



Teamwork Server 19.0 LTR SP4 User Guide

All material contained herein is considered proprietary information owned by No Magic, Inc. and is not to be shared, copied, or reproduced by any means. All information copyright 1998-2020 by No Magic, Incorporated, a Dassault Systèmes company. All Rights Reserved.

1 Teamwork Server Concepts 8

2 Teamwork System Design 9

- 2.1 Design of teamwork system 10
- 2.2 Comparison of repositories. 11
- 2.3 Native Repository 11
- 2.4 SVN Repository 12

3 Getting Started with Teamwork Server 13

- 3.1 Starting Teamwork Server 13
 - 3.1.1 Starting the server using GUI 14
 - 3.1.2 Starting the server without using GUI 15
- 3.2 Stopping Teamwork Server 19
- 3.3 Upgrading Teamwork Server 20
 - 3.3.1 Automatic upgrade of Teamwork Server 20
 - 3.3.2 Manual upgrade of Teamwork Server 21
- 3.4 Importing projects and users from earlier versions of Teamwork Server 23
 - 3.4.1 Replacing the projects folder 24
 - 3.4.2 Modifying the muserver.projects_directory property in muserver.properties file 25
 - 3.4.3 Changing the repository location in Administrator's Console 25
- 3.5 Migrating from the SVN repository to the Native repository type 25
- 3.6 Moving Teamwork Server 26
- 3.7 Backups of server projects 26

4 Managing Teamwork Server 27

- 4.1 Customizing Teamwork Server properties 28
 - 4.1.1 Environment Option dialog. Collaboration options group 28
- 4.2 Changing Administrator Password 29
- 4.3 User Management 30

Contents

4.3.1	User Permissions	34
4.4	Starting the Administrator's Console	36
4.5	Administrator's Console Dialog	36
4.5.1	Active Users tab	36
4.5.2	Projects tab	37
4.5.3	Log File tab	42
4.5.4	The Properties tab	43
4.5.5	Secured Connection tab	43
4.5.6	Repository tab	46
4.5.7	LDAP Integration tab	48
4.6	Administrating Server via Command Line Utility	49
4.6.1	Administrating Server via Command Line Utility	49
4.6.2	Giving commands on the input stream	50
4.7	Data Migration between Different Repositories	51
4.8	Changing the Teamwork Server Debugging Mode	51
4.8.1	Teamwork Administrator's Console - Enable Debugging	52
4.9	Configuring server properties	52
5	LDAP Support	53
5.1	Enabling LDAP Integration	53
5.2	Connection Settings	54
5.2.1	Teamwork Administrator's Console, LDAP Integration tab. Connection Settings	54
5.3	Authentication Settings	55
5.3.1	Teamwork Administrator's Console, LDAP Integration tab. Selecting Authentication Type	56
5.3.2	Authentication setting for the Simple User + Password authentication type	56
5.3.3	Authentication settings for the SASL authentication type	60
5.4	User Data Retrieval Settings	61
5.4.1	Teamwork Administrator's Console, LDAP Integration tab. User Data Retrieval Settings	62

Contents

5.5	Connection Testing	64
5.6	Subversion and LDAP Integration Working at the Same Time	64
5.7	Converting Certificates to JKS Format	64
5.8	Integrating Teamwork Server with a SSL-Enabled Active Directory (5
5.8.1	Creating a new keystore file with the KeyTool IUI (steps 2.2, 2.3, and 2.4) (6
5.8.2	Importing certificate files into the keystore file (steps 2.5, 2.6, 2.7, and 2.8)	67
6	Connecting Teamwork server via SSH encrypted tunnel	67
6.1	Configure the Teamwork Server side	68
6.1.1	The New User dialog	69
6.1.2	The New Group dialog	69
6.2	Client Side Configuration	70
7	Server Synchronization and Multisite Deployment	70
7.1	Synchronization Overview	70
7.1.1	Usage Scenarios, Multisite Deployment	71
7.1.2	Characteristics	72
7.2	Using Synchronization	72
7.2.1	Distinguishing between Local and Remote Projects	74
7.2.2	Controlling the Synchronization Scope	74
7.2.3	Controlling Synchronization Initiation	75
7.2.4	Deleted and Inaccessible Projects	75
7.2.5	Securing the Connection between Servers	75
7.3	Running Synchronization	76
7.3.1	Triggering Synchronization with an Internal Scheduler	76
7.3.2	Triggering Synchronization Using the Command Line Utility	77
8	Transferring Projects between Isolated Servers	79

