

Use the **Startup** list to configure how to start the Model Manager server when installed as a Windows® service:

- **Disabled** — the service is disabled.
- **Manual** — the service will not be started after the installation. You need to start the service manually.
- **Automatic** — the service is configured to start automatically when the host computer boots or restarts. This is the default choice.
- **Automatic (Delayed Start)** — the service starts automatically but is delayed until all automatic-start threads have finished starting.



The **Automatic** setting provides the highest availability to users of the installed server.



If you choose **Disabled** or **Manual**, you can enable or start the Model Manager server service from the command line or by using the **Manage local services** shortcut installed on the **Start** menu in Windows®. The same configuration options are also available in the snap-in **Services** in the **Microsoft Management Console** — search for `services` from the Windows® **Control Panel**.

In the **Service account** list, select the computer account that runs the service. You can choose the predefined **LocalService**, **LocalSystem**, or **NetworkService** accounts in Windows®, which have no password. When using other accounts, choose **Custom** and

provide a username and password. By default, the installer selects the predefined **LocalService** service account.



The **LocalService** account has limited privileges intended for running local services. It is the recommend service account if you intend to set up a Model Manager server database using a single server computer — see [Examples of Server Setups](#). If you, for example, intend to mount a file server or connect to an external SQL database server, you must use a service account with network access privileges — for example, the predefined **NetworkService** account or a custom account.



The PostgreSQL[®] database system does not support running under a user account with administrative privileges on Windows[®] — that is, a member of the Administrators or Power Users group. If you choose to not install this software component, you are still strongly recommended to use an account with less privileges — see also [Security](#).



Consult the documentation that came with the operating system for more information about service accounts.

Default Local Administrative User

Select the **Create default local administrative user** check box to set a username and a temporary password for a root administrator account. You can use that account to log in to the Model Manager server web interface after the installation finishes to continue the setup of the server.



Only clear the **Create default local administrative user** check box if you are repairing or updating a previous installation for which a root administrator account already exists.

Troubleshooting

When you click **Next** from the **Server** step, the installer will verify some of the given settings, including employing a test service to check that the given service account details are valid and that the service account has the right permissions to access the installation directories. The following are some common warning and error messages that may arise and suggestions on how to address them:

The given service account does not seem able to start services due to a logon failure. Please verify that it has the right to log on as a service. The installer failed to start the test service. To verify that a custom user account has the right to log on as a service, you can check the security settings for the **Log on as a service** policy under **Control Panel>Administrative Tools>Local Security Policy>Local Security Settings>Local Policies>User Rights Assignments** in Windows®.

The installation directory is not accessible. Please verify that the directory is correct and that the service account has access to this path. The service launcher failed to locate the test service when starting it. The service account either does not see that location or does not have read permissions. Commonly, this is caused by using an installation location that either has security permissions that do not include the service account, or by the location being on a network share that is not mounted by the service account.

The given service account is invalid or does not exist or the password is wrong. The test service could not be installed or started due to a problem with the service account. Verify that the account details are correct.

Failed to verify the service account. An unexpected error happened. If you believe that the installation settings are correct, you can proceed with the installation. Further details are given in the `comsolsetup.log` file after the installation.

No administrative user has been defined. You did not specify an administrator account under **Default local administrative user**. In this configuration, it will not be possible to log in to Model Manager server unless the server has already been configured with accounts after a previous installation.

INSTALL

The **Install** step shows a list of the software products and components that will be installed. Click the **Install** button to begin the installation.

FINISH

The last **Finish** step is shown when the installation has finished. You can view an installation log if, for example, there were warnings or errors during the installation.

Installing in Linux and macOS

Installing in the Linux® and macOS operating systems is similar to [Installing in Windows](#). The **Server** step contains [Basic Settings](#) and the option of creating a [Default](#)

[Local Administrative User](#). See [Starting a Model Manager Server](#) for more information about how to automatically launch a Model Manager server after installation on Linux® and macOS.



For security reasons, it is not recommended to use an account with administrative privileges to run a Model Manager server. The Apache Solr™ search platform and the PostgreSQL® database system do not support running as a root user on either Linux® or macOS.

Modifying an Installation

To modify an existing Model Manager server installation — for example, to change the default port or reset the administrator password — follow these steps:

- 1 Stop the Model Manager server.
If installed as a Windows® service (default for Windows®), you can stop the Model Manager server service using the **Manage Local Services** shortcut installed on the **Start** menu under COMSOL Launchers.
- 2 Start the installer for the Model Manager server and select a language.
- 3 Choose **Add/Remove Products and Reinstall**. If a dialog box is opened, browse to the installation directory.
- 4 Specify the new configuration in the steps that follow, similarly as for a new installation.
You can clear the **Create default local administrative user** check box if you have already added an administrative user, ignoring the warning that no administrative user has been defined when clicking **Next**.
- 5 When the installer has finished, start the Model Manager server again.



See [Starting a Model Manager Server](#) for instructions on how to start and stop the server on different platforms.

RESETTING THE DEFAULT ADMINISTRATOR PASSWORD

You can reset the default administrator password by following the abovementioned steps for modifying an existing Model Manager server installation. In the **Server** step, select the **Create default local administrative user** check box and write a new password to reset any existing password.

UNINSTALLING

To uninstall an existing Model Manager server installation, follow these steps:

- 1 Stop the Model Manager server.
If installed as a Windows[®] service (default for Windows[®]), you can stop the Model Manager server service using the **Manage Local Services** shortcut installed on the **Start** menu under COMSOL Launchers.
- 2 Start the Model Manager server installer and select a language.
- 3 Choose **Uninstall COMSOL Model Manager Server 6.2**. If a dialog box is opened, browse to the installation directory.
- 4 Click the **Uninstall** button.



Uninstalling a Model Manager server will *not* delete any databases.

Migrating From an Older Installation

To migrate from an older version of a Model Manager server running as a service on Windows[®]:

- 1 Stop the currently running Model Manager server Windows[®] service. Also make sure that the Windows[®] service is not set to be started automatically — that is, set the startup type to **Disabled**.
- 2 Install **COMSOL Model Manager Server 6.2** — see [Installing in Windows](#).
- 3 Once you have verified that the **COMSOL Model Manager Server 6.2** has successfully started, uninstall the old version of the server if desired.

For a systemd service installed on Linux[®], make sure to stop and disable the old service before installing the new service.

The Model Manager databases hosted by the old Model Manager server will now be available in the new server.



You are strongly recommended to at least stop any older installations of Model Manager server before installing a newer version to not risk any unwanted cross-talk in the data directories. When stopping an older Model Manager server installed as a Windows[®] service, make sure that the service is not set to be started automatically. When stopping an older systemd service installed on Linux[®], make sure that the service is also disabled.

When a Model Manager server starts for the first time, it tries to locate a Model Manager server [Preference Directory](#) belonging to an older installation of the server. If such a preference directory is found, all server settings are migrated to the new installation's preference directory. This includes, for example, any account settings and database configurations set up for the older server. You can also manually copy the subfolder `db` in the preference directory of the older installation to that of the newer installation *before* the latter server has been installed and started (you will need to first manually create the newer installation's preference directory). This is useful, for example, if the older installation is located on another computer.

The [Data Directories](#) for any configured databases are *not* copied, however. This means that older installations of Model Manager server may potentially read and write in the same data directories as the newer installation unless the older servers are first stopped.



[Backward Compatibility for Server Databases](#)

Starting a Model Manager Server

In this section, you will learn how to start a Model Manager server in the Windows[®], Linux[®], or macOS operating systems.

In this section:

- [Starting as a Windows Service](#)
- [Starting Manually in Windows](#)
- [Starting in Linux](#)
- [Starting in macOS](#)
- [Firewalls](#)
- [Setting Up the Model Manager Server for Secure Connections](#)
- [Running Behind a Reverse Proxy](#)
- [Command Options](#)

Starting as a Windows Service

By default, a Model Manager server is installed as a Windows[®] service that is set to start automatically. In this case, a link to the Model Manager server web interface is installed on the **Start** menu as **Model Manager Server**. See [Accessing the Web Interface](#) for more information about accessing the web interface and, for example, continuing the setup of a fresh installation.

To manually start or stop the Model Manager server service, or to configure it if it is started automatically, use the **Manage Local Services** shortcut installed on the **Start** menu under COMSOL Launchers. The same configuration options are also available in the snap-in **Services** in the **Microsoft Management Console**. Search for `services` from the Windows[®] **Control Panel**.



Before starting a Model Manager server, you need to start a FlexNet[®] license manager. The license manager can be installed together with a Model Manager server — see [Installing a Model Manager Server](#).

Starting Manually in Windows

If you cleared the **Install Model Manager Server as a Windows service** check box during installation, you can start a Model Manager server manually by doing one of the following:

- Click the **COMSOL Model Manager Server 6.2** shortcut installed on the **Start** menu.
- Double-click the Windows[®] executable in a file browser.
- Run the Windows[®] executable from a command window.

The Model Manager server executable is located at:

```
<Installation directory>\bin\win64\comsolmodelmanagerserver.exe
```

An example of a typical *<Installation directory>* is:

```
C:\Program Files\COMSOL\COMSOL62\ModelManagerServer
```

For all methods of starting a Model Manager server, the command window displays a short message that the server has started and which port is used. Press Ctrl+C to stop the server and exit the command window.



When a Model Manager server has been installed as a Windows[®] service, the shortcut for starting manually is not available on the **Start** menu. It is not recommended to start a Model Manager server manually via its executable when the service is running. Since the service typically runs under a special system user account, it does not share its configuration settings with a Model Manager server that was started manually.



[Installation Directory](#)

Starting in Linux

To start a Model Manager server manually on Linux[®], type

```
<Installation directory>/bin/comsol modelmanagerserver
```

To stop the server, press Ctrl+C in the terminal window where it was started.



The Apache Solr™ search platform and the PostgreSQL® database system do not support running as a root user on Linux®.

SYSTEMD SERVICE

Use the following instructions to start a Model Manager server in a Linux® version that includes systemd:

1 Create a file:

```
/usr/lib/systemd/system/comsolmodelmanagerserver62.service
```

with content similar to:

```
[Unit]
Description=COMSOL Model Manager server 6.2
Wants=network.target network-online.target
After=network.target network-online.target

[Service]
Type=exec
User=comsoluser
Group=comsolgroup
TimeoutSec=300
ProtectHome=off
ExecStart=/usr/local/comsol62/modelmanagerserver/bin/comsol
modelmanagerservice

[Install]
WantedBy=multi-user.target
```

You only need to replace the values for User and Group.

2 Activate the service via:

```
systemctl enable /usr/lib/systemd/system/
comsolmodelmanagerserver62.service
```

3 Start the service via:

```
systemctl start comsolmodelmanagerserver62
```

To stop the Model Manager server service cleanly, run:

```
systemctl stop comsolmodelmanagerserver62
```

Starting in macOS

To start a Model Manager server manually, type

```
<Installation directory>/bin/comsol modelmanagerserver
```

To stop the server, press Ctrl+C in the terminal window where it was started.



The Apache Solr™ search platform and the PostgreSQL® database system do not support running as a root user on macOS.

Firewalls

You must open up firewalls that exist between the server and your users. Open up for incoming TCP connections to the port given during installation (by default, 8181) or controlled by the `-port` command option — see [Command Options](#). To improve security, you can reduce the IP address range to known potential IP addresses for your users. If you are running a Model Manager server behind a reverse proxy, only open up the port of the reverse proxy instead — see [Running Behind a Reverse Proxy](#). If you have added a connector with TLS enabled, only open up the port of that connector instead — see [Adding a Connector With TLS](#).

Internally, a Model Manager server may launch subprocesses and communicate with them on various ports. These additional ports do not need to be open in the firewall for users of the Model Manager server.

Setting Up the Model Manager Server for Secure Connections

You are strongly recommended to set up the Model Manager server for secure connections using one of the following two options — see also [Security](#):

- Place the Model Manager server behind a reverse proxy with TLS encryption configured. This way, no particular configuration of the Model Manager server itself is required. The communication between the reverse proxy and the Model Manager

server will not be encrypted. See also [Running Behind a Reverse Proxy](#) for more information.

- Add a TLS-enabled connector with a certificate obtained from a trusted certificate authority (CA) to the Model Manager server — see [Adding a Connector With TLS](#).



To connect to a Model Manager server using HTTPS, select the **Require secure connection** check box in COMSOL Multiphysics.

Running Behind a Reverse Proxy

In some cases, it is advantageous to use a reverse proxy as an intermediary between the Model Manager server and its users. Use cases of a reverse proxy include:

- Providing many web services with the same hostname — for example, serving the Model Manager server at `example.com/modelmanagerserver` and a webmail server at `example.com/webmail`.
- Offloading TLS encryption onto the reverse proxy, either for using hardware-accelerated encryption or to simply avoid configuring encryption for each web service separately.
- Firewall features that protect the web service from attacks.
- Integration with single sign-on systems for authentication — see [Proxy Authentication](#).

To support serving the Model Manager server at the path `/modelmanagerserver`, as in the preceding example, the reverse proxy must be configured to pass any regular HTTP requests to, for example, `example.com/modelmanagerserver/foo` on to the Model Manager server as `modelmanagerserver.com:8181/foo`. Additionally, the reverse proxy should be configured to rewrite the cookie path `/api` set by the Model Manager server to `/modelmanagerserver/api`.

Any reverse proxy that supports HTTP can be used with the Model Manager server. Two common reverse proxies are Apache `mod_proxy` and NGINX[®]. See the documentation of the reverse proxy software for how to set it up to forward requests.

TLS ENCRYPTION

By configuring the reverse proxy to serve HTTPS, the integrity and confidentiality of the communication between the client and the reverse proxy is protected by the TLS

protocol. This is the recommended way to host the Model Manager server on any network where cleartext traffic could be intercepted.

Command Options

You can specify optional command options when starting a Model Manager server using the syntax

```
comsolmodelmanagerserver.exe [<options>]
```

on Windows[®] and

```
comsol modelmanagerserver [<options>]
```

on Linux[®] and macOS platforms. See [Table 2-1](#) for various command options available for the server.

TABLE 2-1: COMMAND OPTIONS FOR A MODEL MANAGER SERVER.

OPTION	DESCRIPTION
-c	Path to license file.
-comsolinifile	Path to ini-file to use when launching.
-h, -help	Show help message.
-port	Port number used by the server.
-prefsdir	Path to preference directory.
-tmpdir	Path to temporary file directory.
-v, -version	Show version information.

Security

The following guidelines summarize the best practices for running the Model Manager server in a secure way:

- Do not give unprivileged users in the organization shell access to the system where the Model Manager server runs. Although the file system directories and processes of the Model Manager server are protected from interference by other users on the system through operating system permissions, it is general best practice to restrict shell access on production servers.
- Set up the Model Manager server to use TLS (via HTTPS) when connecting from web browsers and COMSOL Multiphysics. This increases the protection of passwords sent when logging in and reduces the risk of data leaks. The easiest way to get transport layer security is to use a reverse proxy with a certificate, which might already have been set up for other systems. Alternatively, you can add a TLS-enabled connector with a certificate to the Model Manager server — see [Adding a Connector With TLS](#).
- Use a low-privilege account when running the Model Manager server process. This reduces the risk of privilege escalation attacks to the system. See [Installing a Model Manager Server](#) for more information.
- Configure the firewall of the computer or network segment where the Model Manager server runs to only expose the main port of the Model Manager server to the outside. If running the Model Manager server behind a reverse proxy, only expose the port of the reverse proxy to the outside instead.
- If you expose the Model Manager server to the internet, make sure to operate the Model Manager server on a network isolated from your regular corporate network. One option for implementing this is to install the Model Manager server on a computer in a so-called *DMZ network* located between the internet and your regular corporate network. Another option is to install the Model Manager server utilizing a cloud service provider.
- Ensure that the [Preference Directory](#) of the Model Manager server is not accessible by untrusted parties.
- Using permissions, you can configure access control for the content stored on the Model Manager server as needed. You can also use separate instances of the Model Manager server — with each server using their own databases — to isolate different

content and user groups. The latter is best practice when exposing the Model Manager server to users outside of your organization.

Password Security

Local passwords stored by the Model Manager server are hashed by 100,000 iterations of the PBKDF2WithHmacSHA256 algorithm. This means that an adversary that gets access to the hashed passwords will not easily be able to obtain the original passwords. However, if an adversary does obtain a local password (for example, by brute-force guessing a weak password), the adversary could log in to the Model Manager server.

By default, the Model Manager server writes hashed local passwords to the file `/db/settings/local.db` in the [Preference Directory](#). The temporary password set for the [Default Local Administrative User](#) during installation is stored using the same hash algorithm in the file `tempadminlogin.properties` in the root of the installation directory.

As mentioned previously, it is recommended to use TLS (via HTTPS) to protect passwords sent from web browsers and COMSOL Multiphysics to the Model Manager server. A warning appears in COMSOL Multiphysics at the time of login if the connection to the Model Manager server is not secure. Web browsers will show similar warnings.

Passwords saved in COMSOL Multiphysics for connecting to the Model Manager server (“Remember password”) are stored encrypted so that only the logged-in user can access them. On Windows[®], the encryption is done using the Data Protection API of the operating system. On Linux[®] and macOS, the encryption is done using a master key stored in the preference directory, protected by file system permissions. The same encryption is used to protect passwords stored by a Model Manager server configured to connect to external SQL database servers or Apache Solr[™] servers using password-based authentication methods — see also [Databases with External Server Components](#).

When a COMSOL Multiphysics client is connected to a COMSOL Multiphysics server on another host computer, the communication is typically on a nonsecure channel. This means that Model Manager server passwords entered in the user interface on the client computer are sent to the COMSOL Multiphysics server computer in an unencrypted cleartext format. To prevent this exposure, either tunnel the connection between the COMSOL Multiphysics client and the COMSOL Multiphysics server using a secure SSH tunnel, or first enter and save the Model Manager server credentials using a local COMSOL Multiphysics instance running on the COMSOL Multiphysics

server computer. A Model Manager server password already saved on the COMSOL Multiphysics server computer is not sent to the COMSOL Multiphysics client computer when connecting to a Model Manager server.

Administration

In this chapter, you will learn how to configure and administer a Model Manager server via its web interface. You can log in to the web interface to, for example, set up a Model Manager server database or use the web-based asset management system — see also [Asset Management](#). This chapter assumes that you have already installed and started a Model Manager server.

In this chapter:

- [Accessing the Web Interface](#)
- [Connectors](#)
- [Login Configuration](#)
- [Accounts](#)
- [Managed Server Components](#)
- [Model Manager Server Databases](#)
- [Database Administration](#)

Accessing the Web Interface

In a web browser, go to `http://localhost:8181`. If you are accessing it remotely, use the server name — computer name and domain, or the local IP address — of your server instead of `localhost`. If you installed with another port number than the default `8181`, use that number instead. Enter your username and password. Click the **Log In** button.

If you entered the username for the root administrator account configured during installation — see [Default Local Administrative User](#) — you will be prompted to change your temporary password. Write the new password in the **Password** field and then repeat the password in the **Repeat password** field. Click **Save**.



Resetting the Default Administrator Password

Overview of Web Pages

The web pages in a Model Manager server can be grouped as follows:

- Web pages for the asset management system. Click the **Home** link (🏠) to open the **Home** page, in which you can search for assets in the [Current Database](#). If you have been granted access to repositories in the database, you can also search for models and data files. See [Asset Management](#) for more details.
- Web pages for system administration. Select **System** in the **Administration** menu (⚙️) to open the administration area for configuring the Model Manager server. Only root administrators can access this area.
- Web pages for database administration. Select **Database** in the **Administration** menu (⚙️) to open the administration area for configuring the [Current Database](#). Although all users can access this area, only those granted the necessary database permissions can make changes. See [Database Administration](#) for more details.
- The web page for the currently logged-in account. Click (👤) and select **My Account** to open [The My Account Page](#).

To log out from the web interface, click (👤) and select **Log Out**.

Current Database

Root administrators can add multiple server databases to a Model Manager server. Select the *current database* to show in the web interface from the list of active databases in the top navigation menu. The list is only shown if there is more than one active database available. The server database set as the *default database* is initially selected in the list when you log in to the web interface — see also [Database Configurations](#).



COMSOL Multiphysics always connects to the default database of a Model Manager server.

Changing the Language

Root administrators can change the language used in the Model Manager server web interface:

- 1 In the **System** administration area, under **Configuration**, click **Language** to open the **Language** administration page.
- 2 Select a language in the list.
- 3 Click **Save**.

Connectors

You add a *connector* to a Model Manager server in order to configure ports that the server should listen on. This is useful if you, for example, want to enable transport layer security (TLS) for secure connections via HTTPS on port 443.



A Model Manager server is automatically configured with a default connector listening on the port set during installation. This connector is, however, *not* configured with TLS enabled.

The **Connectors** page, opened by clicking **Connectors** in the **System** navigation sidebar, shows a table with all connectors that have been manually added to the Model Manager server. The table is empty for a new installation. Click on the label of a connector in the table to show more details for that connector. Click the **Add** button to add a new connector.

Adding a Connector With TLS

You can set up a Model Manager server to accept secure connections by adding a connector with TLS encryption enabled.



An alternative to adding a connector is using a reverse proxy with TLS configured — see [Running Behind a Reverse Proxy](#).

To add a new connector:

- 1 Write a label for the connector in the **Label** field.
- 2 Select when and how the connector should start listening on its port in the **Start mode** list. Select **Automatic** if the connector should start listening when the Model Manager server starts. Select **Manual** if the connector should only start listening when clicking **Start** on the **Connector** page.
- 3 Select **Default** in the **Port** list if the connector should listen on the default port — the default port is 443 if TLS is enabled, 80 if TLS is disabled. Select **Manual** to manually write the port to listen on in the shown input field.

- 4 Select **All** in the **Listener address** list to let the connector listen on all local addresses. Select **Custom** and write an IP address in the shown input field to only listen on that address.
- 5 Select between **Enabled** or **Disabled** in the **Enable TLS** list to explicitly set whether or not TLS is enabled for the connector. Select **Automatic** if TLS should be enabled if, and only if, a TLS host configuration has been added.
- 6 Select between **Enabled** or **Disabled** in the **Support HTTP/2** list to explicitly set whether or not the HTTP/2 protocol is enabled for the connector. Select **Automatic** if HTTP/2 is to be enabled if, and only if, TLS is enabled.

To enable TLS for the new connector, you need to add a TLS host configuration with an associated server certificate for every hostname that should accept secure connections from clients. You can use a wildcard in your hostname if it is supported by the corresponding server certificate.

Click **Add TLS Host Configuration** to add a new TLS host configuration for the connector.

- 1 Select **Default** in the **Hostname** list to accept connections regardless of the hostname used by a connecting client. The **Default** option can be used for at most one TLS host configuration. Select **Custom** and write the name of the host associated with your certificate in the shown text field to use a custom hostname for this TLS host configuration. You can write either a fully qualified domain name, say `modelmanager.example.com`, or a wildcard domain name, say `*.example.com`.
If no TLS host configuration uses the **Default** option for the **Hostname**, the first configuration in the list will be used for all connections for which the hostname of a connecting client does not match any **Custom** hostname — effectively changing it to use the **Default** option. Using **Custom** hostnames is therefore only useful if there is more than one TLS host configuration for a connector.
- 2 Select the TLS versions that are available when communicating with clients in the **Client compatibility** list. Select **Modern** for TLSv1.3, **Intermediate** for TLSv1.2 and TLSv1.3., or **Automatic** if the Model Manager server should decide.

- 3 Under **Certificates**, click **Add Certificate** to add a server certificate.
 - a Either write a new name, or keep the suggested name, for the certificate in the **Name** field. The name must be unique in the list of certificates for a TLS host configuration.
 - b Select the type of location used to store certificates in the **Location** list. A Model Manager server supports the following locations: [PEM Files](#), [PKCS#12 Keystore](#), [Windows Native Certificate Store](#), and [MacOS Native Keychain](#).
 - c Provide the settings specific for the selected location.
- 4 Click **Save** to save the new connector.



If you add a TLS-enabled connector and only want to allow secure connections to the server, you should add firewall rules that block external access on the port set during installation — 8181, by default.

PEM FILES

The **PEM files** location type has the following settings:

- **Certificate file.** The path on the Model Manager server computer to the certificate file.
- **Chain file.** The path on the Model Manager server computer to the certificate chain file. Leave empty if the certificate does not have a chain file.
- **Key file.** The path on the Model Manager server computer to the file containing the certificate key. Leave empty if there is no separate file for the key.
- **Key password.** The key password for the certificate.

PKCS#12 KEYSTORE

The **PKCS#12 Keystore** location type has the following settings:

- **Keystore file.** The path on the Model Manager server computer to the keystore file.
- **Keystore password.** The password for the keystore file.
- **Key alias.** The alias of the certificate to use from the keystore. Select from the list of available aliases.
- **Key password.** The key password for the certificate.

WINDOWS NATIVE CERTIFICATE STORE

Select the alias of the certificate to use from the **Windows native certificate store** in the **Key alias** list. The alias is typically the friendly name for the certificate, as reported by

the Windows[®] certificate manager, or the first encountered common name of the certificate if it does not have a friendly name.

A Model Manager server can only access server certificates in the **Personal > Certificates** directory in the **Current user** certificate store. It cannot access the **Local computer** certificate store or the per-service **Service account** certificate stores.

If the Model Manager server has been installed as a Windows[®] service, the **Current user** certificate store belongs to the user account the service is configured to log on as — for example **NT AUTHORITY\LocalService** for **LocalService** when running Windows[®] using an English system locale. This means that importing or administrating certificates must be done running as this same service account. The tool PsExec from Sysinternals can be used to launch the Windows[®] certificates management tool interactively as a service account, for example:

```
psexec -i -u "NT AUTHORITY\LocalService" mmc certmgr.msc
```



The service account names depend on the system locale used when running Windows[®].

MACOS NATIVE KEYCHAIN

Select the alias of the certificate to use from the **macOS native keychain** in the **Key alias** list.

Common TLS Certificate Formats

A TLS certificate is typically obtained by sending a *certificate signing request* (CSR) for the website you want to host the Model Manager server to a trusted *certificate authority* (CA). In response, you may receive one of several different formats for the certificate and its chain or root certificate. This includes:

- A single *<filename>.pfx* file. This is typically a [PKCS#12 Keystore](#) file containing the certificate as well as its chain or root certificate.
- A collection of up to three files of the form: *<certificate-filename>.cer*, *<chain/root-filename>.cer*, and *<key-filename>.key*. Use these three files in the settings for [PEM Files](#).

The Connector Page

The **Connector** page contains configuration settings and current status of a connector added to the Model Manager server. The connector fields are:

- **Label.** The label of the connector. Used for display purposes in the Model Manager server web interface.
- **Start mode.** Either **Automatic** if the connector should start listening when the Model Manager server starts or **Manual** if the connector should only start listening when clicking the **Start** button.
- **Port.** The port the connector is listening on. A **Default** value if the connector should listen on the default port — the default port is 443 if TLS is enabled, 80 if TLS is disabled. Otherwise the number of the port.
- **Listener address.** Either **All** if the connector listens on all local addresses or an IP address if the connector only listens on that address.
- **Enable TLS.** The enabled status for transport layer security — **Enabled** or **Disabled** if TLS is always enabled or disabled for the connector, **Automatic** if TLS should be enabled if, and only if, the connector has a TLS host configuration.
- **Support HTTP/2.** The enabled status for the HTTP/2 protocol — **Enabled** or **Disabled** if the HTTP/2 protocol is always enabled or disabled for the connector, **Automatic** if the HTTP/2 protocol should be enabled if, and only if, TLS is enabled.
- **Status.** The current status of the connector.

The status shows, for example, if the connector is currently started or if the connector has failed to start due to some configuration issue. For the latter case, read the nested error information to look for hints to the underlying cause.

- **TLS Host Configurations.** A list of TLS host configurations and certificates added to the connector.

Click the **Stop** button to stop the connector from listening on its configured port. Click **Start** to start listening. Click **Reload** to force a started connector to reload based on its current configuration — this is useful, for example, if a TLS certificate has been externally updated.

Click the **Edit** button to edit the configuration of the connector. Click the **Delete Permanently** button to permanently delete the configuration.



Clicking **Delete Permanently** will only delete the configuration for the connector; it will not delete any configured keystore or certificate files.

Login Configuration

A Model Manager server can be configured to use the following authentication mechanisms when a user tries to log in:

- **Local Authentication.** The username and password provided on the **Log In** page are compared with the username and (hashed) password stored in a local settings database in the Model Manager server [Preference Directory](#).
- **External Authentication.** The username and password provided on the **Log In** page are compared with that found in an external credentials storage using, for example, Windows® authentication or lightweight directory access protocol (LDAP).
- **Proxy Authentication.** The authentication is delegated to a reverse proxy placed in front of the Model Manager server. Any HTTP requests passed from the reverse proxy to the Model Manager server are trusted to have already been authenticated by the proxy.

Local Authentication

A root administrator can manually add an account on [The Accounts Page](#) by providing a username and password to be used by a user of Model Manager server. The password is hashed, as described in [Password Security](#), and stored in a local settings database in the Model Manager server [Preference Directory](#). The user may later change their password on [The My Account Page](#).

External Authentication

A Model Manager server uses the pluggable authentication framework *Java authentication and authorization service* (JAAS) to integrate with external authentication protocols. A login configuration is specified by listing vendor-supplied *login modules* that should be applied (in the order they are listed) when the user tries to log in.

Support for Windows® authentication and LDAP authentication as login modules is included with a Model Manager server installation. You may also specify a custom login module using a *module class* obtained from a third-party vendor.

A user that has authenticated via a login module can be automatically assigned membership to one or more groups in a Model Manager server database for the

duration of their login session. This is useful if an administrator has already set up group memberships in the external credentials storage and wants to use the same groups for controlling access to items in a Model Manager server database.



See [Database Administration](#) and the *Model Manager Reference Manual* for more information on user management and access control.

The **External Authentication** page, opened by clicking **External Authentication** in the **System** navigation sidebar, shows all currently configured login modules in the **Login Modules** field and all group membership mappings in the **Group mappings** field. Both lists are empty for a new installation of a Model Manager server.

Click **Edit** to edit the login modules and group membership mappings. Click **Clear** to clear all login modules and group membership mappings.

You can test the validity of the configuration by clicking the **Test** button — see [Testing an External Authentication Configuration](#).

ADDING A LOGIN MODULE

When editing the configuration for external authentication, click **Add Login Module** to add a new login module to the overall list of login modules.

- 1 Select the type of login module in the **Type** list. The available options are [Windows Login](#), [LDAP](#), and [Custom](#) — see below for their specific configuration settings.
- 2 Select how a successful or unsuccessful authentication attempt by the login module should be handled in relation to other login modules in the **Control flag** list. The available options are:
 - a **Requisite**. The login module is required to succeed for the overall authentication to succeed. If it succeeds, the next login module in the list of modules will be applied. If it fails, no other login module is applied.
 - b **Optional**. The login module is not required to succeed for the overall authentication to succeed. Regardless if it succeeds or fails, the next login module in the list of modules will be applied.
 - c **Required**. The login module is required to succeed for the overall authentication to succeed. Regardless if it succeeds or fails, the next login module in the list of modules will be applied.
 - d **Sufficient**. If the login module succeeds, no other login module is applied. If it fails, the next login module in the list of modules will be applied.

- 3 Click the **Add Option** button under **Options** to add custom options in the **Option** and **Value** fields. The supported options are vendor specific to each login module.

The overall external authentication is successful only if all applied login modules with **Required** and **Requisite** succeeded. Note that *applied* is important here — if a login module with **Sufficient** succeeds, only login modules with **Required** and **Requisite** specified *before* that login module need to succeed. At least one login module with **Sufficient** or **Optional** must succeed if there are no login modules with **Required** or **Requisite**.



The value set for the **Control flag** is irrelevant when only a single login module has been added. The overall authentication succeeds if, and only if, the login module is successful.

Windows Login

If you have installed the Model Manager server on the Windows[®] operating system, users can log in using Windows[®] authentication by adding a **Windows login** module.

- 1 Write the format to identify the user being authenticated in the **Principal format** list. Select **Fully qualified names** for usernames of the form `<domain>\<user>`. Select **SID** to use security identifiers to identify users. Select **Both** to allow both fully qualified names and security identifiers.
- 2 Write the format for the returned principal role names of the authenticated user in the **Role format** list. Select **Fully qualified names** to return names of the form `<domain>\<group>`. Select **SID** to return security identifiers. Select **Both** to return both fully qualified names and security identifiers. Select **None** if no principal role names should be returned.

LDAP

You can configure a login module to use LDAP to communicate with a credentials storage supporting that protocol — for example, Windows[®] Active Directory[®] (AD) or OpenLDAP.



See the Java documentation for `com.sun.security.auth.module.LdapLoginModule` to learn more about the configuration settings for the LDAP login module.

- 1 Write the URL connection string used to connect to the LDAP server and directory containing the user to authenticate in the **User Provider** field. You can write several

URL connection strings separated by spaces; each URL connection string will be attempted until a successful connection to the LDAP server is established.

- 2 Write an optional LDAP filter string in the **User filter** field. This specifies a search filter used to locate the user in the LDAP directory by the user's *distinguished name* in LDAP. Use the special token {USERNAME} as a placeholder for the username provided on the **Log In** page.
- 3 Write an LDAP distinguished name or some other string name in the **Authentication identity** field. The name is used to locate the user in the LDAP directory. The name must contain the token {USERNAME}, which acts as a placeholder for the username provided on the **Log In** page. The **User filter** field must be specified if the **Authentication identity** field does not contain a distinguished name.
- 4 Write a string name in the **Authorization identity** field that will be used to associate a principal name with a successfully authenticated user. You can write a single token of the form {<attribute-name>} as a placeholder for a user attribute value in LDAP that holds a principal name.
- 5 Select whether or not the connection to the LDAP server should use secure sockets layer (SSL) in the **Use SSL** list.



You are strongly recommended to enable SSL to avoid sending credentials in cleartext to the LDAP server.

A sample configuration for Windows[®] Active Directory[®] may look like:

- 1 **User Provider:** ldap://ldap.example.com:3268/DC=example,DC=com
- 2 **User filter:** (&(sAMAccountName={USERNAME}))(objectclass=user)
- 3 **Authentication identity:** {USERNAME}@example.com
- 4 **Authorization identity:** {MEMBEROF}
- 5 **Use SSL:** Enabled

A sample configuration for OpenLDAP may look like:

- 1 **User Provider:** ldap://ldap.example.com/ou=People,dc=example
- 2 **User filter:** (&(uid={USERNAME}))(objectClass=inetOrgPerson)
- 3 **Authentication identity:**
- 4 **Authorization identity:** {MAIL}
- 5 **Use SSL:** Enabled

with **Authentication identity** intentionally left blank.

Custom

You can configure a custom login module provided by a third-party vendor. Write the fully qualified Java class name of the login module in the **Module class** field. Consult the documentation of the login module for the custom options that need to be provided.

ADDING GROUP MAPPINGS

You can map principal names returned by login modules to corresponding group names in a Model Manager server database. A successfully authenticated user will be considered a member of mapped groups for the duration of their login session.



A *principal* is an umbrella term for a user, role, or group.

When editing the configuration for external authentication, click **Add Group Mapping** to add a new mapping.

- 1 Write the name of a principal returned by a login module in the **Principal name** field. This can be a username, role name, group name, or some other string name specific to the type of login module.
- 2 Under **Mapped groups**, click **Add Group** and write the name of a group in a Model Manager server database that the authenticated user should become a member of. The group will be automatically created in the database if it did not already exist when the user logged in. Repeat to map the principal to more groups.



Successfully mapped group names are visible in the **External Group Memberships** field on the [The My Account Page](#) for the logged in account.

TESTING AN EXTERNAL AUTHENTICATION CONFIGURATION

You can test the current configuration for external authentication. Click the **Test** button and write a username and password in the opened dialog box.

If the authentication succeeds, you will see the principal names returned by the login modules, including the name used to authenticate with, as well as the names of any groups these principal names would be mapped to. If the authentication fails, you will see the error messages returned by the login modules.

Proxy Authentication

You may find it useful to delegate all authentication to a trusted reverse proxy placed in front of the Model Manager server. To do this, configure the reverse proxy for authenticating clients using any authentication scheme you want. The Model Manager server itself can then identify the externally authenticated user by a HTTP header set by the reverse proxy. The reverse proxy can, for example, use the Basic HTTP authentication scheme and set the `Authorization` HTTP header with the username of the user and an empty password.

The **Proxy Authentication** page, opened by clicking **Proxy Authentication** in the **System** navigation sidebar, shows the current configuration for such a proxy authentication scheme. A new installation of a Model Manager server is not configured to use a proxy authentication scheme, as indicated in the **Use proxy authentication** field.

Click **Edit** to edit the current configuration. Click **Clear** to clear the configuration in its entirety.

When editing the proxy authentication configuration:

- 1 Select the **Use proxy authentication** check box to enable proxy authentication. Clear the check box to disable.

You may prefer disabling instead of clicking the **Clear** button if you later want to enable the proxy authentication again using previously set configuration settings.

- 2 Under **Trusted proxy connections**, select the **From loopback address** check box to trust proxies running on the same computer as the Model Manager server. Write any other host to be trusted under **From hosts** and click the **Add** button. Repeat this with all other trusted hosts.

The Model Manager server will trust that connections made from these hosts have already been successfully authenticated if the hosts supply a username in the request — see below.

- 3 Select **Basic auth username** in the **Username source** list if the authenticated user's username is set in the `Authorization` HTTP header (using an empty password) by the reverse proxy. Select **HTTP header** if the username is set in a custom HTTP header. Write the name of the custom header in the **Username** field.
- 4 Write the name of an optional HTTP header containing the user's display name in the **User display name** field. Leave empty to let the display name be the same as the username.

- 5 Write the name of an optional HTTP header containing the name of a group that the user is a member of in the **Group name** field.
- 6 Write the name of an optional HTTP header containing a display name for a group that the user is a member of in the **Group display name** field. Leave empty to let the display name be the same as the group name.
- 7 Select whether or not the Model Manager server should write detailed log messages when a user tries to authenticate using proxy authentication in the **Detailed logging** check box. This is useful during initial setup of proxy authentication but afterward should be left cleared to not fill up log files.

You may add as many group mappings as you want — group names and display names will be paired in the order they are encountered as headers in the request. An authenticated user will be set as a member of these groups for the duration of their login session. A group will be automatically created in the database if it did not already exist.



Successfully mapped group names are visible in the **External Group Memberships** field on the [The My Account Page](#) for the logged in account.


Accounts

Users log in to a Model Manager server from the COMSOL Desktop environment or via a web browser using an account stored in the local settings database of the server. The account is created automatically if authenticating using [External Authentication](#) or [Proxy Authentication](#). If using [Local Authentication](#), a root administrator first needs to create each account via the web interface. A hashed account password is stored in the local settings database in this latter case — see [Password Security](#). No account password is stored by the Model Manager server for the other authentication mechanisms.

In this section:

- [The My Account Page](#)
- [The Accounts Page](#)

The My Account Page

Click () in the upper-right corner and select **My Account** to open the **My Account** page for editing your own account settings — including, for example, changing your password if using [Local Authentication](#).

To set a new password:

- 1 Click the **Password** button.
- 2 Write your current password in the **Current password** field.
- 3 Write your new password in the **Password** field and then repeat the password in the **Repeat password** field.
- 4 Click **Save**.

You can also change your *display name*, an alternative name for your username used for display purposes. Click **Edit** and write a new name in the **Display name** field. Click **Save**.

If your account has been automatically mapped as a member in groups via a login configuration, those groups are shown in the **External Group Memberships** field.

The Accounts Page

The **Accounts** administration page, opened by clicking **Accounts** in the **System** navigation sidebar, shows a table with all accounts created for the Model Manager server — either manually by a root administrator or automatically by the server.

The table columns are:

- The **Display Name (Name)** column — the display name and username of the account.
- The **Root Administrator** column — whether or not a root administrator account.
- The **Created** column — point in time when the account was created.
- The **Last Modified** column — point in time when the account settings was last modified.
- The **Last Login** column — point in time when the account last logged in to the Model Manager server.

Click on the name of an account in the table to show more details for that account. Click the **Add** button to manually add a new account.

ADDING ACCOUNTS

To add a new account:

- 1 In the **Name** field, write the username that the account uses to authenticate with the Model Manager server.
- 2 In the **Display name** field, write an alternative name used for display purposes.
- 3 Select **Yes** or **No** in the **Root Administrator** list to set whether or not the new account is a root administrator account.
- 4 In the **Password** field, write the password that the account uses to authenticate with the Model Manager server. Repeat the password in the **Repeat password** field.
The password can later be changed by the user on [The My Account Page](#).
- 5 Click **Save**.



Root administrators automatically pass any authorization checks in Model Manager — see also the *Model Manager Reference Manual*. You are strongly encouraged to limit the number of root administrator accounts.



You only need to manually add accounts if using [Local Authentication](#).

THE ACCOUNT PAGE

The **Account** page lets you view and edit a specific account. The account fields are:

- **Name.** The unique username of the account in the Model Manager server.
- **Display name.** The account name used for display purposes in the Model Manager server web interface.
- **Root administrator.** Whether or not the account is a root administrator account.
A root administrator account is allowed to view and configure the Model Manager server via the **System** administration area.
- **Created.** Point in time when the account was created.
- **Last Modified.** Point in time when the account settings was last modified.
- **Last Login.** Point in time when the account last logged in to the Model Manager server.

Click **Edit** to edit the display name and root administrator status of the account.

Click **Change Password** to set a new password for the account if using [Local Authentication](#). This is useful, for example, if a user forgets their account password — the user may later change the reset password on the **My Account** page.

Click **Delete Permanently** to permanently delete the account.



Deleting an account will only remove that user's ability to log in to the Model Manager server — any information related to the user in a Model Manager server database will be left unchanged. If using any authentication mechanism other than [Local Authentication](#), a deleted account will be automatically recreated the next time the user successfully authenticates.

Managed Server Components

A Model Manager server database is a combination of three separate server components — a SQL database for the version control management of assets, models, and data files, a dedicated directory on the file system for storing large binary and text data, and a collection of search indexes used by the Model Manager search functionality. You can either use server components managed by the Model Manager server itself, including starting and stopping component subprocesses and handling backups, or you can use external server components provided by your organization. In the latter case, you will need to handle availability and backups for the server components outside of the Model Manager server.

In this section you will learn how to add and configure such *managed server components*, including setting up backups for their data directories. You will also learn how to restore these components from a backup. See the next section to learn how you combine managed server components into a Model Manager server database, optionally swapping out all, or some, of the components with your own external server components.



To configure a database that is entirely managed by the Model Manager server itself, select the **Managed PostgreSQL®** and **Managed Apache Solr™** check boxes in the [Products](#) step of the Model Manager server installation.

In this section:

- [Managed SQL Database Servers](#)
- [Managed Resources Directories](#)
- [Managed Search Index Servers](#)

Managed SQL Database Servers

A *managed SQL database server* is a server instance of the PostgreSQL® relational database management system controlled by the Model Manager server itself. The **Managed SQL Database Servers** page, opened by clicking **SQL Database Servers** in the **System** navigation sidebar, shows a table with the configurations of all such managed SQL database servers. A single configuration — **Default managed PostgreSQL® server** —

is initially shown in the table. This default managed SQL database server is automatically created by the Model Manager server when the latter is started for the first time — see also [The Default Managed Database](#).



Links to the **Managed SQL Database Servers** page are hidden in the web interface if you cleared the **Managed PostgreSQL®** check box in the **Products** step of the Model Manager server installation.

Click on the label of a managed SQL database server in the table to show more details for that managed server. Click the **Add** button to add a new managed SQL database server.

ADDING MANAGED SQL DATABASE SERVERS

You can add a new managed SQL database server to Model Manager server. This is useful if, for example, you want to use different options than the default ones for the automatically added managed SQL database server but, at the same time, want to keep the default one around as a reference.

To add a new managed SQL database server:

- 1 Write a label for the managed SQL database server in the **Label** field.
- 2 Write the path on the file system where the data directory for the managed SQL database server will be created in the **Data directory** field.
The data directory must be located on a physical disk unless running on Linux[®] using NFS mounted in hard mode. Other network disk configurations are not supported.
- 3 Select when and how the managed SQL database server is started in the **Start mode** list. Select **Automatic** if the server should start when the Model Manager server is started or when the SQL database server is used by a Model Manager database. Select **Manual** if the server must be manually started on [The Managed SQL Database Server Page](#).
- 4 Select the **Enable backup** check box to enable backups of the data directory. Write the path on the file system where backups will be written to in the **Backup root directory** field. The backup root directory will be automatically created if it does not already exist on the file system. See also [Backup of a Managed SQL Database Server](#).
- 5 In the **Trigger mode** list under **Restore**, select **Automatic (most recent)** if the data directory should be automatically restored from the most recent backup if it is missing on disk, or if a previous restore has not yet finished, when the managed SQL

database server starts. Select **Manual** if a restore requires manual triggering on [The Managed SQL Database Server Page](#). See also [Restore of a Managed SQL Database Server](#).