This UserGuide provides information about administrating MagicDraw Teamwork Server.



Tip

For information about working with Teamwork Server from the client side, that is modeling tools developed by No Magic, Inc. - MagicDraw, Cameo System Modeler, or Cameo Enterprise Architecture, please refer to [Collaborative modeling](#)¹.

With MagicDraw Teamwork Server you can assign as many developers as needed to work simultaneously on the same project using multiple workstations. The resulting server project is saved on the server for sharing with other MagicDraw® applications. Users with administrator rights can create new users by creating a name and assigning various permissions to work on projects. The permissions assigned will determine whether the new user can update, commit, edit, create, and delete model elements, diagrams, and projects.

To enable Teamwork support, you should install and run MagicDraw Teamwork Server. Each modeling tool application is a client of Teamwork Server.

The following demo will help you understand how to work with the Teamwork Server:

- [Model Decomposition](#)². This demo introduces the functionality of model decomposition by demonstrating how to split a project and work with a part of it independently.
- [Shared Packages](#)³. This demo presents project segmentation using shared packages functionality, which allows the usage of a model part in multiple projects.
- [Teamwork Server: Add project to teamwork](#)⁴. This demo presents easy way to make a local project accessible for the Teamwork Server users.
- [Teamwork Server: Create a new user in teamwork](#)⁵. This demo presents the management of users accounts, guides through creation of a new user or edition of existing one, introduces various kinds of permissions and how they can be set to project.
- [Teamwork Server: Working in teamwork](#)⁶. This demo introduces the main functionality of teamwork, when two or more users work with the same project. You will see how to lock part of the model and work on this part individually, how to commit changes to the server and share with the other team members.
- [Teamwork Server: Project branching](#)⁷. This demo introduces the Branching functionality and presents samples of branch creation.
- [Efficient Teamwork Server Repository analysis with Project Usage Map](#)⁸. In this short demo, we will present the Project Usage Map and its use case. You will get in-depth information on the use of this powerful analysis capability.



Availability on modeling tool editions

The Teamwork Server functionality is available with modeling tool client Standard, Professional, Architect, and Enterprise editions only.

¹ <https://docs.nomagic.com/display/MD185/Collaborative+modeling>

² https://www.nomagic.com/files/viewlets/MD_viewlets_Model_Decomposition.html?ml=1&iframe=1

³ https://www.nomagic.com/files/viewlets/MD_125_viewlets_Shared_Packages_swf.html?ml=1&iframe=1

⁴ https://www.nomagic.com/files/viewlets/MD_viewlets_Teamwork_Add_Project.html?ml=1&iframe=1

⁵ https://www.nomagic.com/files/viewlets/MD_viewlets_Teamwork_Create_User.html?ml=1&iframe=1

⁶ https://www.nomagic.com/files/viewlets/MD_viewlets_Teamwork_Working.html?ml=1&iframe=1

⁷ https://www.nomagic.com/files/viewlets/MD_viewlets_Branching.html?ml=1&iframe=1