- 6 Click **Save** to add the new configuration. The data directory and the backup root directory will be automatically created as needed the next time the managed SQL database server is started.



The specified data directory must either not exist at all on the file system or, if it exists, be empty if you want to create a new managed SQL database server. A nonempty data directory is assumed to belong to an existing managed SQL database server that you want to add a configuration for in the Model Manager server.



The parent directory to the data directory must already exist on the file system for the server to automatically start after saving. If not, you can create the directory from [The Managed SQL Database Server Page](#) and then manually start the server.



Placing the data directory of a managed SQL database server on a network file system is not supported unless running on Linux[®] using NFS mounted in hard mode — see also [Installation Planning](#).



Prefer placing the backup directory on a physical disk different than the disk containing the data directory. Otherwise, you might risk complete data loss in case of a server disk crash.



At least one SQL database needs to be added to the managed SQL database server to use it for a Model Manager server database — see [SQL Databases](#).

THE MANAGED SQL DATABASE SERVER PAGE

The **Managed SQL Database Server** page contains configuration details for a managed SQL database server that is added to the Model Manager server — see [Adding Managed SQL Database Servers](#). The configuration fields are:

- **Platform.** The database system platform used for the managed SQL database server.
- **Label.** The label of the managed SQL database server. Used for display purposes in the Model Manager server web interface.
- **Data directory.** The file system path to the data directory of the managed SQL database server.

The currently available disk space on the file storage containing the data directory is shown below the path.

You can see the current status of the data directory in its **Status** field. If the data directory does not exist on the file system, click the **Create** button to create it.

- **Start mode.** The start mode of the managed SQL database server — **Automatic** if the server starts when the Model Manager server is started or when the SQL database server is used by a Model Manager database, **Manual** if the server must be manually started.

The adjacent **Status** field shows, for example, if the server is currently started or if the server has failed to start due to some configuration issue. For the latter case, read the nested error information to look for hints to the underlying cause.

Click the **Stop** button to stop a running managed SQL database server. Click **Start** to start it.

To edit the configuration for a managed SQL database server, proceed as follows:

- 1 Deactivate all Model Manager server databases using [SQL Databases](#) in the managed SQL database server. See also [The Database Configuration Page](#).
- 2 Click the **Stop** button to stop the managed SQL database server.
- 3 Click the **Edit** button.
- 4 Enter the new configuration settings. If you change any directory paths, manually move the corresponding directories on the file system via, for example, a system file explorer *before* saving.
- 5 Click the **Save** button.
- 6 Start the managed SQL database server if not automatically started.
- 7 Activate all databases deactivated in the first step.

Deleting a Managed SQL Database Server Configuration

Click the **Delete Permanently** button to permanently delete the configuration.



Clicking **Delete Permanently** will only delete the configuration for the managed SQL database server; it will not delete its data directory or backup root directory. You can add an existing SQL database server again by adding a new configuration that points to its data directory and backup root directory.

SQL DATABASES

The version control management system for simulation models and other data are stored in SQL databases in the managed SQL database server. A single Model Manager server database uses exactly one SQL database. The SQL databases are stored inside the data directory for the managed SQL database server.

The **Databases** section shows a table with all SQL databases in the SQL database server. The table columns are:

- The **Name** column — the name of the SQL database.
- The **Used by** column — the Model Manager database using the SQL database, if any.
- The permanently delete column — click the delete button to permanently delete the SQL Database.



Permanently deleting a SQL database cannot be undone unless you restore the managed SQL database server from a backup.

To add a new SQL database, write the name of the database and click the **Add** button under the **Databases** section on the **Managed SQL Database Server** page. The name may only contain the letters a–z and digits 0–9 and must start with a letter. Some specific names are explicitly forbidden — either because these databases already exist internally in the database server or because they are *reserved names*. The Model Manager server will show an error message if such a forbidden name is entered. A suggested name is `modelmanager` — the same name used by [The Default Managed Database](#).



The managed SQL database server must be running in order to add new databases.

BACKUP OF A MANAGED SQL DATABASE SERVER

When you enable backup for a managed SQL database server, it will continuously back up any data written to its data directory to a subdirectory inside the *backup root directory*. This backup is *incremental* in the sense that it records all changes made in the server's SQL databases as a transactional log. When you restore a managed SQL database server, including all its SQL databases, this transactional log is replayed.

Since the time it takes to replay the transactional log scales with the running time of the managed SQL database server, it is wise to periodically create a new *base backup*. This is a complete point-in-time snapshot of the managed SQL database server. Once you have created such a base backup, the incremental backup will instead record changes using that backup as a new baseline — thereby speeding up any future restore.

On the **Managed SQL Database Server** page, click the **Trigger Base Backup** button to create a new base backup. The **Backup root directory** field shows the root directory path in which backups are stored.



A new base backup is automatically created when you enable backup for a managed SQL database server.

RESTORE OF A MANAGED SQL DATABASE SERVER

If you have configured a managed SQL database server with **Trigger mode** for restoring set to **Automatic (most recent)**, a missing data directory will be automatically restored using the most recently backed-up state in the backup root directory when the managed SQL database server is started.

You can also manually restore the data directory of a managed SQL database server by clicking the **Trigger Restore** button on the **Managed SQL Database Server** page. A confirmation dialog box is shown, informing you that the following steps will be taken by the Model Manager server:

- 1 The SQL database server is stopped if currently running.
- 2 Any existing data directory is renamed to avoid overwriting it with the restored data.
- 3 The most recent base backup is copied from the backup root directory to the data directory.
- 4 The SQL database server is launched in recovery mode and automatically restores itself using the most recently found backed-up state — see [Backup of a Managed](#)

[SQL Database Server](#). Once finished, the SQL database server is made available again.



You are strongly recommended to also recreate any search indexes whose indexed data is populated from SQL databases in the restored SQL database server — see the manual steps in [Restore of a Managed Search Index Server](#). Otherwise, search results in the Model Manager may not reflect what is stored in the SQL database.



You are recommended to deactivate any Model Manager server database using [SQL Databases](#) in the managed SQL database server before triggering a restore. See [The Database Configuration Page](#).

Managed Resources Directories

A *managed resources directory* is a data directory controlled by the Model Manager server itself. The directory contains binary and text data that are deemed too large to store directly inside a SQL database. The **Managed Resources Directories** page, opened by clicking **Resources Directories** in the **System** navigation sidebar, shows a table with the configurations of all such managed resources directories. A single configuration — **Default managed resources directory** — is initially shown in the table. This default managed resources directory is automatically created by the Model Manager server when the server is started for the first time — see also [The Default Managed Database](#).

Click on the label of a managed resources directory in the table to show more details for that directory. Click the **Add** button to add a new managed resources directory.



Unlike [Managed SQL Database Servers](#) and [Managed Search Index Servers](#), a managed resources directory cannot be shared between Model Manager databases. You need to create a new managed resources directory for every Model Manager database you add to the Model Manager server.

ADDING MANAGED RESOURCES DIRECTORIES

You can add a new managed resources directory to Model Manager server. This is useful if you, for example, want to use different options than the default ones for the

automatically added managed resources directory but at the same time want to keep it around as a reference.

To add a new managed resources directory:

- 1 Write a label for the resources directory in the **Label** field.
- 2 Write the path on the file system where the directory will be created in the **Data directory** field.

The parent directory to the data directory must already exist on the file system. If not, open a system file explorer on the server computer and manually create it.

The data directory may be placed on either a physical disk or a network disk.
- 3 Select the **Enable backup** check box to enable backups of the data directory. Write the path on the file system where the backup will be written to in the **Backup directory** field. The backup directory will be automatically created if it does not already exist on the file system. See also [Backup of a Managed Resources Directory](#)
- 4 In the **Trigger mode** list under **Restore**, select **Automatic** if the data directory should be automatically restored from backup if it is missing on disk, or a previous restore has not yet finished, when activating the Model Manager server database using the directory — see [The Database Configuration Page](#). Select **Manual** if a restore requires manual triggering on [The Managed Resources Directory Page](#). See also [Restore of a Managed Resources Directory](#).
- 5 Click **Save** to add the new configuration.



The specified data directory must be empty if you want to initialize a new managed resources directory. A nonempty data directory is assumed to belong to an existing managed resources directory that you want to add a configuration for in the Model Manager server.



Place the backup directory on a physical disk different than the disk containing the data directory. Otherwise, you might risk complete data loss in case of a server disk crash.



You may want to consider placing the data directory for managed resources on a file server whose file system is mounted by the server computer running the Model Manager server. The same goes for the backup directory using a second mounted file server. See [Installation Planning](#).

THE MANAGED RESOURCES DIRECTORY PAGE

The **Managed Resources Directory** page contains details on the configuration of a managed resources directory added to the Model Manager server — see [Adding Managed Resources Directories](#). The configuration fields are:

- **Label.** The label of the managed resources directory. Used for display purposes in the Model Manager server web interface.
- **Data directory.** The file system path to the managed resources directory.
The currently available disk space on the file storage containing the directory is shown below the path.
- **Used by.** The Model Manager database using the managed resources directory, if any.

To edit the configuration for a managed resources directory, proceed as follows:

- 1 Deactivate any Model Manager server database using the resources directory, if any — see also [The Database Configuration Page](#).
- 2 Click the **Edit** button.
- 3 Enter the new configuration settings. If you change any directory paths, manually move the corresponding directories on the file system via, for example, a system file explorer *before* saving.
- 4 Click the **Save** button.
- 5 Activate the database using the resources directory, if any.

Deleting a Managed Resources Directory Configuration

Click the **Delete Permanently** button to permanently delete the configuration.



Clicking **Delete Permanently** will only delete the configuration for the managed resources directory; it will not delete the directory itself. You can add the managed resources directory again using a new configuration that points to the directory and its backup directory.

BACKUP OF A MANAGED RESOURCES DIRECTORY

If you enable backup for a managed resources directory, the Model Manager server will continuously back up any files written to the resources directory to the backup directory. The **Backup directory** field on the **Managed Resources Directory** page shows the directory path for this backup.

You can also trigger a complete backup of the whole resources directory — this will back up *all* files in the resources directory to the backup directory. Click the **Trigger Backup** button on the **Managed Resources Directory** page. A backup will then be scheduled to begin if and when the Model Manager server database using the resources directory is activated.

RESTORE OF A MANAGED RESOURCES DIRECTORY

If you have configured a managed resources directory with **Trigger mode** set to **Automatic**, a missing directory will be automatically restored from the backup directory when a Model Manager server database using the managed resources directory is activated.

You can also manually restore a managed resources directory by clicking the **Trigger Restore** button on the **Managed Resources Directory** page. A restore will then be scheduled to begin when the Model Manager server database using the resources directory is activated. A confirmation dialog box is shown, informing you that the following steps will be taken once the restore has begun:

- 1 The SQL database component of the Model Manager server database will be queried for all files expected to be present in the resources directory.
- 2 All expected files that are missing from the resources directory are copied from the backup directory.



A triggered restore of a managed resources directory only begins once a Model Manager server database using the resources directory is activated — see [The Database Configuration Page](#).

Managed Search Index Servers

A *managed search index server* is a server instance of the Apache Solr™ enterprise search platform controlled by the Model Manager server itself. The **Managed Search Index Servers** page, opened by clicking **Search Index Servers** in the **System** navigation sidebar, shows a table with the configurations of all such managed search index servers.

A single configuration — **Default managed Apache Solr™ server** — is initially shown in the table. This default managed search index server is automatically created by the Model Manager server when the latter is started for the first time — see [The Default Managed Database](#).



Links to the **Managed Search Index Servers** page are hidden in the web interface if you cleared the **Managed Apache Solr™** check box in the [Products](#) step of the Model Manager server installation.

Click on the label of a managed search index server in the table to show more details for that managed server. Click the **Add** button to add a new managed search index server.

ADDING MANAGED SEARCH INDEX SERVERS





You can add a new managed search index server to the Model Manager server. This is useful if you, for example, want to use different options than the default ones for the automatically added managed search index server but at the same time want to keep it around as a reference.

To add a new managed search index server:

- 1 Write a label for the managed search index server in the **Label** field.
- 2 Write the path on the file system where the data directory for the managed search index server will be created in the **Data directory** field.
The data directory must be located on a physical disk. Placing the directory on a network disk is not supported.
- 3 Select when and how the managed search index server is started in the **Start mode** list. Select **Automatic** if the server should start when the Model Manager server is started or when the search index server is used by a Model Manager database. Select **Manual** if the server must be manually started on [The Managed Search Index Server Page](#).
- 4 Click **Save** to add the new configuration. The data directory will be automatically created as needed the next time the managed search index server is started.



The specified data directory must be empty if you want to create a new managed search index server. A nonempty data directory is assumed to belong to an existing managed search index server that you want to add a configuration for in the Model Manager server.

	The parent directory to the data directory must already exist on the file system for the server to automatically start after saving. If not, you can create the directory from The Managed Search Index Server Page and then manually start the server.
	Placing the data directory of a managed search index server on a network file system is not supported — see also Installation Planning .
	At least two search indexes need to be added to the managed search index server in order to use it for a Model Manager server database — see Search Indexes .
	Unlike Managed SQL Database Servers and Managed Resources Directories , a managed search index server requires no backup.

THE MANAGED SEARCH INDEX SERVER PAGE

The **Managed Search Index Server** page contains details on the configuration of a managed search index server added to the Model Manager server — see [Adding Managed Search Index Servers](#). The configuration fields are:

- **Platform.** The search platform used for the managed search index server.
- **Label.** The label of the managed search index server. Used for display purposes in the Model Manager server web interface.
- **Data directory.** The file system path to the data directory of the managed search index server.

The currently available disk space on the file storage containing the data directory is shown below the path.

You can see the current status of the data directory in its **Status** field. If the data directory does not exist on the file system, click the **Create** button to create it.

- **Start mode.** The start mode of the managed search index server — **Automatic** if the server starts when the Model Manager server is started or when the search index

server is used by a Model Manager database, **Manual** if the server must be manually started.

The adjacent **Status** field shows, for example, if the server is currently started or if the server has failed to start due to some configuration issue. For the latter case, read the nested error information to look for hints to the underlying cause.

Click the **Stop** button to stop a running managed search index server. Click **Start** to start it.

To edit the configuration for a managed search index server, proceed as follows:

- 1 Deactivate all Model Manager server databases using [Search Indexes](#) in the managed search index server. See also [The Database Configuration Page](#).
- 2 Click the **Stop** button to stop the managed search index server.
- 3 Click the **Edit** button.
- 4 Enter the new configuration settings. If you change any directory paths, manually move the corresponding directories on the file system via, for example, a system file explorer *before* saving.
- 5 Click the **Save** button.
- 6 Start the managed search index server if not automatically started.
- 7 Activate all databases deactivated in the first step.

Deleting a Managed Search Index Server Configuration

Click the **Delete Permanently** button to permanently delete the configuration.



Clicking **Delete Permanently** will only delete the configuration for the managed search index server; it will not delete its data directory. You can add an existing search index server again by adding a new configuration that points to its data directory.

SEARCH INDEXES

The search data used by the Model Manager search functionality is populated in *search indexes* in the managed search index server. A Model Manager server database requires two search indexes. The search indexes are stored inside the data directory for the managed search index server.

The **Search indexes** section shows a table with all search indexes in the search index server. The table columns are:

- The **Name** column — the name of the search index.
- The **Used by** column — the Model Manager database using the search index, if any.
- The permanently delete column — click the delete button to permanently delete the search index.



A permanently deleted search index can be recreated by first adding back an empty search index. The empty search index is then automatically populated with search data when the Model Manager server database using the search index is activated.

To add a new search index, write the name of the search index and click the **Add** button under the **Search Indexes** section on the **Managed Search Index Server** page. The name may only contain the letters a–z and digits 0–9 and must start with a letter. Suggested names are **assets** and **items** — the same names used by [The Default Managed Database](#).



The search index server must be running in order to add new search indexes.

Unlike the data directories of [Managed SQL Database Servers](#) and [Managed Resources Directories](#), the data directory and its search indexes do not require backup because the contents can be recreated from that of the SQL database — see [Restore of a Managed Search Index Server](#).

RESTORE OF A MANAGED SEARCH INDEX SERVER

A managed search index server is automatically restored if its data directory is missing when the server is started, although all of its search indexes are then empty. The search indexes will be automatically populated with search data when the Model Manager server database using the search indexes is activated.

You can also manually restore the data directory as follows:

- 1 Deactivate all Model Manager server databases using [Search Indexes](#) in the managed search index server. See [The Database Configuration Page](#).
- 2 Stop the managed search index server.

3 Move any existing data directory to a new location.

4 Start the managed search index server.

At this point, the data directory and its search indexes will be recreated based on the configuration settings of the managed search index server. The search indexes will, however, be empty of search data.

5 Activate all databases deactivated in the first step.

At this point, the empty search indexes will be automatically repopulated with search data loaded from the corresponding SQL database as needed.



Populating the search indexes after a restore may take some time depending on the size of the SQL database.

Model Manager Server Databases

A Model Manager server database consists of the following components:

- A SQL database — either added to a SQL database server managed by the Model Manager server or added to an external SQL database server provided by your organization.
- A resources directory — either a directory managed by the Model Manager server or an external directory.
- Two search indexes — either added to a search index server managed by the Model Manager server or added to an external search index server provided by your organization.

In this section, you will learn how to configure a Model Manager server database using these components. You will see that you can use any combination of [Managed Server Components](#) and external server components for the database. You will also learn how to safely move a Model Manager server database on the file system, as well as how to restore your database from a backup in case of failure.

- [Database Configurations](#)
- [Databases with Managed Server Components](#)
- [Databases with External Server Components](#)
- [Moving a Model Manager Server Database](#)
- [Backup and Restore of a Model Manager Server Database](#)
- [Backward Compatibility for Server Databases](#)



Read [Installation Planning](#) for general guidelines and background information to help you with planning your Model Manager server setup.

Database Configurations

A Model Manager server database can be set as either *active* or *inactive* depending on whether or not it should accept connections from COMSOL Multiphysics or the Model Manager server web interface. You would typically deactivate a Model Manager server database when you want to change its configuration settings or perform a restore from backup. One of the databases can also be set as the *default* one, which is

the default server database that users will connect to when accessing the Model Manager server from COMSOL Multiphysics. It is also the database initially selected as the [Current Database](#) in the web-based asset management system.

The **Database Configurations** page, opened by clicking **Databases** in the **System** navigation sidebar, shows a table with the configurations of all Model Manager server databases. If you selected the **Managed PostgreSQL®** and **Managed Apache Solr™** check boxes in the **Products** step of the Model Manager server installation, the configuration of a single database — **Default managed database** — is initially shown in the table. This database is automatically created by the Model Manager server when the latter is started for the first time — see also [The Default Managed Database](#).

The table columns are:

- The **Label** column — the label of the database. Used for display purposes in the Model Manager server web interface.
- The **Status** column — the active and default status of a database.

Click on the label of a Model Manager server database on the **Database Configurations** page to show more details for that database. Click the **Add** button to add a new database.

THE DEFAULT MANAGED DATABASE

If you selected the **Managed PostgreSQL®** and **Managed Apache Solr™** check boxes when installing the server, a default Model Manager server database is automatically created when the Model Manager server starts for the first time. This database uses managed server components with default locations for their data directories and backup directories. These locations are not ideal, as backups are placed on the same physical disk as the data itself. You are recommended to either add new [Managed Server Components](#) or update the configurations of the automatically added default components — see also [Moving a Model Manager Server Database](#).



A default managed database is only created if you selected both the **Managed PostgreSQL®** and **Managed Apache Solr™** check boxes in the **Products** step of the Model Manager server installation. If you select just one of them, only the corresponding default managed server component will be created.



Place the backup directories for the managed SQL database server and the managed resources directory on a physical disk different than that of their corresponding data directories if you decide to use the default managed database. This will protect you from complete data loss in case of a server disk crash. See [The Managed SQL Database Server Page](#) and [The Managed Resources Directory Page](#) for instructions.



- [Default Data Directories](#)
 - [Default Managed Backup Directories](#)
-

ADDING DATABASES

You can add a new database to the Model Manager server. This is useful if, for example, you want to use different options than [The Default Managed Database](#) but at the same time want to keep the default database around as a reference.

To add a new database:

- 1 Write a label for the database in the **Label** field.
- 2 Write an optional alias for the database in the **Alias** field. When set, the alias must only contain lowercase Latin alphabet characters, numerical characters, underscores, and dashes. The alias can contain, at most, 100 characters.

The alias field is required if you want to connect to a Model Manager server database other than the default server database from a COMSOL Multiphysics installation.
- 3 In the **SQL Database>Type** list, select among the options:
 - **Managed** — to use a SQL database in one of the [Managed SQL Database Servers](#). See [Managed SQL Databases](#).
 - **Custom** — to use a SQL database in an external SQL database server. See [External SQL Databases](#).
- 4 In the **Resources Directory>Type** list, select among the options:
 - **Managed** — to use one of the [Managed Resources Directories](#).
 - **Custom** — to use an external directory on the file system. See [External Resources Directories](#).

- 5 In the **Search Indexes>Type** list, select among the options:
 - **Managed** — to use search indexes in one of the [Managed Search Index Servers](#). See [Managed Search Indexes](#).
 - **Custom** — to use search indexes in an external search index server. See [External Search Indexes](#).
- 6 Click the **Save** button.



You need to activate the database before users can connect to it from COMSOL Multiphysics or the Model Manager server web interface — see [The Database Configuration Page](#).

THE DATABASE CONFIGURATION PAGE

The **Database Configuration** page contains details on the configuration of a database added to the Model Manager server. The configuration fields are:

- **Label.** The label of the Model Manager server database. Used for display purposes in the Model Manager server web interface.
- **Alias.** An optional alias for the Model Manager server database.

The alias is used as an identifier to, for example, connect to the Model Manager server database from a COMSOL Multiphysics installation even when the database is not set as the default one.
- **SQL Database.** The SQL database used by the Model Manager database.
- **Resources Directory.** The resources directory used by the Model Manager database.
- **Search Indexes.** The two search indexes used by the Model Manager database.

Click the **Edit** button to edit the configuration for the database.

Click the **Test Connection** button to test if the configuration can be used to connect to the database.

Click the **Deactivate** button to deactivate the database. Users will not be able to connect to the database from COMSOL Multiphysics or the Model Manager server web interface. Click **Activate** to make the database available again.



Always deactivate a database before editing its configuration.

If you have configured multiple databases, you can click the **Set as Default** button to set the database as the current default. Click **Remove as Default** to remove it as the current default.

Click the **Delete Permanently** button to permanently delete the database configuration.



Clicking **Delete Permanently** will only delete the configuration for the Model Manager server database; it will not delete any data. You can add the database again using a new configuration.

Databases with Managed Server Components

You can configure a Model Manager server database to use components managed by the Model Manager server itself.

MANAGED SQL DATABASES

On the pages for adding or editing a database configuration:

- 1 Select **Managed** in the **SQL Database>Type** list to use a SQL database in one of your [Managed SQL Database Servers](#).
- 2 Select a managed SQL database server in the **Server** list.
- 3 Select the database in the **Database** list.



A SQL database in a managed SQL database server can be used by, at most, one Model Manager server database.

MANAGED RESOURCES DIRECTORIES

On the pages for adding or editing a database configuration:

- 1 Select **Managed** in the **Resources Directory>Type** list to use one of your [Managed Resources Directories](#).
- 2 Select a managed resources directory in the **Directory** list.



A managed resources directory can be used by, at most, one Model Manager server database.

MANAGED SEARCH INDEXES

On the pages for adding or editing a database configuration:

- 1 Select **Managed** in the **Search Indexes>Type** list to use search indexes in one of your [Managed Search Index Servers](#).
- 2 Select a managed search index server in the **Server** list.
- 3 Select two different search indexes to use in the **Assets** and **Items** lists.



A search index in a managed search index server can be used by, at most, one Model Manager server database.

Databases with External Server Components

You can configure a Model Manager server database to use components managed externally by your organization. This is useful if, for example, you already have a SQL database server installation with associated backup routines set up.

EXTERNAL SQL DATABASES

On the pages for adding or editing a database configuration:

- 1 Select **Custom** in the **SQL Database>Type** list to use a SQL database in an external SQL database server.
- 2 Select the type of SQL database platform in the **Platform** list. The supported platforms are:
 - [PostgreSQL® Database](#)
 - [Microsoft® SQL Server® Database](#)
 - [MySQL® Database](#)
 - [Oracle® Database](#)



Set up a backup routine for the SQL database server that includes the SQL database used by the Model Manager server.

PostgreSQL® Database

- 1 In the **Server name** field, write the name of the computer that the PostgreSQL® database server process runs on. Defaults to `localhost`.

- 2 In the **Port** field, write the number of the port that the PostgreSQL[®] database server listens on. Defaults to 5432.
- 3 In the **Database name** field, write the name of a SQL database that you want the Model Manager to use.
- 4 In the **Method** list, under **Authentication**, select the method of authentication. Select **Password** to authenticate using password-based authentication methods. Select **Passwordless** to authenticate using passwordless authentication methods (for example, SSPI on Windows[®]).
- 5 In the **Username** field, write the username used to authenticate with the PostgreSQL[®] database server.
- 6 In the **Password** field, write the password used to authenticate with the PostgreSQL[®] database server. This is only available when using password-based authentication methods.

The password will be encrypted by the Model Manager server and stored in a local SQLite[®] database in the Model Manager server [Preference Directory](#)— see also [Password Security](#).



The Model Manager supports PostgreSQL[®] version 10.0 and newer.

Microsoft[®] SQL Server[®] Database

- 1 In the **Server name** field, write the name of the computer that the Microsoft[®] SQL Server[®] database server process runs on. Defaults to localhost.
- 2 In the **Port** field, write the number of the port that the PostgreSQL[®] database server listens on. Defaults to 1433.
- 3 In the **Database name** field, write the name of a SQL database that you want the Model Manager to use.
- 4 In the **Instance name** field, write the instance name of the Microsoft[®] SQL Server[®] database server.
- 5 In the **Application name** field, write an application name to use when connecting to the database.
- 6 Select the **Use integrated security** check box to authenticate using the Windows account running the Model Manager server process.

- 7 In the **Username** field, write the username used to authenticate with the Microsoft® SQL Server® database server. Only available if the **Use integrated security** check box is cleared.
- 8 In the **Password** field, write the password used to authenticate with the Microsoft® SQL Server® database server. This is only available if the **Use integrated security** check box is cleared.

The password will be encrypted by the Model Manager server and stored in a local SQLite® database in the Model Manager server [Preference Directory](#)— see also [Password Security](#).



The Model Manager supports Microsoft® SQL Server® 2014 and newer.



Microsoft® SQL Server® must be installed with the Full Text Search component to use it with a Model Manager server database.



You are strongly recommended to enable row versioning via the READ_COMMITTED_SNAPSHOT isolation level for the Microsoft® SQL Server® database you intend to use with the Model Manager server database.

MySQL® Database

- 1 In the **Server name** field, write the name of the computer that the MySQL® database server process runs on. Defaults to `localhost`.
- 2 In the **Port** field, write the number of the port that the MySQL® database server listens on. Defaults to `3306`.
- 3 In the **Database name** field, write the name of a SQL database that you want the Model Manager to use.
- 4 Select the **Use SSL** check box to require secure connections.
- 5 Select the **Verify server certificate** check box to require that the MySQL® database server's certificate is verified.
- 6 In the **Username** field, write the username used to authenticate with the MySQL® database server.

- 7 In the **Password** field, write the password used to authenticate with the MySQL[®] database server.

The password will be encrypted by the Model Manager server and stored in a local SQLite[®] database in the Model Manager server [Preference Directory](#)— see also [Password Security](#).

Connecting with the Model Manager server to MySQL[®] requires a JDBC driver that is licensed by your organization. In the file `comsolmodelmanagerserver.ini`, found inside the installation directory for the Model Manager server:

- Windows[®]: `<Installation directory>\bin\win64`
- Linux[®]: `<Installation directory>/bin/glnxa64`
- macOS: `<Installation directory>/bin/maci64`

add the line

```
-Dcs.modelmanager.mysql.jdbcdriver=<Path to JDBC driver>
```

with `<Path to JDBC driver>` being the path to the JDBC driver's JAR file on the file system.

If running as a Windows[®] service, the file is `comsolmodelmanagerservice.ini`.



The Model Manager supports MySQL[®] version 8.0.20 and newer.

Oracle[®] Database

- 1 In the **Server name** field, write the name of the computer that the Oracle[®] Database server process runs on. Defaults to `localhost`.
- 2 In the **Port** field, write the number of the port that the Oracle[®] Database server listens on. Defaults to `1521`.
- 3 In the **System Identifier** field, write the system identifier (SID) of the Oracle[®] Database instance.
Leave this field empty if you want to connect using a service name instead.
- 4 In the **Service name** field, write the service name for the Oracle[®] Database instance.
Leave this field empty if you want to connect using the system identifier instead.
- 5 In the **Username** field, write the username used to authenticate with the Oracle[®] Database server.

- 6 In the **Password** field, write the password used to authenticate with the Oracle® Database server.

The password will be encrypted by the Model Manager server and stored in a local SQLite® database in the Model Manager server [Preference Directory](#)— see also [Password Security](#).

Connecting with the Model Manager server to Oracle® Database requires a JDBC driver that is licensed by your organization. In the file `comsolmodelmanagerserver.ini`, found inside the installation directory for the Model Manager server:

- Windows®: `<Installation directory>\bin\win64`
- Linux®: `<Installation directory>/bin/glnxa64`
- macOS: `<Installation directory>/bin/maci64`

add the line

```
-Dcs.modelmanager.oracle.jdbcdriver=<Path to JDBC driver>
```

with `<Path to JDBC driver>` being the path to the JDBC driver's JAR file on the file system.

If running as a Windows® service, the file is `comsolmodelmanagerservice.ini`.



The Model Manager supports Oracle® Database version 12.2.0.1 and newer.

EXTERNAL RESOURCES DIRECTORIES

On the pages for adding or editing a database configuration:

- 1 Select **Custom** in the **Resources Directory>Type** list to use an external resources directory.
- 2 In the **Directory** field, write the path on the file system to a directory.



Set up a backup routine that backs up the resources directory — including all its subfolders and files — to a physical disk separate from that of the resources directory itself.

EXTERNAL SEARCH INDEXES

On the pages for adding or editing a database configuration:

- 1 Select **Custom** in the **Search Indexes>Type** list to use search indexes stored in an external search index server.
- 2 In the **Platform** list, select **Apache Solr™ server**.
- 3 In the **Base URL** field, write the base URL used to connect to the Apache Solr™ server. Defaults to `http://localhost:8983/solr`.
- 4 In the **Method** list, select the method of authentication. Select **Password** to authenticate using a username and password. Select **None** if the Apache Solr™ server does not require authentication.
- 5 In the **Username** field, write the username used to authenticate with the Apache Solr™ server.
- 6 In the **Password** field, write the password used to authenticate with the Apache Solr™ server.

The password will be encrypted by the Model Manager server and stored in a local SQLite® database in the Model Manager server [Preference Directory](#)— see also [Password Security](#).
- 7 In the **Assets** and **Items** fields, write the names of Apache Solr™ cores to use for the Model Manager server database.



The Model Manager supports Apache Solr™ version 8.7.0 and newer.



You can optionally set up a backup routine that includes the Apache Solr™ cores used by the Model Manager server database. This will make the restore process run faster because there will be less search data to transfer from the SQL database to the cores.

Moving a Model Manager Server Database

You can move the data directories and backup directories of a Model Manager server database using [Managed Server Components](#) to another file system location. This is necessary, for example, if you want to keep using the **Default managed database** for a Model Manager server but the available space on the disk containing the server installation is not sufficient.

This section contains the necessary steps to move the data directories and backup directories of the **Default managed database** when running the Model Manager server

as a Windows® service using the predefined **LocalService** user account. The new directories are inside a hypothetical **D:\ModelManagerServer** for the data directories and **E:\ModelManagerServerBackup** for the backup directories. You can adapt the example by skipping certain steps if you only want to move some of the directories. You can also adapt it by modifying paths if you are running with a different user account, if running on another operating system, or if moving some other Model Manager server database.

Start by creating the *parent* directories to the new data directories and backup directories via, for example, a system file explorer on the server computer:

- 1 Create the **ModelManagerServer** directory on the D drive and the **ModelManagerServerBackup** directory on the E drive.

Next, deactivate the Model Manager server database via the Model Manager server web interface:

- 2 On the **Database Configurations** page in the **System** administration area, click **Default managed database**.
- 3 On [The Database Configuration Page](#) for the **Default managed database**, click the **Deactivate** button.

By deactivating the database, you prevent users from accessing the database via COMSOL Multiphysics or via the Model Manager server web interface while its data directories are being moved.



Database Configurations

You will move the directories for each of the three managed server components used by the **Default managed database** by first stopping the component's subprocess (if any), moving its directories, updating its configuration, and then starting the subprocess again. For the managed SQL database server:

- 4 On the **Managed SQL Database Servers** page in the **System** administration area, click **Default managed PostgreSQL® server**.
- 5 On [The Managed SQL Database Server Page](#) for the **Default managed PostgreSQL® server**, click the **Stop** button.

The data directory of the managed SQL database server is located inside


```
C:\Windows\ServiceProfiles\LocalService\AppData\Local\COMSOL\
ModelManager\ManagedDatabases\default-managed
```

as the `data` subdirectory. Its backup directory is located inside

```
C:\Windows\ServiceProfiles\LocalService\AppData\Local\COMSOL\
ModelManager\ManagedDatabases\default-managed-backup
```

as the `data_backup` subdirectory.

- 6 Move the `data` subdirectory to the `ModelManagerServer` directory on the D drive so that its new path becomes `D:\ModelManagerServer\data`.
- 7 Move the `data_backup` subdirectory to the `ModelManagerServerBackup` directory on the E drive so that its new path becomes `E:\ModelManagerServerBackup\data_backup`.
- 8 On [The Managed SQL Database Server Page](#) for the **Default managed PostgreSQL® server**, click the **Edit** button.
- 9 Write `D:\ModelManagerServer\data` in the **Data directory** field.
- 10 Write `E:\ModelManagerServerBackup\data_backup` in the **Backup root directory** field.
- 11 Click the **Save** button.
- 12 Click the **Start** button to start the managed SQL database server.



Managed SQL Database Servers

The `data` directory and backup directory for the managed resources directory is found next to those of the managed SQL database server as the `resources` subdirectory and the `resources_backup` subdirectory respectively.

- 13 Move the `resources` subdirectory to the `ModelManagerServer` directory on the D drive so that its new path becomes `D:\ModelManagerServer/resources`.
- 14 Move the `resources_backup` subdirectory to the `ModelManagerServerBackup` directory on the E drive so that its new path becomes `E:\ModelManagerServerBackup/resources_backup`.
- 15 On the **Managed Resources Directories** page in the **System** administration area, click **Default managed resources directory**.
- 16 On [The Managed Resources Directory Page](#) for the **Default managed resources directory**, click the **Edit** button.

- 17 Write D:\ModelManagerServer\resources in the **Data directory** field.
- 18 Write E:\ModelManagerServerBackup\resources_backup in the **Backup root directory** field.
- 19 Click the **Save** button.



Managed Resources Directories

For the managed search index server:

- 20 On the **Managed Search Index Servers** page in the **System** administration area, click **Default managed Apache Solr™ server**.
- 21 On [The Managed Search Index Server Page](#) for the **Default managed Apache Solr™ server**, click the **Stop** button.

The data directory for the managed search index server is found next to that of the managed SQL database server as the `index` subdirectory. It has no backup directory.

- 22 Move the `index` subdirectory to the `ModelManagerServer` directory on the D drive so that its new path becomes `D:\ModelManagerServer\index`.
- 23 On [The Managed Search Index Server Page](#) for the **Default managed Apache Solr™ server**, click the **Edit** button.
- 24 Write `D:\ModelManagerServer\index` in the **Data directory** field.
- 25 Click the **Save** button.
- 26 Click the **Start** button to start the managed search index server.



Managed Search Index Servers

Finally, activate the Model Manager server database:

- 27 On [The Database Configuration Page](#) for the **Default managed database**, click the **Activate** button.

The Model Manager server database is now accessible again.

Backup and Restore of a Model Manager Server Database

You should configure backup of your Model Manager server database to protect from data loss in case of, for example, a server disk crash. Backup for a Model Manager server database using [Managed Server Components](#) can be configured from the Model Manager server web interface. Backup for a database using external components needs to be configured outside of the Model Manager server.



Installation Planning

BACKUP OF A MODEL MANAGER SERVER DATABASE

Backup for a Model Manager server database involves configuring separate backups for its SQL database and resources directory. Its search indexes do not require backup.

See [Backup of a Managed SQL Database Server](#) and [Backup of a Managed Resources Directory](#) for the backup of [Managed Server Components](#).

See [External SQL Databases](#) and [External Resources Directories](#) for the backup of custom server components.

RESTORE OF A MODEL MANAGER SERVER DATABASE

The steps required to restore a Model Manager server database from backup depend on how much has been lost in terms of configuration.

If the Model Manager server [Preference Directory](#) is missing, and there is no backup of that directory available, you will first need to recreate the configuration for the Model Manager server database. For [Databases with Managed Server Components](#), begin by recreating the configuration of each managed server component:

- [Adding Managed SQL Database Servers](#)
- [Adding Managed Resources Directories](#)
- [Adding Managed Search Index Servers](#)

Remember to specify the path for the backup root directory of the managed SQL database server, as well as for the backup directory of the managed resources directory. If you set the **Trigger mode** for restoring to **Automatic (most recent)** for the managed SQL database server, and its data directory is missing on disk, the data directory will be automatically restored from backup when the server starts. Similarly, if the **Trigger**

mode for restoring is set to **Automatic**, a managed resources directory is automatically restored if the directory is missing on disk.

Follow the steps in [Adding Databases](#) to recreate a configuration for the Model Manager server database — optionally replacing managed server components with external server components.

When a configuration for the Model Manager server database is available (for example, after recreating it):

- 1 If currently activated, deactivate the Model Manager server database — see [The Database Configuration Page](#) for instructions.
- 2 Restore the SQL database:
 - For an external SQL database server, use the restore functionality specific to the platform.
 - For a managed SQL database server, click the **Trigger Restore** button on the **Managed SQL Database Server** page unless a restore has already been triggered automatically. Wait until the restore finishes. See also [Restore of a Managed SQL Database Server](#).
- 3 Restore the resources directory:
 - For an external resources directory, copy files from the backup directory to their corresponding location in the resources directory — optionally skipping files already present in the resources directory.
 - For a managed resources directory, click the **Trigger Restore** button on the **Managed Resources Directory** page unless a restore will trigger automatically. The restore will begin once the database is activated again; see below. See also [Restore of a Managed Resources Directory](#).
- 4 Restore the search indexes:
 - For an external search index server, recreate the search indexes used by the Model Manager server database or, if you have one available, restore the search index server from backup.
 - For a managed search index server, follow the steps in [Restore of a Managed Search Index Server](#).
- 5 Activate the database if not already activated by a previous step.

At this point, if the Model Manager server database uses a managed resources directory, all missing files will be restored from the backup directory to the data directory. All search indexes will also be populated with search data loaded from the

SQL database as needed. The time for both of these steps to complete depends on the size of the database.



When the Model Manager server database has been activated, it is possible to connect to the database from COMSOL Multiphysics. Search results may be incomplete, however, until all search indexes have been repopulated with search data. Some simulation models and auxiliary data files may also fail to load if they depend on data in the resources directory that have yet to be copied from the backup directory by the ongoing restore.


Backward Compatibility for Server Databases

When a newer version of a Model Manager server connects to a server database created using an older Model Manager server, the database will be automatically upgraded to support any functionality added in the newer version. This upgrade does not modify the storage format of models and data files, however, so older COMSOL Multiphysics versions connecting to the Model Manager server will still be able to open models and data files in the database even after this upgrade.



Models saved in a Model Manager server database have the same version requirement constraint as models stored as MPH-files on the file system: a model saved to a database from a newer version of COMSOL Multiphysics cannot be opened from an older version of COMSOL Multiphysics.

Database Administration

You can administer the [Current Database](#) from the database administration area of the web interface. Select **Database** in the **Administration** menu () in the upper-right corner. Although all users may access this area, only users granted the appropriate database permissions are authorized to make changes.

The **Database** navigation sidebar is divided into two sections, **Security** and **Assets**. The **Security** section is similar to the **Security** node in the **Databases** tree in the Model Manager workspace of COMSOL Multiphysics. You can, for example, manage local group memberships of users or grant general database permissions to users and groups.

The **Assets** section contains pages for administrating and extending the asset management system included with a Model Manager server database. You may, for example, want to add new asset libraries with custom permissions or add new types of assets with their own sets of attributes. See also the [Asset Management](#) chapter to learn more.

In this section:

- [Users](#)
- [Groups](#)
- [Database Permissions](#)
- [Permission Templates](#)
- [Asset Libraries](#)
- [Asset Types](#)
- [Primitive Attributes](#)
- [Composite Attributes](#)



See the *Model Manager Reference Manual* for more information on user management and access control in a Model Manager database.

Users

A *user* in a Model Manager database is primarily used to identify the individual that has saved a database object, the owner of a database object, or the individuals that have

been granted permissions for a database object. A new user is automatically created based on your account the first time you connect to a Model Manager server database.



See [Users](#) in the *Model Manager Reference Manual*.

You can manage users in the database from the **Users** page, opened by clicking **Users** in the **Database** navigation sidebar. You can, for example, see all users that have been active in the database or save group memberships in the database. The latter is useful, for example, if you are unable to provide group memberships via an external credentials storage — see [External Authentication](#). Click on the name of a user in the table to show more details for that user. Click the **Add** button to add a new user.

ADDING USERS

While a user is automatically created as needed when connecting to a Model Manager server database, there may be situations when you want to manually add a new user to the database before that user has connected. One example is when you preemptively want to grant them permissions to various database objects.

To add a new user:

- 1 Write the username of the user in the **Name** field. This username must match that of the account used when authenticating with the Model Manager server — see [Accounts](#).
- 2 Write an alternative display name for the new user in the **Display Name** field. This is the name that will primarily be shown in the user interface.
- 3 All groups that the new user will be a member of are shown as a list under **Group memberships**. Select a group from the **Add group membership** list to add the user as a member. The group membership list is hidden if there are no groups in the database.
- 4 Click **Save** to add the new user.



You do not need to specify any group memberships if you are using [External Authentication](#) and you have set up group mappings — see [Adding Group Mappings](#). The same goes if using [Proxy Authentication](#) with a group name header.

THE USER PAGE

The **User** page contains details on a user in the database, including the display name and all group memberships that are stored explicitly in the database. The fields are:

- **Name.** The unique username of the user in the database. This is mapped to a corresponding account via the account's username — see also [The Account Page](#).
- **Display name.** The name used for display purposes in the Model Manager server web interface.
- **Group memberships.** The groups that the user is a member of. This does not include groups mapped to the corresponding account by a [Login Configuration](#) — see [The My Account Page](#).



Group membership is transitive: if a user is a member of a group, which in turn is a member of another group, the user is considered a member of the latter group as well.

Click **Edit** to change the display name or group memberships of the user. Click **Delete** to delete the user in the database. The deletion is not permanent — you can restore the deleted user by clicking **Restore**.

Groups

A *group* in a Model Manager database is a collection of users and other groups. Groups are used to more easily manage permissions common to several users in the database. You can either create groups manually in the database or, by [Adding Group Mappings](#) to an [External Authentication](#) mechanism, let them be automatically created based on the principal names returned by the external credentials storage. Groups will also be created automatically if using [Proxy Authentication](#) with a group name header.



See [Groups](#) in the *Model Manager Reference Manual*.

The **Groups** page, opened by clicking **Groups** in the **Database** navigation sidebar, shows a table with all groups in the database. Click on the name of a group in the table to show more details for that group. Click the **Add** button to add a new group.

ADDING GROUPS

To add a new group:

- 1 Write the name of the group in the **Name** field. This name must match the principal name returned by the external credentials storage if you want the added group to be associated with a principal in the storage.
- 2 Write an alternative display name for the new group in the **Display Name** field. This is the name that will primarily be shown in the user interface.
- 3 All members of the new group are shown as a list under **Group members**. Select a user or group from the **Add group member** list to add them as a member.
- 4 Click **Save** to add the new group.



You do not need to specify any group members for the group if you are using [External Authentication](#) and you have set up explicit group mappings — see [Adding Group Mappings](#). The same goes if you are using [Proxy Authentication](#) with a group name header.