⁸ https://www.nomagic.com/files/viewlets/Project_Usage_Map.html?ml=1&iframe=1

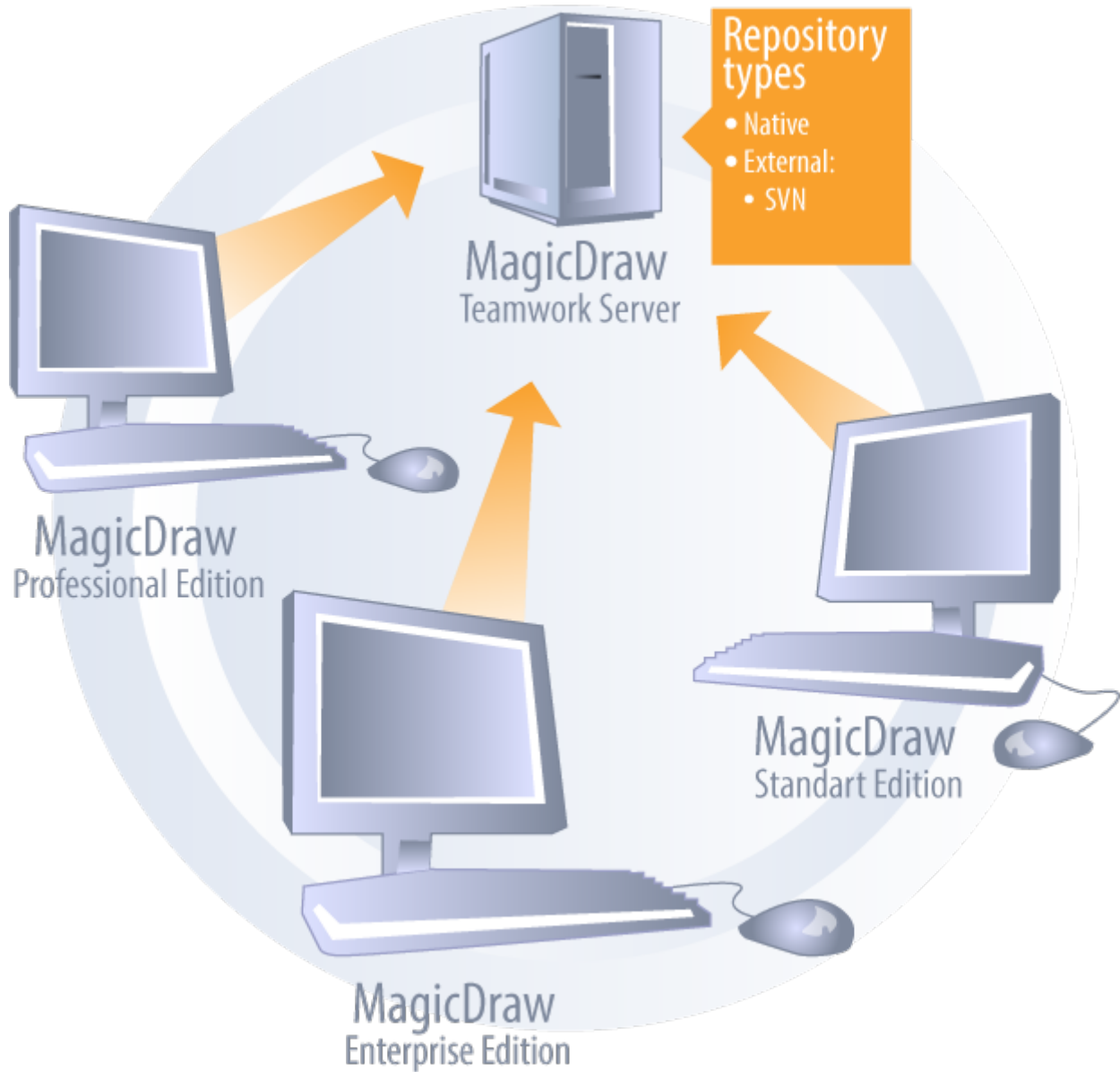
1 Teamwork Server Concepts

The following table presents Teamwork concepts.