THE GROUP PAGE

The **Group** page contains details on a group in the database, including the display name and all group members that are stored explicitly in the database. The fields are:

- **Name**. The unique name of the group in the database.
- **Display name**. The name used for display purposes in the Model Manager server web interface.
- **Group members**. The members of the group. This does not include members whose group membership is mapped by a [Login Configuration](#) — see [The My Account Page](#).

Click **Edit** to change the display name or group members of the group. Click **Delete** to delete the group in the database. The deletion is not permanent — you can restore the deleted group by clicking **Restore**.

Database Permissions

Root administrators can delegate a few administrative tasks to users by granting them permissions for actions that target the database itself. The **Database permissions** page, opened by clicking **Database Permissions** in the **Database** navigation sidebar, shows a

table with users and groups mapped to their granted permissions. The table columns are:

- The **Name** column — the name of a user or group.
- The **Permissions** column — the database permissions granted to the user or group.

The table is initially empty, which means that only root administrators can perform these administrative tasks.



See [Access Control](#) in the *Model Manager Reference Manual*.

Click the **Edit** button to edit the table of granted database permissions.

To grant permissions to users or groups:

- 1 Select a user or group from the **Add** list.
- 2 In the added table row, select the permissions to grant from the **Permissions** list.
- 3 Repeat for other users or groups.
- 4 Click the **Save** button to save the table.

Most of the available permissions are listed in the [Permission Catalog](#) in the *Model Manager Reference Manual*. The remaining ones apply to the asset management system and are listed in [Table 3-1](#).

TABLE 3-1: AVAILABLE ASSET MANAGEMENT SYSTEM PERMISSIONS FOR THE DATABASE ITSELF.

PERMISSION	DESCRIPTION
Add asset libraries	Allowed to create a new asset library.
Manage asset types	Allowed to create, edit, and delete asset types and attributes.
Permanently delete assets	Allowed to permanently delete assets.

Permission Templates

You can create *permission templates* in a Model Manager database as a way of reusing permission assignments for different models, files, or assets. This saves you the work of manually granting the same set of permissions to users and groups for multiple database object. Creating permission templates also allows you to propagate permission requirement changes to multiple database objects by only updating the

permissions in one place — the permission assignments of the permission template itself.



See [Permission Templates](#) in the *Model Manager Reference Manual*.

The **Permission Templates** page, opened by clicking **Permission Templates** in the **Database** navigation sidebar, shows a table with all user-defined permission templates in the database. The table columns are:

- The **Name** column — the name of a permission template.
- The **Type** column — the type of database objects that the permission template applies to.

Click on the name of a permission template in the table to show more details for that permission template. Click the **Add** button to add a new permission template.

ADDING PERMISSION TEMPLATES

To add a new user-defined permission template:

- 1 Write a name for the permission template in the **Name** field.
- 2 Select the type of permission template to create in the **Type** list. You can select between **Asset**, **File**, or **Model**.
- 3 Select the user or group you want to grant permissions for in the **Add** list. Added users or groups shows up as table rows. Select each user's or group's granted permissions in the **Permissions** list.
- 4 Click the **Save** button to add the new permission template.

THE PERMISSION TEMPLATE PAGE

The **Permission Template** page contains details on a user-defined permission template in the database. The fields are:

- **Name**. The name of the permission template.
- **Type**. The type of database objects that the permission template applies to.

The permissions that are granted by the permission template are shown in a table.

Click **Edit** to edit the permission template.



Updating the permission assignments for a permission template affects not only future applications of the template but also the permissions of database objects to which the permission template has already been applied.

Click **Delete** to delete the permission template in the database. This deletion is not permanent — you can restore the deleted template by clicking **Restore**.

Click **Delete Permanently** if you *do* want to permanently delete the permission template in the database. Permanent deletion will only succeed if the permission template is not assigned to any database objects.

Asset Libraries

An *asset library* is a container for a collection of assets and their versions in a Model Manager database. When the database is created, a first asset library — **Asset library 1** — is automatically added for you. You would typically add more libraries to the database if you want to restrict access to a collection of assets for a particular set of [Users](#) and [Groups](#). By granting permissions on the asset library, you can, for example, restrict who is able to browse and search the assets in the library.



See [Controlling Access to Assets Using Libraries](#).

The **Asset Libraries** page, opened by clicking **Asset Libraries** in the **Database** navigation sidebar, shows a table with all asset libraries in the database. Click on the name of an asset library in the table to show more details for that asset library.

Click the **Add** button to add a new asset library. Write a name for the new asset library in the **Name** field. Click **Save**.

THE ASSET LIBRARY PAGE

The **Asset Library** page contains details on an asset library in the database. Click **Rename** to change the name of the asset library. Click **Delete** to delete the asset library in the database. The deletion is not permanent — you can restore the deleted asset library by clicking **Restore**.

Click **Owner** to transfer ownership of the asset library to another user. Only the current owner or a root administrator can transfer ownership of the asset library. The user that creates the asset library is automatically set as its initial owner.



See [Owners](#) in the *Model Manager Reference Manual*.

Click **Permissions** to grant permissions for the asset library to users or groups. Only the current owner or a root administrator can grant permissions for the asset library. See [Table 3-2](#) for the permissions available for an asset library.

TABLE 3-2: ALL AVAILABLE PERMISSIONS FOR ASSET LIBRARIES.

PERMISSION	DESCRIPTION
Delete asset library	Allowed to delete the asset library.
Restore asset library	Allowed to restore the asset library when deleted.
Save in asset library	Allowed to save assets as well as perform any other save operation inside the asset library.
Save asset library	Allowed to save the asset library itself to, for example, rename it.
See asset library	Allowed to see the asset library in the user interface. This is a necessary permission for browsing assets inside the asset library.



See [Granting Permissions](#) in the *Model Manager Reference Manual*.

When the Model Manager authorizes a database action that targets an asset in the asset management system, it consults up to two levels of protection: the asset library that the asset belongs to and, possibly, the asset itself. See [Table 4-1](#) for all available permissions for assets.

The necessary permission combinations for possible database actions targeting assets are summarized as follows:

TABLE 3-3: NECESSARY PERMISSION COMBINATIONS FOR PERFORMING POSSIBLE DATABASE ACTIONS THAT TARGET ASSETS IN AN ASSET LIBRARY.

ACTION	ASSET LIBRARY	ASSET
Browse and search assets	See asset library	Open asset
Create a new asset	Save in asset library	N/A
Open an asset page	See asset library	Open asset
Save a new asset version	Save in asset library	Save asset
Delete an asset	Save in asset library	Delete asset
Restore an asset	Save in asset library	Restore asset

Asset Types

You use an *asset type* to define the collection of *attributes* that can be edited on assets having that particular asset type. When the database is created, a first asset type — simply named **Asset** — using a few predefined attributes is automatically created for you. You can modify this asset type by adding new attributes to its definition, or you can create entirely new asset types using a combination of predefined and new attributes.

An asset type has a list of *sections*, with each section having a title and a subset of the asset type’s associated attributes. The order of the attributes in each section, as well as the overall order of the sections themselves, determines how the corresponding **Asset** page for an asset having that particular asset type is displayed.

The **Asset Types** page, opened by clicking **Asset Types** in the **Database** navigation sidebar, shows a table with all asset types in the database. Click on the name of an asset type in the table to show more details for that asset type. Click the **Add** button to add a new asset type.



See also [Adding New Types of Assets](#).

ADDING ASSET TYPES

To add a new asset type:

- 1 Write a name for the asset type in the **Name** field.
- 2 Write a shorthand alias for the asset type in the **Alias** field. The alias must be unique among the set of all aliases and start with an uppercase Latin alphabet character

followed by zero or more such characters, numbers, or underscores. The alias can contain, at most, 16 characters.

The alias is used in web links for assets of this asset type, as seen, for example, in the web browser's address bar on the **Asset** page. Once set for an asset type, it cannot be changed.

- 3 Attributes defined for the asset type are added via sections. Under **Sections**:
 - a Write a title for the section in the **Title** field.
 - b Select attributes for the section in the **Add attribute** list. Added attributes appear in the **Attributes** table. You can rearrange the attributes in the order they should appear in the section using drag and drop — see also [Rearranging Table Rows](#).
- 4 Click the up and down arrows to rearrange the sections in the order you want them to appear on an asset page.
- 5 Click **Save** to add the new asset type.



Different asset types can reuse the same attributes. A single attribute can, however, only appear once as a top-level member on an asset type. An option is to add different composite attributes containing the same primitive attribute — see [Primitive Attributes](#) and [Composite Attributes](#).

THE ASSET TYPE PAGE

The **Asset Type** page contains details on an asset type in the database. The fields are:

- **Name**. The name of the asset type.
- **Alias**. The alias of the asset type.
- **Sections**. The list of sections and their corresponding attributes for the asset type.

Click **Edit** to edit the asset type, including adding, rearranging, and removing its sections or attributes.

Click **Delete** to delete the asset type in the database. The deletion is not permanent — you can restore the deleted asset type by clicking **Restore**.

Click **Delete Permanently** if you *do* want to permanently delete the asset type in the database. Permanent deletion will only succeed if the asset type is not assigned to any assets.

Primitive Attributes

You define the data fields available for a particular asset type using *primitive attributes*. A primitive attribute added to an asset type will appear as an editable data field on all assets having that asset type.



See [Adding New Types of Data Using Attributes](#).

Each primitive attribute has a *value type* defining the kind of data it can store and how you can search on that data. This can be a simple type, such as a text or a number, or an advanced type, such as a link to a model version stored in the database. All available value types are listed in [Table 3-5](#).

A primitive attribute also has a *widget type* defining how the data is edited in the web interface. This can, for example, be an input field, a list of options in a combo box, or a picker for selecting a model version. All available widget types are listed in [Table 3-4](#).

Given a particular widget type, only a subset of all value types are available to combine the widget type with. Some widget types also require a set of *allowed values* for the data, which becomes an additional constraint on the data that a primitive attribute can store.

When a Model Manager server database is created, two primitive attributes — **Attachment** and **Model Version** — are automatically created for you. They correspond, respectively, to an uploaded file attachment and a link to a model version.

The **Primitive Attributes** page, opened by clicking **Primitive Attributes** in the **Database** navigation sidebar, shows a table with all primitive attributes in the database. The table columns are:

- The **Label** column — the label of the primitive attribute.
- The **Widget** column — the type of widget used when editing the primitive attribute.
- The **Value** column — the type of value that can be set for the primitive attribute.
- The **Description** column — the optional description of the primitive attribute.

Click on the label of a primitive attribute in the table to show more details for that primitive attribute. Click the **Add** button to add a new primitive attribute.

ADDING PRIMITIVE ATTRIBUTES

To add a new primitive attribute:

- 1 Write an identifier for the primitive attribute in the **Identifier** field. This identifier must be unique among the set of all attributes — primitive and composite — and start with a Latin alphabet character followed by zero or more additional such characters or an underscore.

The identifier of a primitive attribute is, for example, used in the Model Manager search syntax when filtering on attribute values — see [Searching on Asset Attributes](#).

- 2 Write a label for the primitive attribute in the **Label** field. The label is shown on an asset page next to the asset’s attribute value.
- 3 Write a description of your attribute in the **Description** field.
The description is shown as a tooltip to the attribute label on [The Asset Page](#).
- 4 Select the type of widget used to edit the primitive attribute in the **Widget** list. See also [Widget Types](#).
- 5 Select the type of value that can be stored in the primitive attribute in the **Value** list. See also [Value Types](#).
- 6 For applicable widget types, add allowed values for the primitive attribute in **Allowed values**. Click **Add** to add each value. At least one allowed value must be added.
- 7 Click **Save** to add the new primitive attribute.

Widget Types

The widget type of a primitive attribute determines how the corresponding attribute value is displayed and edited on an asset page. Each widget type supports a subset of all [Value Types](#). The following widget types are available:

TABLE 3-4: ALL AVAILABLE WIDGET TYPES FOR PRIMITIVE ATTRIBUTES.

WIDGET TYPE	SUPPORTED VALUE TYPES	DESCRIPTION
Check box	Boolean	A check box that can be selected or cleared.
Check box list	Keyword array	A set of options in which a subset is chosen by selecting check boxes.
Combo box	Integer or Keyword	A set of options in which a single option can be selected from a combo box.
Date picker	Date	A calendar in which a date can be selected.
File upload	Attachment	A file browser for uploading a file as an attachment.

WIDGET TYPE	SUPPORTED VALUE TYPES	DESCRIPTION
File version picker	File version	A search widget for finding and selecting a data file version in the database.
Hyperlink	Link	Input fields for editing an address and a link text for a web resource.
Input field	Date, Integer, or Keyword	An input field for editing a date, integer, or keyword string.
Input fields	Keyword array	Multiple text input fields for editing short, name-like, keyword strings.
List box	Keyword array	A set of options in which a subset is chosen by selecting rows from a list.
Model version picker	Model version	A search widget for finding and selecting a model version in the database.
Number	Integer	A number input field for writing an integer.
Radio button	Integer or Keyword	A set of options in which a single option can be selected from a row of radio buttons.
Text area	Date, Integer, Keyword, or Text	A text input field for writing longer text that can, for example, contain several paragraphs.
User picker	User	A search widget for finding and selecting a user in the database.

The check box list and list box widget types are functionally equivalent, with the main difference that the latter always shows all available options when editing. Similarly for the combo box and radio button widget types.

The difference between the file upload and the file version picker widget types may not be obvious. The former is used to attach files to assets such that these attachments are version controlled together with the asset itself. The latter is used to link to existing data files that are version controlled as separate items in a Model Manager database.

Value Types

The value type of a primitive attribute determines how the corresponding attribute data is validated, as well as how that data is made available for searching and filtering. The following value types are available:

TABLE 3-5: ALL AVAILABLE VALUE TYPES FOR PRIMITIVE ATTRIBUTES.

VALUE TYPE	DESCRIPTION
Attachment	A file uploaded as an attachment on an asset. The attribute value contains a unique identifier for the attachment.
Boolean	A true or false value.
Date	An UTC date and time value in the ISO-8601 format.
File version	A link to a data file version in the database. The attribute value contains a unique identifier for the data file version.
Integer	An integer value.
Keyword	A keyword value. Typically used for short, name-like, search strings, such as an author, a project identifier, or a status value.
Keyword array	Multiple keyword values.
Link	A link to a web resource. The attribute value contains both a link address and a link text.
Model version	A link to a model version in the database. The attribute value contains a unique identifier for the model version.
Text	A value that contains text, such as for an abstract or for a summary.
User	A link to a user in the database. The attribute value contains a unique identifier for the user.

THE PRIMITIVE ATTRIBUTE PAGE

The **Primitive Attribute** page contains details on a primitive attribute in the database. The fields are:

- **Identifier.** The unique identifier of the primitive attribute.
- **Label.** The label of the primitive attribute.
The label is used as the field text of the attribute on [The Asset Page](#).
- **Description.** The description of the primitive attribute.
The description is shown as a tooltip on the field text of the attribute on [The Asset Page](#).

- **Widget.** The type of widget used when editing the primitive attribute.
- **Value.** The type of value that can be set for the primitive attribute.
- **Allowed values.** The set of allowed values for the primitive attribute. This field is only shown for [Widget Types](#) that support a set of allowed values.

Click **Edit** to edit the primitive attribute, including possibly changing its widget type and value type or adding allowed values.

Changing a primitive attribute from one widget type to another, while leaving the value type and, when applicable, any allowed values unchanged, is always harmless — assets that use the primitive attribute will simply change how their data is displayed and edited. The data itself is left untouched even when saving a new version of the asset.

Changing the value type of a primitive attribute should be done with care. If you open and subsequently save a new version of an asset that uses a primitive attribute whose value type has been changed, Model Manager will try to automatically convert the data value to the new type using natural conversion rules. The save will be canceled if there is a conversion error, in which case the attribute value would first need to be manually adjusted. If a conversion is known to be impossible — for example, changing the value type from **Date** to **Attachment** — Model Manager will simply remove the value.

Click **Delete Permanently** if you want to permanently delete the primitive attribute in the database. The attribute will be automatically removed from any asset type using it. Any assets with data for the primitive attribute will, however, be left untouched until you save a new version of the asset (in which case the attribute data will simply be excluded from the new version).



If you permanently delete a primitive attribute by accident, and do not want to go to the trouble of restoring your database from backup, simply create the attribute anew and add it to the applicable asset types. Any assets that have been updated in the interim will need to have their lost attribute data manually recovered from older versions — see also [Asset Version History](#).

Composite Attributes

You can combine primitive attributes as members of a *composite attribute*. This combination can either be done in *simple mode* or in *table mode*. In simple mode, you can store a single composite data entry in the composite attribute, such that the data

entry is a combination of one value for each primitive attribute member. In table mode, you can store multiple such data entries in the composite attribute — when displayed and edited, each primitive attribute becomes a table column and each data entry a table row.



See [Adding New Types of Data Using Attributes](#).

When a Model Manager server database is created, two composite attributes in table mode — **Attachments** and **Model Versions** — are automatically created for you. They correspond, respectively, to a list of uploaded file attachments and a list of linked model versions.

The **Composite Attributes** page, opened by clicking **Composite Attributes** in the **Database** navigation sidebar, shows a table with all composite attributes in the database. The table columns are:

- The **Label** column — the label of the composite attribute.
- The **Description** column — the optional description of the composite attribute.

Click on the label of a composite attribute in the table to show more details for that composite attribute. Click the **Add** button to add a new composite attribute.

ADDING COMPOSITE ATTRIBUTES

To add a new composite attribute:

- 1 Write an identifier for the composite attribute in the **Identifier** field. This identifier must be unique among the set of all attributes — primitive and composite — and start with a Latin alphabet character followed by zero or more additional such characters or an underscore.

The identifier of a composite attribute is used in the Model Manager search syntax when filtering on attribute values — see [Searching on Asset Attributes](#).

- 2 Write a label for the composite attribute in the **Label** field. The label is shown on an asset page as a header to the primitive attribute members.
- 3 Write a description of your attribute in the **Description** field.
- 4 Select between **Simple** and **Table** mode in the **Mode** list.
- 5 Add member primitive attributes in the **Select primitive attribute** list. Added primitive attributes are shown in a table under **Member attributes** in the order they

will appear on the asset page. You can rearrange the member attributes using drag and drop — see also [Rearranging Table Rows](#).

- 6 Click **Save** to add the new composite attribute.



Defining a composite attribute in **Table** mode with a single primitive attribute is how you get a data field that can contain a list of values. Examples include a list of supplementary files or a list of linked model versions. The only exception is a list of string values for which you can use a primitive attribute with the **Keyword array** value type — see [Table 3-5](#).

THE COMPOSITE ATTRIBUTE PAGE

The **Composite Attribute** page contains details on a composite attribute in the database. The fields are:

- **Identifier.** The unique identifier of the composite attribute.
- **Label.** The label of the composite attribute.
The label is used as the field text of the attribute on [The Asset Page](#).
- **Description.** The description of the composite attribute.
- **Mode.** The mode of the composite attribute — either **Table** or **Simple**.
- **Member attributes.** The list of primitive attributes making up the composite attribute.

Click **Edit** to edit the composite attribute, including possibly changing or rearranging its primitive attribute members.

Removing primitive attribute members from a composite attribute should be done with care. If you open and subsequently save a new version of an asset that uses a composite attribute for which a member has been removed, any data for that primitive attribute member will be excluded from the new version. Similarly, if you change from **Table** mode to **Simple** mode, only the data in the first table row will be kept in a new version of an asset.

Click **Delete Permanently** if you want to permanently delete the composite attribute in the database. The attribute will be automatically removed from any asset type using it. Any assets with data for the composite attribute will, however, be left untouched until

you save a new version of the asset (in which case the data will simply be excluded from the new version).



If you permanently delete a composite attribute by accident, and do not want to go to the trouble of restoring your database from backup, simply create the attribute anew and add it to the applicable asset types. Any assets that have been updated in the interim will need to have their lost attribute data manually recovered from older versions — see also [Asset Version History](#).

Asset Management

Read this chapter to learn how the web-based asset management system included with a Model Manager server installation helps simulation engineers to collaborate on simulation projects with people in your organization who may not have access to the COMSOL Multiphysics software.

In this chapter:

- [Managing Simulation Projects Using Assets](#)
- [Customizing the Asset Management System](#)
- [Searching on Asset Attributes](#)
- [Example: A Project Asset Type](#)

Managing Simulation Projects Using Assets

A Model Manager server includes a web-based asset management system that enables users to access models and data files in Model Manager without requiring a COMSOL Multiphysics installation. Simulation engineers can, for example, share output files from simulation runs by exporting them to the Model Manager server from COMSOL Multiphysics, while other engineers can upload new versions of data files via the Model Manager server web interface — versions which are then immediately available to the simulation engineer from the COMSOL Desktop. The asset management system also enables you to version control various documents, presentations, project notes, slides, and other supplementary files and metadata related to models and data files— all while keeping everything in the same Model Manager server database.

In this section, you will learn what is included for this system by default with a new Model Manager server installation. See the next section for various ways to extend the system based on the needs of your organization.

In this section:

- [Introduction to Asset Management](#)
- [Adding a New Asset](#)
- [The Asset Page](#)
- [The Home Page](#)
- [The Model Page](#)
- [The File Page](#)
- [Adding A New File](#)

Introduction to Asset Management

Simulation engineers working in the COMSOL Desktop modeling environment often do so in the context of some broader objective. This can, for example, be the development of a new product at a company or a project at a research institute. In these contexts, the simulation work will inevitably involve people in your organization who uses other tools and software, and who may not have access to a COMSOL Multiphysics installation. An experimentalist may need to provide the simulation

engineer with new experimental data or a CAD engineer may have updated a CAD geometry. The simulation engineer, in turn, may want to share simulation results in the form of animations, images, and plots. There will also be a need to manage various documents, presentations, notes, slides, and other supplementary files and metadata related to the simulation project.

Managing and sharing all this data can be done using a variety of tools: network file systems together with shared spreadsheets, custom intranet applications, commercial document management systems, project tracking software, or product lifecycle management (PLM) software. Depending on sophistication, these tools may support access control, auditing (who updated what and when), process and workflow management (state transitions), searching and filtering, and version control.

A Model Manager server installation includes a web-based system for managing simulation projects via *assets*. This *asset management system* forms a layer on top of the existing version control system for models and data files. You may think of an asset as a container for links to your model versions, data file versions, attached supplementary files, and various custom metadata fields. You can use the asset management system as either a standalone tool or as a subcomponent to existing systems within your organization — the existing systems could then, for example, contain web links to assets in the Model Manager server.

Assets are version controlled in their own right — every time you save changes to an asset, a new asset version is stored in the database. This enables you to both track changes made to assets over time, as well as revert such changes if need be. Storing links to your models together with supplementary files and metadata on assets also gives you a searchable archive of past and present projects — all while keeping everything in the same database as your models.

You can search assets on their linked model and file versions, attachments, and any metadata fields — either via full text search or by applying filters. This is true both for predefined fields common to all assets in a Model Manager server database, as well as for fields that you have added via customization.


The asset management system uses the same user management functionality available for models and data files. Each asset has an associated owner; each asset version has a record of the user that saved it. You grant permissions to users in order to control who has access to an asset. Assets can also be organized into libraries with individual access control settings.

The asset management system also gives you direct access to the version-controlled models and data files themselves directly from the web interface. You can search for


versions using the same search functionality available in the COMSOL Desktop. You can also view the version history of these items, as well as browse their relationships. Finally, new versions of data files can be saved by uploading files directly from the web interface.

The combination of assets together with direct access to models and data files results in a powerful tool for collaboration. At the start of a simulation project, a new asset is created. By adding links to models, users can quickly navigate both to the models themselves, as well as to all input files that these models depend on, or all output files these models have generated. New input files for a simulation can be uploaded via the web interface; animations, images, plots, reports and other output files can be downloaded. Once the project is completed, the version controlled asset gives you a valuable historical record of how the project evolved from start to finish.

Adding a New Asset

The **Home** page for the asset management system is accessed by clicking either the COMSOL logo or the **Home** link () in the top navigation bar after logging in to the web interface — see [Accessing the Web Interface](#). For a new Model Manager server database, there are no assets and the page shows an empty search result.

To add a new asset:

- 1** Click **Add** () and select **New Asset** in the top navigation bar. This starts a two-page wizard for adding a new asset.
- 2** On the first page of the wizard, write a title for the new asset in the **Asset title** field. Unlike all other fields on an asset, the title field is mandatory. Click **Continue**.
- 3** On the second page of the wizard:
 - a** Write a description for the asset in the **Description** field.
 - b** Click **Choose thumbnail** to browse for a thumbnail image. Write a caption for the image in the **Caption** field.
 - c** To add a link to a model version in the database, click **Add Row** under **Model versions**. Click **Search for Model Version** in the added table row to open a dialog box in which you can search for a model version — see [Searching for Model Versions to Add](#). Once you have found the model version to add, click the **Add to Asset** button.
 - d** To upload a file from your file system, click **Add Row** under **Attachments**. In the added table row, either drag and drop a file to the dotted rectangular area or click **Click here to upload** to browse for a file on your computer.

- 4 Once you have finished adding model versions and file attachments, click **Save** to add the new asset to the database.



See also [Example: A Project Asset Type](#) for a tutorial on adding and editing assets.

You can link to an arbitrary selection of model versions on an asset — both to versions of the same model, as well as versions of different models. The model versions can be selected from any repository and branch you are permitted to browse.

Which version of a model you choose to link to depends on the workflow that best suits your needs. You may, for example, create an asset at the beginning of a new project and link to the first version of a model, or set of models, relevant to the project. As the work progresses and you reach important milestones, you might add or replace certain model versions on the asset. Perhaps when the project is finished, you link to the final version of a model. There is, however, no urgency to continually update links on the asset whenever you save new versions of your models in the database — from [The Model Page](#) of a linked model version, you can quickly navigate to the latest version of the model.

There are no restrictions on the types of supplementary files you can upload on the asset. Examples include notes, word-processing documents, presentations, reports, slides, images, and videos. Files used as auxiliary data by models — for example, CAD data or interpolation functions — are best version controlled inside the repositories, though, side by side with the models. See also [The File Page](#).

The Asset Page

The **Asset** page shows the latest version of an asset by default. The toolbar at the top of the page contains the buttons:

- **Edit.** Edit the asset and save the changes as a new version — see [Editing an Asset](#).
- **Permissions.** Set permissions for the asset — see [Granting Permissions to an Asset](#).
- **Owner.** Set the owner of the asset — see [Transfer Ownership of an Asset](#).
- **History.** View older versions of the asset — see [Asset Version History](#).
- **Delete.** Delete the asset — see [Deleting an Asset](#).
- **Delete Permanently.** Permanently delete the asset — see [Deleting an Asset](#).

The asset page has a top section containing the title, description, and thumbnail image of the asset. The upper right corner shows:

- **Library.** The asset library that the asset belongs to.
A new Model Manager server installation comes predefined with a single asset library with the name **Asset library 1**. See also [Asset Libraries](#).
- **Owner.** The user currently set as the owner of the asset.
- **Saved.** The point in time when the version was saved.
- **Saved by.** The user that saved the version.

Links to other, related, assets in the asset management system are shown in the **Related assets** section. Click on a link to open the corresponding **Asset** page for the related asset.

Other data fields for the asset are shown in collapsible sections further down on the page. The default asset page has two such sections:

- **Model versions.** A table with model versions in the database that are linked by the asset. Click on the title to open [The Model Page](#) for the model version.
- **Attachments.** A table with supplementary file attachments uploaded on the asset. Click on the filename to download the file to your computer.



See [Customizing the Asset Management System](#) to learn how you can extend the asset page by adding custom data fields tailored to the needs of your projects and workflow. See [Searching on Asset Attributes](#) to learn how you can search assets by matching on those data fields.

EDITING AN ASSET

Click **Edit** to make changes to the asset. The **Asset** page is shown with various input fields for updating its data, similar to when adding a new asset. You can change the title, description, and thumbnail image of the asset, as well as add or remove model version links and attached files. None of the changes made on the page are stored in the database until you click **Save**, in which case all changes are saved collectively as a new asset version. Click **Cancel** to return the page to its read mode with all data restored as it was before the edit began.

Searching for Model Versions to Add

Click the **Search for Model Version** button in a table row to add or replace a linked model version. You can search among the latest versions of models by writing search terms in

the **Search** field — you can write both plain search words as well as any number of filter expressions using the Model Manager search syntax. Plain search words will match on the titles, descriptions, tags, and filenames of models.

Select the repository and branch to search in via the **Repository** and **Branch** lists, respectively. These lists are hidden if there is only one repository and branch in the database.



See [Searching and Filtering](#) in the *Model Manager Reference Manual*.

The search result is shown in a paginated table. Click on a title to show more details about that model version. Click the **Add to Asset** button to link the model version to the asset.

Rearranging Table Rows

You can rearrange the rows in a table by moving the mouse pointer over the crossed arrows icon in a row, pressing and holding down the mouse button to “grab” the row, moving the mouse pointer to “drag” the row to its desired location in the table, and then releasing the mouse button to “drop” the row in its new location.

RELATING ASSETS

You can add links to other assets in the asset management system on the **Asset** page as a way of *relating* them to the current asset.

- 1 Click the pen icon in the **Related assets** section.
- 2 In the **Related assets** dialog box, write the title of the asset you want to add a link to.
- 3 Select the asset from the result list.
- 4 Click **Save**.

You can repeat steps 2 and 3 to add multiple assets at once.

TRANSFER OWNERSHIP OF AN ASSET

You control who can set permissions on assets via ownerships — every asset has an owner, which is one of the [Users](#) in the database. Except for root administrators, only

the owner can change an asset's permissions. The user that creates an asset is automatically set as its initial owner.



See [Owners](#) in the *Model Manager Reference Manual*.

You can transfer the ownership of an asset to another user if you own the asset or are a root administrator. Select **Owner** in the list opened via the **Edit** button. In the opened dialog box, the **Current owner** field shows the username of the user that currently owns the asset. Under **New Owner**, select the user to transfer ownership to. Click **Save**.

GRANTING PERMISSIONS TO AN ASSET

You can change permissions for an asset if you own the asset or are a root administrator. Granting permissions to users and groups enables you to control, for example, who can view the asset or make changes to the asset. All available permissions for an asset are listed in [Table 4-1](#).



See [Granting Permissions](#) in the *Model Manager Reference Manual*.

TABLE 4-1: ALL AVAILABLE PERMISSIONS FOR ASSETS.

PERMISSION	DESCRIPTION
Open asset	Allowed to see and open an asset.
Save asset	Allowed to save versions of the asset.
Delete asset	Allowed to delete the asset.
Restore asset	Allowed to restore the asset when deleted.

To set permissions, select **Permissions** in the list opened via the **Edit** button. In the **Permissions** list in the opened dialog box:

- Select **None** to not set any permission requirements on the asset.
- Select **Public**, **Protected**, or **Private** to set a predefined permission template — see [Predefined Permission Templates](#) in the *Model Manager Reference Manual*.
- Select one of the user-defined permission templates for assets — see [Permission Templates](#).
- Select **Custom** to set up custom permissions for the asset — see [Custom Permissions](#) in the *Model Manager Reference Manual*.

Click **Save** to save the new permissions for the asset.

ASSET VERSION HISTORY

You can view older versions of an asset on the **Asset History** page. Click **History** on the **Asset** page to show a table with all versions listed in chronological order. Click on a version in the **Saved** column to see how the asset looked in that version. To restore the asset from an older version, click **Restore Version**. The older version will be automatically resaved as a new latest version.

Click the **Back** button to navigate to the **Asset** page.

DELETING AN ASSET


Click **Delete** on the **Asset** page to delete the asset. The asset will still be present in the database but no longer show up in search results by default. To restore a previously deleted asset, click **Restore**.

Select **Delete Permanently** in the list opened via the toggle button next to the **Delete** button to permanently delete the asset and all its versions.



Permanently deleting an asset cannot be undone unless you restore the Model Manager server database from a backup.

The Home Page

You reach the **Home** page of the asset management system by clicking on the COMSOL logo or the **Home** link () in the top navigation bar. The **Home** page is the default landing page when you log in to the Model Manager server web interface. Click **Assets** in the upper left corner to search for assets in the database. Click **Models and Files** to search for models and data files.



- [Searching Assets](#)
 - [Searching Models and Files](#)
-

SEARCHING ASSETS

You can search for the latest versions of assets from the **Home** page of the asset management system. Write search and filter expressions in the **Search** field and press Enter or click the magnifying glass to trigger a new search. You can write plain search words and any number of filter expressions using the Model Manager search syntax —

see [Searching on Asset Attributes](#). Plain search words will match on the title and description of assets.

Matching assets are shown in a paginated table below the **Search** field. Select **Rank**, **Title**, or **Last Modified** to change the field being sorted — sorting by rank means that those assets that best match the search expression are shown first. Toggle between ascending and descending order by clicking the vertical arrow button next to the sort field list. Select **Table View** or **Compact View** to show or hide the assets' thumbnail images in the table.

The default table columns are:

- The **Title** column — the current title of the asset.
- The **Description** column — the current description of the asset.
- The **Last Modified** column — the point in time when the asset was last modified.

Click on the title of an asset in the table to open its **Asset** page.

The Model Manager hides assets that you are not permitted to open from the search result — see also [Table 4-1](#).

Asset Filters

You can apply asset filters from the **Filters** menu — click the vertical expander bar on the left side of the **Home** page if the menu is not already visible. The available filter options for a new Model Manager server installation are:

- **Last Modified**. A date range for when the latest version of an asset was saved. Leave **From** or **To** empty to not specify a lower or upper bound for the date.
- **Last Modified By**. The user that saved the latest version of an asset. A multiple selection of users is combined with OR-logic when filtering.
- **Owner**. The user that owns an asset. A multiple selection of users is combined with OR-logic when filtering.
- **Deleted**. Optionally show assets that have been deleted. The default is to hide such assets.
- **Attachment**. The filename of an uploaded file attachment.
- **Model version**. The title of a linked model version.



The list of available filter options will grow as you extend the asset management system via customization — see [Customizing the Asset Management System](#).

Click **Apply Filters** to apply specified filter values. Click **Clear Filters** to clear all specified filter values.

Click the eye icon next to a filter option to show or hide the corresponding table column in the search result. You can also show or hide the **Title**, **Description**, **Thumbnail**, **Library**, and **Asset Type** fields as table columns.

SEARCHING MODELS AND FILES

Click **Models and Files** on the **Home** page of the asset management system to search for versions of *items* — that is, models and data files — in the Model Manager server database. Write search and filter expressions in the **Search** field and press Enter or click the magnifying glass to trigger a new search. You can write plain search words and any number of filter expressions using the Model Manager search syntax.

You can select between two search modes from the list in the upper right corner:

- **Latest Versions for Location.** Search the latest versions of items in a branch. Expand the accompanying location selector to select another branch to search in.
- **All Versions in Database.** Search all versions of items in the database. Expand the accompanying location selector to apply one or more repositories and branches as filters. Only versions saved in selected repositories and branches are returned in the search result.



[Searching Versions](#) in the *Model Manager Reference Manual*.

Matching versions are shown in a paginated table below the **Search** field. Select **Rank**, **Title**, **Saved**, **Size**, or **Computed Data** to change the field being sorted — sorting by rank means that those versions that best match the search expression are shown first. Toggle between ascending and descending order by clicking the vertical arrow button next to the sort field list.



[Sorting Search Results](#) in the *Model Manager Reference Manual*.

The default table columns are:

- The type column — an icon used to represent of the type of the version.
- The **Title** column — the title of the version.

- The **Tags** column — the tags assigned to the item that the version belongs to. The **Tags** column is only shown for the **Latest Versions for Location** search mode.
- The **Saved** column — the point in time when the version was saved.
- The **Saved by** column — the user that saved the version.
- The **Owner** column — the user that owns the item that the version belongs to.

Click on the title of a model version or file version in the table to open its **Model** page or **File** page, respectively.

The Model Manager hides versions in repositories and branches that you are not permitted to see from the search result.

Item Filters

You can filter the search result by applying item filters from the **Filters** menu — see also [Asset Filters](#). The available filter options are:

- **Title**. The title of the version.
- **Description**. The description of the version.
- **Tags**. The tags assigned to the item that the version belongs to. The **Tags** filter is only available for the **Latest Versions for Location** search mode.
- **Item Version Type**. The type of the item version.
- **Item Save Type**. The save type of the item that version belongs to.
- **Saved**. A date range for when the version was saved. Leave **From** or **To** empty to not specify a lower or upper bound for the date.
- **Saved By**. The user that saved the version. A multiple selection of users is combined with OR-logic when filtering.
- **Commit Comment**. The commit comment written when the version was saved.
- **Filename**. The filename of a model version and the filenames of the file resources of a file version.
- **File Type**. The file type extensions of the file resources of a file version.
- **Size**. The estimated disk space usage of a version when stored on the file system.
- **Computed Data**. The estimated disk space usage of the built, computed, and plotted data of a model version when stored on the file system.
- **Owner**. The user that owns the item that the version belongs to. A multiple selection of users is combined with OR-logic when filtering.

Click **Apply Filters** to apply specified filter values. Click **Clear Filters** to clear all specified filter values.

Click the eye icon next to a filter option to show or hide the corresponding table column in the search result. You can also show or hide the **Repository** and **Branch** fields as table columns.



[Item Filters](#) in the *Model Manager Reference Manual*.

The Model Page

The **Model** page shows the latest version of a model — with respect to a specific a branch — by default. The toolbar at the top of the page contains the buttons:

- **Edit.** Edit the model and save the changes as a new version — see [Editing a Model](#).
- **Versions.** View older versions of the model — see [Model Version History](#).
- **Copy Location to Clipboard.** Copy a location string identifier to your computer’s clipboard. This identifier can be used to open the model version in the COMSOL Desktop — see also [Opening Model Versions in the COMSOL Desktop](#).

The model page shows the title, description, assigned tags, filename, and thumbnail image of the model, with a placeholder image used for models lacking a thumbnail. The list of metadata fields on the left are:

- **Location.** The repository and branch that the model version is saved in.
- **Owner.** The user currently set as the owner of the model.
- **Saved.** The point in time when the version was saved.
- **Saved by.** The user that saved the version.
- **Saved in.** The COMSOL Multiphysics version that the model was saved from.



[Models](#) in the *Model Manager Reference Manual*.

MODEL VERSION REFERENCES

You can see references between the current model version and other versions in the **References** table on the **Model** page. Select **Show Referenced Versions** to list all versions

referenced by the current model version; select **Show Referencing Versions** to list all versions *referencing* the current model version.

The references are categorized by the following types:

- **Input file.** A data file version used as input by a model version. Examples include CAD data, interpolation data, and mesh data.
- **Output file.** A data file version generated as output from a model version. Examples include animations, images, plots, and reports.
- **Geometry part.** A geometry part loaded from another model version.

You can filter the table by selecting from these types in the **Reference Type** list.

The table columns are:

- The type column — an icon used to represent of the type of the version.
- The **Title** column — the title of the version.
- The **Reference Type** column — the type of version reference.
- The **Saved** column — the point in time when the version was saved.
- The **Saved by** column — the user that saved the version.
- The **Repository** column — the repository that the version was saved in.
- The **Branch** column — the branch that the version was saved in.
- The **Comments** column — the commit comment written when the version was saved.

Click on the title of a version in the table to open its **Model** or **File** page.



The versions shown in the **References** table — with **Show Referenced Versions** selected — are so-called *auxiliary data* used by the model as either an input source or an output target.



[The References Window](#) in the *Model Manager Reference Manual*.

EDITING A MODEL

Click **Edit** to make changes to some of the model's basic fields. The **Model** page is shown with input fields for updating its title, description, assigned tags, and filename.

None of the changes made on the page are stored in the database until you click **Save**, in which case all changes are saved collectively as a new model version. Click **Cancel** to return the page to its read mode with all data restored as it was before the edit began.

To save a new model version:

- 1 Click the **Save** button at the bottom of the page.
- 2 Write a commit comment in the opened **Save Model** dialog box.
- 3 Click **Save** in the dialog box.



[Model Settings](#) in the *Model Manager Reference Manual*.

MODEL VERSION HISTORY

You can view the model's version history with respect to the latest version on a particular branch on the **Versions** page. Click **Versions** on the **Model** page to show a table with versions saved on the branch sorted chronologically with most recent first. To view the version history with respect to another branch, select it from the location selector in the upper right corner.

The table columns are:

- The type column — an icon used to represent of the type of the version.
- The **Title** column — the title of the version.
- The **Saved** column — the point in time when the version was saved.
- The **Saved by** column — the user that saved the version.
- The **Repository** column — the repository that the version belongs to.
- The **Branch** column — the branch that the version belongs to.
- The **Comments** column — the commit comment written when the version was saved.

Click on a title in the **Title** column to open the **Model** page for that specific version.

Click the **Back** button to navigate to the **Model** page for the latest version.



[The Versions Window](#) in the *Model Manager Reference Manual*.

OPENING MODEL VERSIONS IN THE COMSOL DESKTOP

Users with a COMSOL Multiphysics installation can open a model version in the COMSOL Desktop:

- 1 Click the **Copy Location to Clipboard** button on the **Model** page.
A location string identifier for the model version has now been copied to the computer's clipboard.
- 2 Launch the COMSOL Multiphysics software if not already running. Select **File>Open From** in the COMSOL Desktop.
- 3 Select the **Clipboard** menu option in the **Open** window. The model version is shown as a single option in the list on the right. Click **Open**.
You may be asked to fill in your username and password for the Model Manager server.



You will get an error message if there is no database configuration for the Model Manager server database in the Model Manager workspace. Select **File>Open From>Add Database>Connect to Server Database** and fill in the connection details for the Model Manager server. See also [Connecting to a Server Database](#) in the *Model Manager Reference Manual*.



The clipboard is only available when accessing the Model Manager server web interface through a secure connection using HTTPS.

The File Page

The **File** page shows the latest version of a data file — with respect to a specific a branch — by default. The toolbar at the top of the page contains the buttons:

- **Edit**. Edit the file and save the changes as a new version — see [Editing a File](#).
- **Versions**. View older versions of the file — see [File Version History](#).
- **Download**. Download the file contents to your computer's file system. File contents for a fileset is downloaded as a compressed archive file.

The file page shows the title, description, and assigned tags of the file. The list of metadata fields on the left are:

- **Location**. The repository and branch that the file version is saved in.

- **Owner.** The user currently set as the owner of the file.
- **Saved.** The point in time when the version was saved.
- **Saved by.** The user that saved the version.
- **File size.** The total size of the file version in bytes.



Files in the *Model Manager Reference Manual*.

FILE VERSION CONTENTS

Binary and text data for the file version are shown as *file resources* in the **Contents** table. A file version containing a single file resource is referred to as a plain *file*. Examples include an interpolation text file or an animation file. A file version containing multiple file resources is referred to as a *fileset*. Examples include a CAD assembly with external component files, an image sequence, or an HTML report.

The table columns are:

- The **Filename** column — the filename of the file resource.
- The **File Size** column — the size of the file resource in bytes.
- The **Last Modified** column — the date and time when the contents of the file resource was last modified.

File resources in a file version may be organized into a hierarchy of directories. A directory is shown with a triangle symbol next to its name in the **Filename** column. Click the triangle to list the file resources and, possibly, subdirectories found inside that directory. File resources found on the top level in the table are thought of as belonging to an implicit *root directory*.

FILE VERSION REFERENCES

You can see references between the current file version and other versions in the **References** table on the **File** page similar to how such references are shown on the **Model** page — see also [Model Version References](#).

EDITING A FILE

Click **Edit** to edit the file. The **File** page is shown with input fields for updating its title, description, and assigned tags. You can also add, replace, or remove file resources for the file.

The **Contents** table is shown with an additional top table row representing the root directory when the **File** page is in edit mode. Click the **Add File** button to upload a file resource to the root directory. In the opened dialog box, either drag and drop a file to the dotted rectangular area or click **Click here to upload** to browse for a file on your computer.

Click the **Replace** button next to an existing file resource to replace its binary or text data with that of a file on your computer. The existing file resource will keep its current filename.

Click **Delete** to remove a file resource from the table.

You can also create a new folder under the root directory. Click **Add Folder** to open the **Add Folder** dialog box. Write the name of the folder in the **Title** field. Click **Add**. With the folder added to the table, you can continue by adding new files and other folders under that folder.

None of the changes made on the page are stored in the database until you click **Save**, in which case all changes are saved collectively as a new file version. Click **Cancel** to return the page to its read mode with all data restored as it was before the edit began.

To save a new file version:

- 1 Click the **Save** button at the bottom of the page.
- 2 Write a commit comment in the opened **Save File** dialog box.
- 3 Click **Save** in the dialog box.



A directory hierarchy of a file version is only stored implicitly in the database via the relative paths, with respect to the implicit root directory, of the file resources. No information is stored about the directories themselves. As a consequence, empty folders are ignored by Model Manager when saving.



A fileset with a deeply nested hierarchy is expected to be rare. The prime example of a fileset with a hierarchy is an HTML report with a main HTML file at the top and associated images inside a folder.