Concept	Definition
Author	A user who has committed a new project version.
Version	A unique number assigned to the committed project. Project version numbers begin at zero (for the initial version) and increase with every new project version.
Comment	Optional description of changes in the committed version.
Tag	Information about the status of a project (approved, initially tested, etc.), or other important information.
Administrator Login	<p>The default Administrator's account in Teamwork Server is:</p> <ul style="list-style-type: none">• Login name: Administrator• Password: Administrator <p>We recommend changing the Administrator's password to prevent illegal access.</p> <p>For more information, see the Managing Teamwork Server(see page 27) section.</p>
Teamwork Server Administrator's Console	A remote connection for Teamwork Server status observation and administrative control. The server holds information about active users and loaded projects. The Administrator can shut down or restart the server, change its properties, and view log files (including debug information) for the server and separate projects.
Repository	A storage place for projects and their versions managed by the Teamwork Server.
Project category	A concept enabling visual grouping of projects in the Teamwork Server repository.
Native User	A user whose account data is stored locally, i.e. in the native Teamwork Server repository.
External User	A user whose account data (all except the login name) is stored in an external database, e.g., Subversion, or LDAP.
Used Server Project	A server project containing one or more shared packages. Used projects are created to reuse shared packages or to decompose projects into parts.

Concept	Definition
Dependency between two elements	A situation where one element (dependent element) refers to the data of another element (independent element).
Home server	A server where a project was initially created.
Domestic project	A project created on the home server
Foreign project	A project transferred from its home server after synchronization. A foreign project cannot be modified on the server to which it was transferred. However, it can be browsed, analyzed, selected for report generation, and used in other projects on that server. A foreign project can have domestic (editable) branches.

2 Teamwork System Design



2.1 Design of teamwork system

Clients communicate (using Java RMI, over TCP/IP) with Teamwork Server to retrieve projects stored on the server, edit them, and commit them to the server for storage.

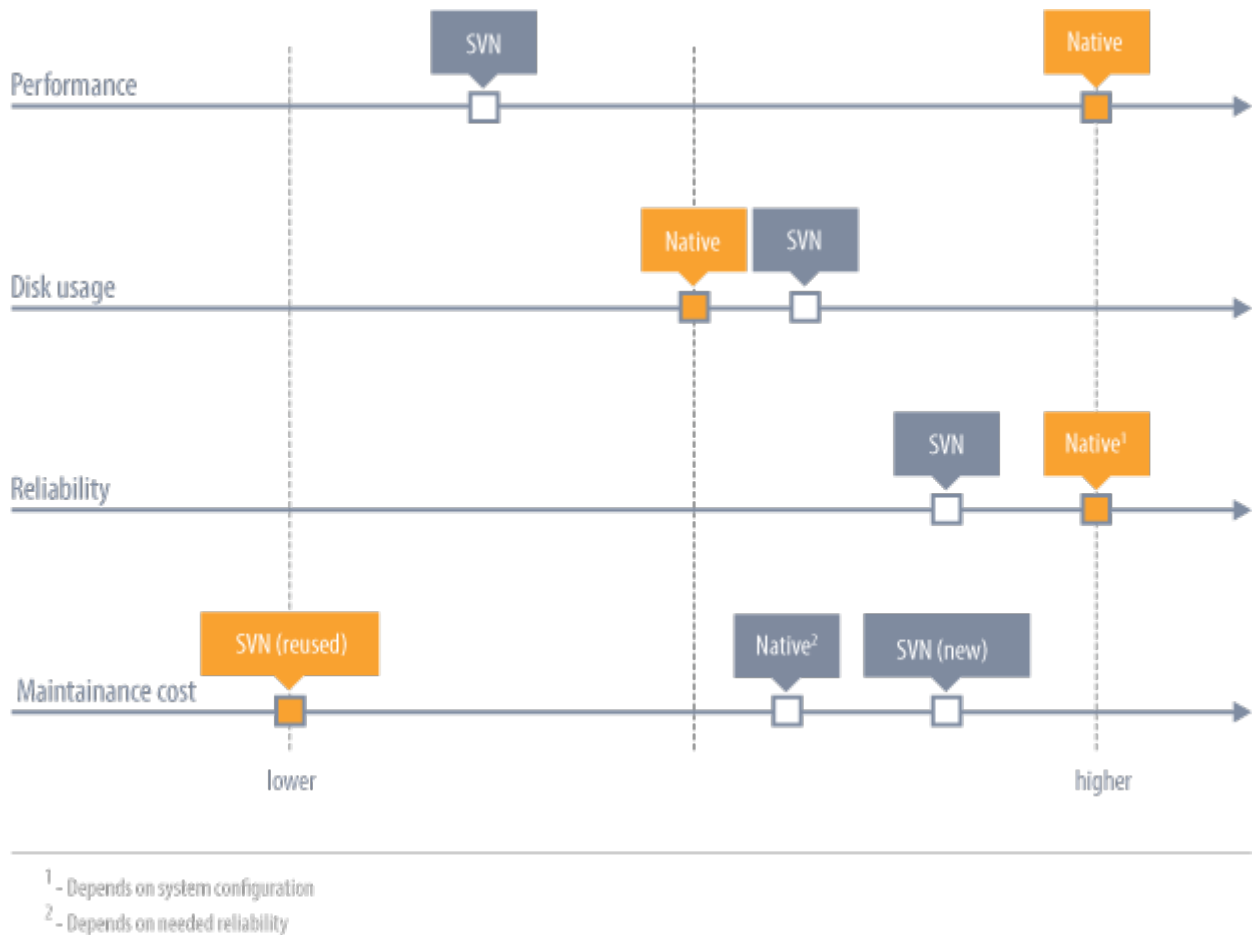
The Teamwork Server keeps track of project versions. Additionally, it performs several administrative functions to access projects, including user login and authentication, as well as checking permissions.