[File Settings](#) in the *Model Manager Reference Manual*.


FILE VERSION HISTORY

You can view the file's version history with respect to the latest version on a particular branch on the **Versions** page similar to how this history is shown for models — see also [Model Version History](#).

Adding A New File

You can save a new file to the database via the Model Manager server web interface. This is useful, for example, when you want to distribute a new input data file to simulation engineers working in the COMSOL Multiphysics software.

To save a new file:

- 1 Click **Add** () and select **New File** in the top navigation bar.
- 2 Select the target repository and branch in the location selector in the upper right corner.
- 3 Add one or more file resources for the new file version to the **Contents** table by uploading them from your computer — see also [Editing a File](#).
The filename of the first file resource added to the table will be automatically used in the **Title** field.
- 4 Either keep the suggested title in the **Title** field or write a custom title.
- 5 Optionally write a description of the file in the **Description** field. You can also select tags to assign the file in the **Tags** list.
- 6 Write a commit comment in the opened **Save File** dialog box.
- 7 Click **Save**.

The **File** page is automatically opened with the saved version of the new file.

Customizing the Asset Management System

As you may have noticed, the basic **Asset** page provided with a new Model Manager server database is somewhat rigid when it comes to adding data. You can, for example, add text to the **Description** field, link to model versions in the **Model versions** section, and upload supplementary files in the **Attachments** section. Other than that, there is not much flexibility.

In this section, you will learn how the asset management system can be customized so that more types of data can be added to an asset — either by extending the asset page predefined for a Model Manager server database or by defining new types of assets. You will also learn how assets can be organized into asset libraries.

- [Adding New Types of Data Using Attributes](#)
- [Adding New Types of Assets](#)
- [Controlling Access to Assets Using Libraries](#)



See also [Example: A Project Asset Type](#) for an introductory tutorial that includes customization of the asset management system.

Adding New Types of Data Using Attributes

You can extend the available fields on an asset page by defining new *primitive attributes*. A primitive attribute is a data field that can hold a specific *value type*. Examples include (see [Table 3-5](#) for the complete list):

- A **Keyword** value. A project identifier, author, search label, project status, or some other name-like metadata.
- A **Text** value. An abstract, project notes, or some other multiple-paragraph text.
- A **Date** value. A publication date, review date, or some other date.
- A **Boolean** value. An approved flag or some other true or false choice.
- A **Link** value. A link to a web page or some other web resource.
- An **Attachment**. Documentation files, presentations, slides, videos, or any other type of supplementary file.

- A **Model version**. A link to a version-controlled model stored in the Model Manager database.
- A **File version**. A link to a version-controlled data file stored in the Model Manager database.
- A **User**. A link to a user in the Model Manager database.

The new attributes are added to sections on the **Asset** page, either to one of the existing sections or by defining new sections for the page. Each primitive attribute has an *attribute label* which is shown next to the attribute value when viewing or editing the asset. Both the sections and the primitive attributes inside each section can be rearranged as you see fit.

You define how the values of a primitive attribute are edited by selecting a *widget type*. Examples include (see [Table 3-4](#) for the complete list):

- An **Input field** for writing a keyword value in a single-line input field.
- A **Text area** for writing text, possibly split into several paragraphs, in a rectangular text area.
- A **Date picker** for selecting a date from a calendar.
- A **Check box** for selecting a Boolean true or false value.
- A **Combo box** or **Radio button** for selecting a single value among a set of allowed values. This could be a status flag or some other predetermined set of options.
- A **Check box list** or **List box** for selecting multiple values among a set of allowed values. This could be product labels or some other predetermined set of options.
- A **Hyperlink** for specifying a web page or some other web resource.
- A **File upload** for browsing and selecting a file on the file system to be uploaded as an attachment.
- A **Model version picker** for searching and selecting a version-controlled model in the Model Manager database to add as a link.
- A **File version picker** for searching and selecting a version-controlled data file in the Model Manager database to add as a link.
- A **User picker** for searching and selecting a user in the Model Manager database.

Given a particular widget type for a primitive attribute, only a subset of all value types are available to select from. A **Date picker**, for example, can only store a **Date** value and a **File upload** can only store an **Attachment**.

The primitive attributes also show up in the [Asset Filters](#) menu on the **Home** page. You can apply these attributes as search filters via filter widgets analogous to the ones used when editing.

A Model Manager database comes predefined with two primitive attributes, **Model version** and **Attachment**, whose attribute labels have been set to be identical to their corresponding value type. They appear as table columns in the two sections, **Model versions** and **Attachments**, on the predefined **Asset** page.



See [Primitive Attributes](#) to learn how administrators can define new primitive attributes via the **Administration>Database** area. See [Asset Types](#) to learn how to add these primitive attributes to sections on the **Asset** page.



See [Searching on Asset Attributes](#) to learn how you can search and filter on your custom data fields.

PRIMITIVE ATTRIBUTES OF VERSION TYPE

Primitive attributes of value type **Model version** or **File version** are special in that they only store a link to a fixed version of a model or data file in the database. These attributes are displayed on the **Asset** page with the version's title, the point in time when the version was saved, the user that saved the version, and the item version type of the version. You can click on the title to open the corresponding **Model** or **File** page for the version.

You edit these primitive attributes by searching and selecting a model or data file version in the database. See [Searching for Model Versions to Add](#) for editing an attribute of **Model version** value type. Editing an attribute of **File version** value type works identically.

GROUPING PRIMITIVE ATTRIBUTES INTO COMPOSITE ATTRIBUTES

You can group one or more primitive attributes together by adding a *composite attribute*. The composite attribute can be defined using one of two *modes*:

- A **Simple** mode. This is useful if a collection of primitive attributes “logically” belong together. The primitive attributes appear under a common header given by the label of the composite attribute. An example of such a composite attribute is a journal

reference consisting of primitive attributes for the author, publication, URL, and year.

- A **Table** mode. This is useful if the values of a collection of primitive attributes belong together *and* you want to add multiple data entries for the primitive attributes when editing assets. The primitive attributes appear as columns in a table. Each data entry is then a row in the table. An example of such a composite attribute is a table of journal references with the listed primitive attributes as columns.

A Model Manager database comes predefined with two composite attributes, **Model versions** and **Attachments**, whose labels are the pluralized forms of the (single) primitive attributes they contain. Both use the **Table** mode.



See [Composite Attributes](#) to learn how administrators can define new composite attributes via the **Administration>Database** area. See [Asset Types](#) to learn how to add these composite attributes to sections on the **Asset** page.

Table Filters

If you add a primitive attribute with a set of allowed values — for example, a primitive attribute with a **Radio button** for editing — to a composite attribute using **Table** mode, that attribute will also be available as a table filter on the **Asset** page. Select check boxes in the list to only show table rows with the corresponding attribute values. Clear all check boxes to not apply a filter on the attribute.

Adding New Types of Assets

You can customize which attributes are available on an asset page by defining new *asset types*, each asset type having its own list of sections and attributes. This is useful if you want to use the asset management system to store different types of content. You could, for example, add a *Project* asset type, an *Internal Training Course* asset type, or a *Papers and Presentations* asset type — each with their own collection of attributes.

When there is more than one asset type in the database, a list of available asset types appear on the first page of the **New Asset** wizard — see [Adding a New Asset](#). There is also an **Asset Type** filter option in the [Asset Filters](#) menu on the **Home** page.



You cannot change the assigned asset type after you have created an asset.

A Model Manager server database comes predefined with a single asset type, simply named **Asset**. It has two sections, **Model versions** and **Attachments**, whose labels are the same as the predefined composite attributes they contain — see [Grouping Primitive Attributes into Composite Attributes](#).



See [Asset Types](#) to learn how administrators can define new asset types via the **Administration>Database** area.

Controlling Access to Assets Using Libraries

You can organize your assets into different *asset libraries*. This is useful, for example, if you want to set up permissions that differ between various collections of assets — perhaps hiding sensitive assets from all but a select group of users.



Assets in different asset libraries can use the same asset type.

When there is more than one asset library in the database, a list of available libraries appear on the first page of the **New Asset** wizard — see [Adding a New Asset](#). There is also a **Library** filter option in the [Asset Filters](#) menu on the **Home** page.



You cannot move an asset from one asset library to another.

A Model Manager server database comes predefined with one asset library, **Asset library I**. You can rename the library in the **Administration>Database** area — see [The Asset Library Page](#).



See [Asset Libraries](#) to learn how administrators can add new asset libraries, as well as grant permissions to these libraries, via the **Administration>Database** area.

Searching on Asset Attributes

In this section, you will learn how to search for assets in a Model Manager server database. You will see how you can match assets by performing full text searches and by applying various filter criteria. Filters can target the predefined fields that are common to all assets as well as any attributes that have been added by [Customizing the Asset Management System](#).

In this section:

- [Full Text Search](#)
- [Filters](#)
- [Asset Search Syntax](#)

Full Text Search

A fast way for you to find assets in a Model Manager server database is to write individual search words, separated by spaces, in the **Search** field on either the **Home** page or, for other pages, in the top navigation bar. The search words are matched with the title and description fields of an asset.

The Model Manager combines all search words that you write with **AND**-logic. In practical terms, this means that the assets in the search result are such that *all* search words are found somewhere in the title or description of the asset.

You can use a wildcard asterisk in a search word to match zero or more arbitrary characters. Appending, for example, a wildcard to a search word will match that word against the beginning of a word or the whole word in the searched text. You can also match on phrases — that is, multiple words in a sequence — by enclosing the search words in quotation marks.



See [Wildcard Matching](#) and [Phrase Matching](#) in the *Model Manager Reference Manual*.

Filters

You can narrow down a search result by applying filters, either by specifying filter options in the [Asset Filters](#) menu or by writing filter expressions in the **Search** field using the Model Manager search syntax — see also [Asset Search Syntax](#).



Filters match in a case-insensitive way in the Model Manager.

The available filters can be categorized based on the types of fields that they match on. These field types affect how the field values specified in the filters are interpreted when searching the database. Text fields and keyword fields, for example, are used for text and name-like search data, respectively. A date or numeric field type enables you, for example, to match either exactly or in a range, while a selection field enables you to match values in a predetermined set.



See [Field Types](#) in the *Model Manager Reference Manual*.



To include spaces in a keyword field's value, precede each space using a backslash — see [Keyword Field](#) and [Escaping Reserved Characters](#) in the *Model Manager Reference Manual*. No such escaping is necessary when searching a text field.

ASSET FIELD FILTERS

The field filters common to all assets are:

- [Title](#)
- [Description](#)
- [Library](#)
- [Asset Type](#)
- [Last Modified](#)
- [Last Modified By](#)
- [Owner](#)
- [Deleted](#)

Title

The **Title** filter is a text field filter that matches on the title of an asset. You may find this filter useful when the searched title contains a word that is also commonly found

in descriptions, such that performing a [Full Text Search](#) would yield too many results. The filter is only available using the Model Manager search syntax — see [Table 4-3](#).

Description

The **Description** filter is a text field filter that matches on the description of an asset. Similar to a [Title](#) filter, you may find this useful when the searched description is also a common title. The filter is only available using the Model Manager search syntax — see [Table 4-3](#).

Library

The **Library** filter is a selection filter that matches on the name of the asset library that an asset belongs to.

Asset Type

The **Asset Type** filter is a selection filter that matches on the name of the asset type assigned to an asset.

Last Modified

The **Last Modified** filter is a date field filter for when the latest asset version was saved.

Last Modified By

The **Last Modified By** filter is a selection field filter on the user that saved the latest asset version. Write the name or display name of the user. A space in a searched name must be escaped by preceding the space with a backslash.

Owner

The **Owner** filter is a selection field filter on the current owner of an asset. Write the name or display name of the user. A space in a search name must be escaped by preceding the space with a backslash.

Deleted

The **Deleted** filter is a Boolean field filter on whether or not an asset is deleted.

ATTRIBUTE FIELD FILTERS

Filter options for attributes appear alphabetically in the lower part of the [Asset Filters](#) menu on the **Home** page. Similar to [Asset Field Filters](#), each attribute filter option has an associated field type that affects how the values are interpreted when searching the database. The assigned field type depends on the value type of the primitive attribute, see [Table 4-2](#).

Filters on primitive attributes of a **Date** or **Number** value type are specified as a range in the [Filters](#) section. Primitive attributes with a set of allowed values are filtered by selecting a subset of values from a list.

A primitive attribute with the **Link** value type can be matched on either the link address or the link text. Only the link text can be specified in the [Asset Filters](#) menu — searching on the link address field is available via the Model Manager search syntax, see [Table 4-4](#).

A primitive attribute with the **Attachment** value type can be matched on multiple fields, including the filename, file size, and last modified date of the uploaded file. Only the filename can be specified in the [Asset Filters](#) menu — searching on other fields is available via the Model Manager search syntax, see [Table 4-5](#).

A primitive attribute with the **File version** or **Model version** value types can be matched on either the identifying key or the title of the linked version.

A primitive attribute with the **User** value type can be matched on either the identifying key or the name or display name of the user.

TABLE 4-2: THE (MAIN) FIELD TYPE ASSIGNED TO EACH PRIMITIVE ATTRIBUTE VALUE TYPE.

PRIMITIVE ATTRIBUTE VALUE TYPE	FIELD TYPE
Attachment	Multiple types
Boolean	Boolean
Date	Date
File version	Text
Integer	Numeric
Keyword	Keyword
Keyword array	Keyword
Link	Text
Model version	Text
Text	Text
User	Text

Asset Search Syntax

The filter functionality in the asset management system of a Model Manager server is based on the same tailor-made search syntax used when searching models and their

contents. The syntax enables you to write filter queries that find assets whose asset and attribute fields satisfy arbitrarily complex constraints.

In this section, you will learn how you can formulate such custom filter queries. You will see how to write simple expressions for filters on single asset fields and primitive attribute fields as well as how to nest field expressions using the primitive attribute members of a composite attribute. For more details on the search syntax, see [The Model Manager Search Syntax](#) in the *Model Manager Reference Manual*.

BASIC ASSET FIELD EXPRESSIONS

An asset filter consists of one or more *field expressions* combined with Boolean operators — for example, AND, OR, and NOT — and other grouping operators. Each field expression specifies which *field* is searched and the *field value* being searched on.

You write an asset field expression using an @-notation of the general form:

```
@<field-name>:<field-value>
```

with *<field-name>* equal to the name of one of the available asset fields in [Table 4-3](#), and *<field-value>* the value being filtered on. Write, for example,

```
@title:crane
```

to find assets whose title contains the word `crane`. To match on several search words, enclose the words with parentheses. Write

```
@title:(mounted crane)
```

to find assets whose title contains the words `mounted` and `crane`.

A space between two search words is automatically interpreted as a Boolean AND in the search syntax. The previous expression is thus equivalent to:

```
@title:(mounted AND crane)
```

Write

```
@title:(mounted OR crane)
```

if you want to find assets whose title contain `mounted` *or* `crane`.

If you want to combine a full text search with a custom filter query, write the former first. The following is valid:

```
mounted crane @assetType:project @lastModifiedBy:alice
```

and matches on assets whose title or description contains the words `mounted` and `crane`, whose asset type is `Project`, and that was last modified by user `Alice`. The following is not valid:

```
@assetType:project @lastModifiedBy:alice mounted crane
```

and results in an error message.



See [Basic Field Expressions](#) in the *Model Manager Reference Manual*.

TABLE 4-3: FIELD EXPRESSIONS FOR ASSET FIELDS.

SYNTAX	TYPE	DESCRIPTION
@assetType:...	Selection	Matches on the name of the asset type of an asset.
@description:...	Text	Matches on the description of an asset.
@isDeleted:...	Boolean	Matches on the deleted state of an asset.
@lastModified:...	Date	Matches on the last modified date of an asset.
@lastModifiedBy:...	Selection	Matches on the name or display name of the user that last modified an asset. Escape spaces in names with a backslash.
@library:...	Selection	Matches on the name of the asset library that an asset belongs to.
@owner:...	Selection	Matches on the name or display name of the user that owns the asset. Escape spaces in names with a backslash.
@title:...	Text	Matches on the title of an asset.

PRIMITIVE ATTRIBUTE FIELD EXPRESSIONS

You can filter on primitive attribute fields using a search syntax similar to that of the predefined asset fields:

```
@<attribute-identifier>:<field-value>
```

with `<attribute-identifier>` equal to the identifier of a primitive attribute and `<field-value>` the value being filtered on. Write, for example,

```
@status:planned
```

to find all assets with the value `planned` for a hypothetical primitive attribute `status` of **Keyword** value type. Similarly, the expression

```
@week_estimate:[5 TO 10]
```

matches all assets with a range of values for a hypothetical primitive attribute `week_estimate` of **Integer** value type.



A primitive attribute field expression matches both on top-level primitive attributes as well as primitive attributes that are members of composite attributes. See [Composite Attribute Field Expressions](#) to only match on the latter.



If the attribute identifier is the same as one of predefined field names in [Table 4-3](#), precede the identifier with a backslash. Thus, the field expression `@title:<field-value>` matches on the title of an asset; the field expression `@\title:<field-value>` matches on the attribute with identifier `title`.

A field expression for a primitive attribute of an **Attachment** value type matches on the filename. Write, for example,

```
@product_manual:truck\ mounted\ crane.pdf
```

to find all assets with an uploaded file `truck mounted crane.pdf` for a hypothetical **Attachment** primitive attribute `product_manual`.



The filename field of an attachment has a keyword field type, which means that spaces must be escaped by a backslash.

To search on other file properties of an **Attachment**, write

```
@<attribute-identifier>{@<field-name>:<field-value>}
```

with `<attribute-identifier>` equal to the identifier of the **Attachment** attribute and `<field-name>` equal to the name of one of the available fields in [Table 4-5](#). Write, for example,

```
@product_manual{@fileType:pdf @lastModified:[* TO 12/31/20]}
```

to find all assets with product manuals of PDF type that were created before 2021.

TABLE 4-4: FIELD EXPRESSIONS FOR PRIMITIVE ATTRIBUTES OF LINK VALUE TYPE.

SYNTAX	FIELD TYPE	DESCRIPTION
@text:...	Text	Matches on the link text of the link.
@url:...	Keyword	Matches on the URL address of the link.

TABLE 4-5: FIELD EXPRESSIONS FOR PRIMITIVE ATTRIBUTES OF ATTACHMENT VALUE TYPE.

SYNTAX	FIELD TYPE	DESCRIPTION
@filename:...	Keyword	Matches on the filename of the attachment.
@fileType:...	Keyword	Matches on the file type extension of the attachment.
@lastModified:...	Date	Matches on the last modified date of the attachment.
@sizeInBytes:...	Numeric	Matches on the size in bytes of the attachment.

COMPOSITE ATTRIBUTE FIELD EXPRESSIONS

You can filter on primitive attributes that are members of composite attributes using a syntax of the form:

```
@<composite-identifier>{@<primitive-identifier>:<field-value>}
```

with *<composite-identifier>* equal to the identifier of a composite attribute, *<primitive-identifier>* equal to the identifier of a primitive attribute that is a member of the composite attribute, and *<field-value>* equal to the value being filtered on. Write, for example,

```
@journal_reference{@conference:paris @year:2020}
```

to find all assets with the values Paris and 2020 for the two primitive attributes, conference and year, belonging to the composite attribute journal_reference. This syntax is especially useful when you want to match on multiple cells in a table row corresponding to a composite attribute in **Table** mode. Write, for example,

```
@models{@model:(mounted crane) @product_version:11.2}
```

to find all assets with a table row in a hypothetical models composite attribute table in which the model column has a model version title containing mounted crane, and the product_version column has the value 11.2.

Example: A Project Asset Type

This section showcases some of the tools available in the asset management system of a Model Manager server using the example of a hypothetical *Project* asset type. You will learn how to customize the asset management system by defining new attributes that may, for example, be relevant to a project in a product development context. These attributes are added to a new asset type to be used by project-like assets. You will also learn how to create, edit, and search such assets, including working with linked model versions, supplementary files, and other metadata added to the assets.

Some of the steps in the tutorial involves the use of the Model Manager in the COMSOL Multiphysics software — you may skip these steps if you do not have access to a COMSOL Multiphysics installation.

- [Defining Project Attributes](#)
- [Adding the Project Asset Type](#)
- [Adding Projects](#)
- [Linking to Simulation Models in the Database](#)
- [Viewing Older Versions](#)
- [Finding Projects Using Attribute Filters](#)
- [Opening Linked Simulation Models in the COMSOL Desktop](#)



See [Working with Models in Databases](#) in the *Model Manager Reference Manual*.

Defining Project Attributes

The projects saved in the asset management system will have fields for the following collection of data:

- **Status.** One of `Planned`, `Ongoing`, or `Completed` for the status of a project.
- **Work group.** A label for the group within the organization working with a project.
- **Documentation.** An uploaded file that serves as documentation for a project.
- **Presentation.** An uploaded file used during a presentation related to a project.

- **Presented.** The date when a presentation was held.
- **Model version.** A simulation model being used in a project.

Except for the last field, these fields are not available in the default asset management system included with a new Model Manager server database. The first task will therefore be to extend the asset management system by defining new primitive attributes that can hold the corresponding data.



You need to either be a root administrator or have been granted the **Manage asset types** database permission to customize the asset management system.

ADDING YOUR FIRST PRIMITIVE ATTRIBUTE

- 1 Select **Database** in the **Administration** menu (⚙️) in the upper-right corner.
- 2 Click **Primitive Attributes** in the **Database** navigation sidebar.
- 3 Click **Add**.
- 4 Write `status` in the **Identifier** field.
- 5 Write `Status` in the **Label** field.
- 6 Write `The current status of a project` in the **Description** field.
- 7 Select **Combo box** in the **Widget** list.
- 8 Select **Keyword** in the **Value** list.
- 9 In the **Allowed Values** field, write `Planned`. Click **Add**. Repeat this with `Ongoing` and `Completed`.
- 10 Click **Save**.

You have created your first primitive attribute. As its label and description indicates, it can be used to track the current status of a project.



There will already be two primitive attributes on the **Primitive Attributes** page — **Attachment** and **Model version**. These are automatically created for a new Model Manager server database.

As currently defined, the **Status** primitive attribute is edited on an **Asset** page by selecting a value in a **Combo box** widget — a list of options that you can show or hide by clicking on the widget. This is useful when there are many allowed values to select among. In our case, though, the **Radio button** widget may be more suitable:

- 1 Click on the **Status** primitive attribute on the **Primitive Attributes** page.
- 2 Click **Edit**.
- 3 In the **Widget** list, select **Radio button**.
- 4 Click **Save**.

The **Radio button** widget always shows all options when editing — useful in this case, as there are only three options to choose from.