The Teamwork Server uses repositories for project version storage. The administrator can select any one of the supported repository types in the Teamwork Administrator's console to configure the server (for more information, see [Starting the Administrator's Console](#)). Data can migrate from one to another repository type. This functionality is also accessible from the Teamwork Administrator's console.

Two different types of repositories are supported:

- [Native](#)(see page 11) (the default MagicDraw repository type).

- [SVN](#)(see page 12) (since v12.5).



2.2 Comparison of repositories.

Regardless of the repository used, users will not feel the difference because the user workflow remains the same.

✓ Repositories

- For more information about specifying repositories, see the Administrator's Console dialog description of the [Repository tab](#)(see page 46).
- For more information about importing or exporting a project to the native repository, see the Administrator's Console dialog description of the [Projects tab](#)(see page 37).

Related Pages:

- [Native Repository](#)(see page 11)
- [SVN Repository](#)(see page 12)

2.3 Native Repository

This is a default repository type. When the Teamwork Server is first installed and started, it is configured to use the Native repository. This is the only type of repository available in versions prior to 12.5 of Teamwork Server. When the Teamwork Server is configured to use the Native repository, a directory is designated for project storage. The Server then uses its internal proprietary code to implement a versioned repository for a collection of projects. Additionally, a simple user authentication/authorization scheme is implemented in the repository to store a simple list of users and their passwords (securely encrypted using one-way encryption) in a user file. When MagicDraw users log in to the server, the Native repository uses this user file to verify these users and their passwords. Users' rights to access different projects are also described using this file.

Related Pages:

- [Teamwork System Design](#)(see page 9)
 - [Native Repository](#)(see page 11)
 - [SVN Repository](#)(see page 12)

2.4 SVN Repository

The Teamwork Server can be configured to use the SVN repository as a back-end. In this mode, the Teamwork Server retrieves and commits project versions into the SVN repository.

To use this repository type, the SVN client executable must be correctly installed on the computer where the Server runs. The Teamwork Server must be able to launch the SVN executable; the SVN executable must be accessible on the system's PATH and have appropriate permissions to execute. Supported SVN client versions are 1.4, 1.5, 1.6, 1.7, and 1.8.

All SVN repository access methods are supported:

- local access through file:// type URLs
- remote repository access through svn://, svn+ssh:// type URLs
- remote repository access through HTTP and WebDAV - http://, https:// type URLs




Passwords Based Logins

Only password based logins are supported (public/private key logins for svn+ssh:// and https:// access methods are currently not available).

Teamwork Server, with the SVN repository, supports pass-through authentication into the SVN. A pass-through authentication is used for all access methods, except the file:// method. In this case, the Teamwork Server only maintains a list of users but does not store their passwords. When a user tries to log on the Teamwork Server, the server does not verify the password. It logs on the SVN with the typed user name and password. The server performs all project update/commit actions on the repository on the user's request. Thus, if you explore the repository with SVN tools, you will see that all the changes are attributed to the correct user.

For file:// type URLs, the pass-through authentication is not possible. Teamwork Server uses the same built-in authentication method as the Native repository type, maintaining the users list with their encrypted passwords in a repository file. The server authenticates users using this file. The server

performs actions in a repository on the user's request. If the server is started as the NT service, all actions in the repository will be attributed to the Local System user (unless a different user is specified in service settings). If the server is started manually, all actions in the repository will be attributed to the user who started the server. This difference can only be seen when examining the SVN repository with SVN native tools. When looking at the project versions with the MagicDraw client, all commit actions will be attributed to the users who performed them.

 When a project file is committed into the SVN repository, the server stores auxiliary information about the project in an additional directory. For example, if you commit the MyProject.mdzip project into the server, the auxiliary information will be stored in the MyProject_files directory nearby. **Do not** delete this directory from the repository.

For the best performance, the Teamwork Server and SVN repository should have a good link between them. Optimally, Teamwork Server could run on the machine where the repository is installed.

Related Pages:

- [Teamwork System Design](#)(see page 9)
 - [Native Repository](#)(see page 11)
 - [SVN Repository](#)(see page 12)

3 Getting Started with Teamwork Server

Teamwork Server makes it easy to exchange data directly in the context of your work. It provides a central repository for storing any kind of models. Using Teamwork Server, team members can access, review, or modify the same model or even the same diagram at the same time. It supports importing, exporting, updating, branching or comparing in-server-stored models.

Server data and configuration can be managed through a separate administrative console with a minimal technical knowledge and effort.

Related Pages:

- [Starting Teamwork Server](#)(see page 13)
- [Stopping Teamwork Server](#)(see page 19)
- [Upgrading Teamwork Server](#)(see page 20)
- [Importing projects and users from earlier versions of Teamwork Server](#)(see page 23)
- [Migrating from the SVN repository to the Native repository type](#)(see page 25)
- [Moving Teamwork Server](#)(see page 26)
- [Backups of server projects](#)(see page 26)

3.1 Starting Teamwork Server

To start the server for the first time, run the file *teamwork_server.exe* from the *bin* folder and browse [the license key](#)⁹ when prompted. Note that the key will only be required the first time the server is run. For