ADDING MORE PRIMITIVE ATTRIBUTES

You will use a simple input field that stores a keyword value that specifies the work group within the organization that is involved with the project. Another reasonable widget for such a field would be a combo box with a predefined list of available work groups.

- 1 Click **Add** on the **Primitive Attributes** page.
- 2 Write `work_group` in the **Identifier** field.
- 3 Write `Work group` in the **Label** field.
- 4 Write `Organizational unit` in the **Description** field.
- 5 Select **Input field** in the **Widget** field.
- 6 Select **Keyword** in the **Value** field.
- 7 Click **Save**.

Add the remaining primitive attributes to be used by the project assets.

- 1 Click **Add** on the **Primitive Attributes** page.
- 2 Write `documentation` in the **Identifier** field.
- 3 Write `Documentation` in the **Label** field.
- 4 Write `Documentation file` in the **Description** field.
- 5 Select **File Upload** in the **Widget** list.
The **Value** field automatically defaults to **Attachment**.
- 6 Click **Save**.

Repeat this with `presentation`, `Presentation`, `Presentation file`, and **File Upload** for the **Identifier**, **Label**, **Description**, and **Widget** fields, respectively. Finally, repeat with `presented`, `Presented`, `Date of presentation`, and **Date picker**.

The last primitive attribute, **Model version**, already exists. It is used by the default asset type that is automatically created for a new Model Manager server database. You will reuse this attribute for your project asset type as well.



Primitive Attributes

ADDING TABLES

The primitive attributes of **File upload** widget type — **Documentation** and **Presentation** — can store a single uploaded file. To enable storing multiple uploaded files on your projects, you will add these attributes as columns in two separate tables.

- 1 Click **Composite Attributes** in the **Database** navigation sidebar.
- 2 Click **Add**.
- 3 Write `documentation_files` in the **Identifier** field.
- 4 Write `Documentation files` in the **Label** field.
- 5 Write `Documentation files` in the **Description** field.
- 6 Select **Table** in the **Mode** list.
- 7 In the **Select primitive attribute** list, select the **Documentation** attribute.
- 8 Click **Save**.

The **Documentation files** composite attribute is a table with a single column. Each table row can store one uploaded documentation file.

Any presentation held for a project has a corresponding presentation file and a date when the presentation was given. To store this data, add a table with two columns:

- 1 Click **Add** on the **Composite Attributes** page.
- 2 Write `presentations` in the **Identifier** field.
- 3 Write `Presentations` in the **Label** field.
- 4 Write `Presentations held` in the **Description** field.
- 5 Select **Table** in the **Mode** list.
- 6 In the **Select primitive attribute** list, select the **Presentation** attribute. Also select the **Presented** attribute.

7 Click **Save**.



There will already be two composite attributes on the **Composite Attributes** page — **Attachments** and **Model versions**. These are automatically created for a new Model Manager server database.

You will use the existing **Model versions** composite attribute to link multiple model versions on a project.



Composite Attributes

Adding the Project Asset Type

The primitive and composite attributes you have added form the building blocks for your project assets. The final step is to combine them into a new project asset type, including defining how the attributes are laid out on the **Asset** page for a project.

- 1 Click **Asset Types** in the **Database** navigation sidebar.
- 2 Click **Add**.
- 3 Write **Project** in the **Name** field.
- 4 Write **PROJ** in the **Alias** field.

The alias **PROJ**, combined with a unique integer, is used in web links for project assets — the first project you create will get the identifier **PROJ-1**. Once the new asset type has been saved to the database, the alias cannot be changed.

Attributes are shown in collapsible sections on an asset page. You will add three sections for the project asset type:

- 5 Under **Sections**, write **Information** in the **Title** field. In the **Add attribute** list, select the **Status** attribute. Also select **Work group**.
- 6 Click the **Add Section** button to add a second section. Write **Files** in the **Title** field. In the **Add attribute** list, select **Documentation files**. Also select **Presentations**.
- 7 Click the **Add Section** button to add a third and final section. Write **Simulation models** in the **Title** field. In the **Add attribute** list, select **Model versions**.

8 Click **Save**.



There will already be a default asset type on the **Asset Types** page. This asset type is automatically created for a new Model Manager server database. It uses the two predefined composite attributes — **Attachments** and **Model versions**.

Click the project asset type on the **Asset Types** page to open its details page. You should see three sections containing five attributes in total — two primitive attributes and three composite attributes. Click on the **Presentations (presentation)** link to open its details page. You should see the primitive attributes **Presentation** and **Presented**, the two members of the **Presentations** composite attribute.



Asset Types

Adding Projects


With the new asset type in place, it is time to start adding projects to the database. You will add two projects that correspond to the development of two hypothetical products — a wrench and a busbar. These examples demonstrate how you can add, edit, and search for assets in your customized asset management system.



See also *Example 1: Structural Analysis of a Wrench* and *Example 2: The Busbar — A Multiphysics Model* in *Introduction to COMSOL Multiphysics*.

- 1 Click **Add** () and select **New Asset** in the top navigation bar.
- 2 Write Combination wrench in the **Asset title** field.
- 3 Select **Project** in the list.
- 4 Click **Continue**.
- 5 In the **Description** field, write Improving the structural integrity of the combination wrench product line.
- 6 In the **Status** field, select **Ongoing**.
- 7 In the **Work group** field, write Research and Development.

8 Click **Save**.

You have now created a first asset in the database of the project asset type. Click the **Home** link () in the top navigation bar to open the **Home** page. Write **Combination wrench** in the **Search** field and press Enter. The project asset is shown in the search result table. Click on its title to return to the **Asset** page for the project.



Adding a New Asset

EDITING THE PROJECT

At the moment, the project does not contain much data other than its title, description, and status value. There are, for example, no supplementary files uploaded on the project.

1 Click **Edit** in the upper-right corner.

The **Asset** page for the project is shown in its editing state.

2 Click the **Add Row** button under the **Documentation files** table in the **Files** section.

3 Click the **Click here to upload** link and browse for a file on your computer. For the purpose of this tutorial, you may select any file on your file system to masquerade as some documentation manual.

The selected file is added as a row in the table. You can add more documentation files by repeatedly clicking the **Add Row** button under the **Documentation files** table and selecting a file via the **Click here to upload** link.

4 Click the **Add Row** button under the **Presentations** table.

5 Upload any file in the **Presentation** column much like you did for the documentation files.

6 In the **Presented** column, click the calendar and select an imagined date the presentation was held.

7 Click **Save**.

The **Files** section is now filled with a few uploaded files associated with the project. You can click the filename of a file to download it to your computer.



The Asset Page


EDITING THE ASSET TYPE

You can modify and extend an existing asset type even after you have created your first asset of that type. You may, for example, want to rearrange the layout of sections and attributes on the project asset type, or perhaps add new types of attributes to your projects.



See [Table 3-4](#) for a complete list of attribute widgets that can be added to an asset page, including the types of values that can be edited using these widgets.

As an example, you are going to add the predefined **Attachments** composite attribute to the project asset type. You may, for example, decide to use this for miscellaneous files that are neither documentation files nor presentations.

- 1 Select **Database** in the **Administration** menu () in the upper-right corner.
- 2 Click **Asset Types** in the **Database** navigation sidebar.
- 3 Click the **Project** asset type on the **Asset Types** page.

The details page for the asset type is opened. You can see the current layout of sections and the attributes they contain.

- 4 Click **Edit**.
- 5 In the **Files** section, select **Attachments** in the **Add attribute** list.

The **Attachments** composite attribute is added last to the section. You can rearrange the attributes by moving the mouse pointer over the crossed arrows icon in a table row, pressing and holding down the mouse button to “grab” the table row, moving the mouse pointer to “drag” the table row to its desired location in the table, and then releasing the mouse button to “drop” the table row in its new location.

- 6 Click **Save**.

Open the **Asset** page for your combination wrench project. An empty **Attachments** table is now visible in the **Files** section.


There are no restrictions on the types of supplementary files you decide to upload on your projects. Examples include project notes, word processing documents, reports, slides, images, and videos. Files used as auxiliary data by simulation models — for

example, CAD data or interpolation functions — are best version-controlled inside repositories, though, side by side with the models.



ADDING MORE PROJECTS

Add a second asset of the project asset type to the database:



- 1 Click **Add** () and select **New Asset** in the top navigation bar.
- 2 Write **Busbar redesign** in the **Asset title** field.
- 3 Select **Project** in the list.
- 4 Click **Continue**.
- 5 In the **Description** field, write **New design** for the busbar component.
- 6 In the **Status** field, select **Planned**. Leave **Work group** empty.
- 7 Click **Save**.





A second project asset is created in the database. The same set of attributes available for editing on the first asset are also present on this new asset.

Linking to Simulation Models in the Database



You can add links to an arbitrary selection of model versions on your project assets — both to versions of the same model, as well as versions of different models. Which version of a particular model you choose to add depends on the workflow that best suits your needs. You may add the first version of a model, or set of models, relevant to the project. As you reach important milestones in your simulation work, you may add, or replace with, newer model versions by editing the asset. Perhaps when the **Status** attribute is set to **Completed**, you add the final version of a model.

To demonstrate this linkage, you will save two models to the Model Manager server database by connecting to the database from the COMSOL Desktop modeling environment. For the combination wrench project:

- 1 Launch the COMSOL Multiphysics software.
- 2 In the **File** menu, select **Application Libraries** ().
- 3 Select **COMSOL Multiphysics>Structural Mechanics>wrench**.
- 4 Click **Open** ().

- 5 In the **File** menu, select **Save To** ()
- 6 In the **Save** window, select **Add Database** () in the list of options.
- 7 Click the **Connect to Server Database** button ()
- 8 Follow the steps in [Connecting to a Server Database](#) in the *Model Manager Reference Manual* to connect to the Model Manager server database.
After COMSOL Multiphysics has finished connecting to the Model Manager server, the server database is shown selected in the **Save** window.
- 9 Click the **Save** button () to save a first version of the model in the database.

Similarly for the busbar redesign project:

- 1 Select and open **COMSOL Multiphysics>Multiphysics>busbar** in the **Application Libraries** window.
- 2 Select **Save To** () in the **File** menu.
- 3 In the **Save** window, select the Model Manager server database in the list of options.
- 4 Click **Save** ()

Return to the **Asset** page for the busbar redesign project in the asset management system.

- 1 Click **Edit**.
- 2 Under the **Model versions** table, click the **Add row** button.
- 3 In the added table row, click the **Search for Model Version** button in the **Model version** column.
- 4 Write **busbar** in the **Search** field in the opened dialog box.
- 5 Click the **Add to Asset** button.




The dialog box is closed and the linked model version for the busbar is shown in the **Model version** column.

- 6 Click **Save**.

Open the **Asset** page for the combination wrench project and repeat the previous steps for the wrench model.

Opening Linked Simulation Models in the COMSOL Desktop

Once you have found a model version of interest on an asset, you can quickly open that model version in the COMSOL Desktop. On the **Asset** page for the combination wrench project:

- 1 In the **Model versions** table, click the title of the wrench model in the **Model version** column.
- 2 Click the **Copy Location to Clipboard** button on the opened **Model** page.
A location string identifier for the model version is automatically copied to your computer's clipboard.
- 3 In the **File** menu in COMSOL Multiphysics, select **Open From** ()
- 4 In the **Open** window, the **Clipboard** () option is automatically selected in the list.
- 5 Click the **Open** button () to open the wrench model in the COMSOL Desktop.



The clipboard is only available when accessing the Model Manager server web interface through a secure connection using HTTPS.



[The Model Page](#)

Viewing Older Versions

Every time you save an asset, a new version is stored in the database. You can see older versions of an asset via the **Asset History** page. You can also restore an older version as a new latest version — useful, for example, if you make unwanted changes to an asset by mistake.

On the **Asset** page for the combination wrench project:

- 1 Click the **History** button in the upper-right corner.
The **Asset History** page shows a table with the three saved versions of the combination wrench project.
- 2 Click the middle row in the table.
The **Asset** page is opened for that version. Notice that there is no linked model version in the **Model versions** table.
- 3 Click the **History** button once more and then click the bottom row in the table.
The **Asset** page is opened for the first version of the project. There are no uploaded files on the project.

- 4 Click the **View Latest Version** link at the top of the page.

The **Asset** page is opened for the latest version. There are both uploaded files and a linked model version present on the page, as expected.

Finding Projects Using Attribute Filters

As your collection of assets in the database grows in size, it will become important to be able to search and filter on the assets. On the **Home** page:

- 1 If not already expanded, click the vertical bar on the left side of the page to expand the **Filters** menu.

- 2 Under **Asset Type**, select **Project** in the list.

- 3 Click the **Apply Filters** button.

The search result is updated to show the two assets with the project asset type.

- 4 Under **Status**, select **Ongoing** in the list. Click **Apply Filters**.

The search result shows the ongoing combination wrench project.

- 5 Click the **Clear Filters** button to clear the two applied filters.

- 6 Under **Model version**, write `electrical heating busbar` and press Enter.

A filter on the **Model version** attribute matches on the title of the linked model version. In this case, the search result shows the busbar redesign project. You can also try a wildcard search. Test with, for example, `electric*` in the **Model version** field.

- 7 Click the **Clear Filters** button to clear the **Model version** filter.

- 8 Under **Work group**, write `research and development` and press Enter.

You may be surprised to find that no assets match the filter. The reason is that spaces must be escaped with a backslash when filtering on attributes of keyword value type. Alternatively, you can filter using a wildcard or using phrase matching.

- 9 Replace the filter value with `research\ and\ development` and press Enter. Also test with an appended wildcard, `research*`, and as a phrase, “`research and development`”.

In all three cases, you will match on the combination wrench project.

- 10 Click **Clear Filters** to clear the **Work group** filter.

You can also filter by writing a filter expression directly in the **Search** field. Write, for example, `@status:planned` to match on the busbar redesign project.



See [Searching on Asset Attributes](#) to learn more on how you can search and filter assets on their attributes.

This concludes this introductory tutorial on the asset management system.

I n d e x

- A** accounts
 - adding 60
 - deleting 61
 - editing 61
- accounts (page) 60
- activating databases 80
- active database 77
- adding
 - accounts 60
 - asset libraries 100
 - asset types 102
 - assets 116
 - composite attributes 109
 - connectors 46
 - database permissions 98
 - databases 79
 - files 131
 - groups 96
 - managed resources directories 69
 - managed search index servers 72
 - managed SQL database servers 63
 - permission templates 99
 - primitive attributes 104
 - users 95
- administration
 - database 94
 - system 44
- administrator password, resetting 31
- Apache Solr
 - external 87
 - managed 71
- asset (page) 117
- asset history (page) 121
- asset libraries 136
 - adding 100
 - administrating 100
 - deleting 100
 - editing 100
 - owners 101
 - permissions 101
 - restoring 100
- asset libraries (page) 100
- asset management 114
 - customization 132
- asset types 135
 - adding 102
 - administrating 102
 - deleting 103
 - editing 103
 - permanently deleting 103
 - restoring 103
 - sections 102
- asset types (page) 102
- assets
 - adding 116
 - deleting 121
 - editing 118
 - filters 139
 - history 121
 - owners 119
 - permanently deleting 121
 - permissions 120
 - relating 119
 - restoring 121
 - searching 121
- attachments 118
- attributes
 - composite 134
 - filters 140
 - primitive 132
- authentication
 - external 52

- local 52
 - proxy 57
- auxiliary data 126
- B** backups
 - data directories 22
 - managed resources directories 71
 - managed SQL database servers 67
 - planning 16
 - preference directory 18
 - restoring from 91
- backward compatibility 93
- C** certificate 49
 - changing language 45
 - changing password 59
 - client compatibility 47
 - command options 39
 - composite attributes 134
 - adding 109
 - administrating 108
 - editing 110
 - field expressions 145
 - permanently deleting 110
 - simple mode 134
 - table mode 135
 - composite attributes (page) 109
 - connectors
 - adding 46
 - editing 50
 - permanently deleting 50
 - connectors (page) 46
 - copy location to clipboard 128
 - current database 45
- D** data directories 20
 - database administration 94
 - database configurations (page) 78
 - database permissions 98
 - database permissions (page) 97
- databases
 - activating 80
 - active 77
 - adding 79
 - backup 22
 - backward compatibility 93
 - current 45
 - deactivating 80
 - default 77
 - deleting 81
 - editing 80
 - moving 87
 - restoring 91
- deactivating databases 80
- default database 77
- deleting
 - accounts 61
 - asset libraries 100
 - asset types 103
 - assets 121
 - databases 81
 - groups 97
 - managed resources directories 70
 - managed search index servers 74
 - managed SQL database servers 66
 - permission templates 100
 - users 96
- DMZ network 40
- downloading files 128
- drag and drop 119
- E** editing
 - accounts 61
 - asset libraries 100
 - asset types 103
 - assets 118
 - composite attributes 110
 - connectors 50
 - database permissions 98

- databases 80
- files 129
- groups 97
- managed resources directories 70
- managed search index servers 74
- managed SQL database servers 65
- models 126
- permission templates 100
- primitive attributes 108
- users 96
- emailing COMSOL 10
- external authentication 52
 - group mappings 56
 - testing 56
- external authentication (page) 53
- F** field expressions 142
 - composite attributes 145
 - primitive attributes 143
- field types 139
- file (page) 128
- file resources 129
- files
 - adding 131
 - downloading 128
 - editing 129
 - linking 134
 - searching 123
- filtering 139
 - assets 139
 - attributes 140
- filters menu 122, 124
- firewalls 37
- full text search 138
- G** getting started 12
- granting permissions 120
- groups
 - adding 96
 - deleting 97
 - editing 97
 - restoring 97
- groups (page) 96
- H** home (page) 121
- I** installation
 - directory 17
 - license 26
 - license manager 27
 - options 26
 - planning 16
 - preference directory 17
 - products 26
 - server 27
 - troubleshooting 29
- installing a Model Manager server
 - in Linux 30
 - in macOS 30
 - in Windows 25
- internet resources 9
- J** JAAS 52
- K** knowledge base, COMSOL 10
- L** language (page) 45
- LDAP 54
- license manager, installation 27
- license, installation 26
- lightweight directory access protocol 54
- linking files 134
- linking models 118
- local administrative user 29
- local authentication 52
- log files 17
- logging in to the Model Manager server
 - 44
- login modules 52
- M** macOS native keychain 49
- managed Apache Solr 71

- managed PostgreSQL 62
- managed resources directories 68
 - adding 69
 - backup 71
 - deleting 70
 - editing 70
 - restoring 71
- managed resources directories (page) 68
- managed search index servers 71
 - adding 72
 - deleting 74
 - editing 74
 - restoring 75
- managed search index servers (page) 71
- managed server components 62
- managed SQL database servers 62
 - adding 63
 - backup 67
 - deleting 66
 - editing 65
 - restoring 67
- managed SQL database servers (page) 62
- memory footprint 24
- Microsoft SQL Server 83
- model (page) 125
- Model Manager server
 - administrating 44
 - example setups 25
 - installation directory 17
 - installing 16
 - log files 17
 - migrating 33
 - preference directory 17
 - securing 40
 - starting 34
 - upgrading 32
- models
 - editing 126
 - linking 118
 - opening 128
 - searching 123
- my account (page) 59
- MySQL 84
- N** network disk 20
- new asset wizard 116
- NFS 20
- O** opening models 128
- options, installation 26
- Oracle Database 85
- owners
 - asset libraries 101
 - assets 119
 - transferring ownerships 119
- P** pages
 - accounts 60
 - asset 117
 - asset history 121
 - asset libraries 100
 - asset types 102
 - composite attributes 109
 - connectors 46
 - database configurations 78
 - database permissions 97
 - external authentication 53
 - file 128
 - groups 96
 - home 121
 - language 45
 - managed resources directories 68
 - managed search index servers 71
 - managed SQL database servers 62
 - model 125
 - my account 59
 - permission templates 99
 - primitive attributes 104

- proxy authentication 57
 - users 95
 - versions 127
- password security 41
- password, changing 59
- pem files 48
- permanently deleting
 - asset types 103
 - assets 121
 - composite attributes 110
 - connectors 50
 - permission templates 100
 - primitive attributes 108
- permission templates
 - adding 99
 - deleting 100
 - editing 100
 - permanently deleting 100
 - restoring 100
- permission templates (pages) 99
- permissions
 - asset libraries 101
 - assets 120
 - granting 120
 - levels 101
 - templates 98
- phrase matching 138
- pkcs#12 keystore 48
- port number 27
- PostgreSQL
 - external 82
 - managed 62
- preference directory 17
 - migrating 33
- primitive attributes 132
 - adding 104
 - administrating 104
 - editing 108
 - field expressions 143
 - permanently deleting 108
 - value types 104
 - widget types 104
- primitive attributes (page) 104
- principals 56
- products, installation 26
- proxy authentication 57
- proxy authentication (page) 57

Q quick start guide 12

R relating assets 119

- resetting administrator password 31
- resources directories 20
- restoring
 - asset libraries 100
 - asset types 103
 - assets 121
 - databases 91
 - groups 97
 - managed resources directories 71
 - managed search index servers 75
 - managed SQL database servers 67
 - permission templates 100
 - users 96
- reverse proxy 38

S search indexes 74

- search syntax 141
- searching
 - assets 121
 - filters 139
 - full text search 138
 - items 123
 - phrases 138
 - syntax 141
 - wildcards 138
- sections 102
- secure connections 37

- securing a Model Manager server 40
 - server certificate 49
 - server, installation 27
 - simple mode 134
 - SQL databases 66
 - SQL Server 83
 - starting a Model Manager server
 - in Linux 35
 - in macOS 37
 - in Windows 34
 - system administration 44
- T**
- table mode 135
 - TCP connections 37
 - technical support, COMSOL 10
 - TLS
 - connectors 46
 - reverse proxy 38
 - TLS host configurations 47
- U**
- updating
 - accounts 61
 - asset libraries 100
 - asset types 103
 - assets 118
 - composite attributes 110
 - connectors 50
 - database permissions 98
 - databases 80
 - files 129
 - groups 97
 - managed resources directories 70
 - managed search index servers 74
 - managed SQL database servers 65
 - models 126
 - permission templates 100
 - primitive attributes 108
 - users 96
 - users
 - adding 95
 - deleting 96
 - editing 96
 - restoring 96
 - users (page) 95
- V**
- value types 104
 - examples 132
 - versions
 - history 121
 - restoring 121
 - versions (page) 127
- W**
- websites, COMSOL 10
 - widget types 104
 - examples 133
 - wildcard matching 138
 - Windows login 54
 - Windows native certificate store 48