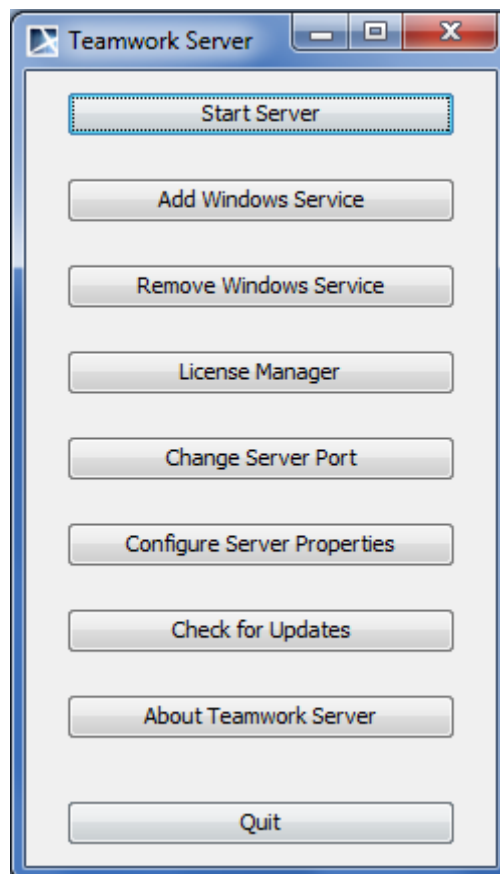
⁹ <https://docs.nomagic.com/display/NMDOC/License+Manager+dialog>

more information about node locked license activation, see at [Activating the Teamwork Server commercial license](#)¹⁰.

3.1.1 Starting the server using GUI

To start Teamwork Server

1. Do one of the following:
 - Start Teamwork Server the same way you would start any application on your operating system.
 - Run *teamwork_server.exe* in the server bin folder. The **Teamwork Server** startup dialog opens.
2. Click the **Start Server** button.




Teamwork Server startup dialog

To change the server license key

1. Run *teamwork_server.exe* in the server bin folder. The **Teamwork Server** startup dialog opens.
2. Click the **License Manager** button. The **Teamwork Server License Manager** dialog opens.

¹⁰ <https://docs.nomagic.com/display/NMDOC/Activating+the+Teamwork+Server+commercial+license>

3. On the opened dialog, click the **Select License Key Files** button to browse for a file with the Teamwork Server license key.
4. In a browser, open a new license key file.
5. Click **OK** after you have changed the server license key.

 Restart the server to apply changes. Make sure all users are logged out before restarting the server.

To change the server port for the current launch

1. Run `teamwork_server.exe` in the server bin folder. The **Teamwork Server** startup dialog opens.
2. Click the **Change Server Port** button and enter the new server port. This port is used when launching the server, or when adding the Windows service.

The Teamwork Server exports remote objects through one port: RMI registry port.

3.1.2 Starting the server without using GUI


To start Teamwork Server from the command line

1. Do one of the following:
 - Add the parameter `NOGUI` when starting the server from the command line.
 - Run `teamwork_server_nogui.exe` in the server bin folder.
2. Type `I agree` and press **Enter**.
3. Type the path to the license key file and the file name. For example, `C:\key\MagicDraw_17_0_5_TeamworkServer_key.txt` and press **Enter**.
 - If you have the commercial license that requires activation, before applying the license key, activate the license. For more information, see <http://www.nomagic.com/support/installation-and-use/teamwork-server-install.html#activating>
4. Press `y` and **Enter**, to apply the key.
5. Type `I agree` and press **Enter**.
6. Press **Enter**.
7. Do one of the following:
 - Press `y`, if you want to import projects that were stored in earlier versions of Teamwork Server and press **Enter**. Then, specify the location to the previous installation and press **Enter**. Press `y` if you want to import all projects.
 - Press `n`, if you do not want to import earlier projects and then press **Enter**.
8. Do one of the following:
 - Press `y` and then **Enter**, if you want to start the Teamwork Server with administrative permissions.
 - Press `n` and then **Enter** if you want to start the Teamwork Server without administrative permissions.

To add Teamwork Server to Windows services

1. Run `teamwork_server.exe` in the server bin folder. The **Teamwork Server** startup dialog appears.

2. Click the **Add Windows Service** button.
3. After the service is added, select one of the following:
 - Start this service from the Windows Services list.
 - Reboot the computer and the Service will start automatically.
 - To run the server, click the **Start Server** button.

 This feature is available only on Windows operating systems. Windows 7 OS and Windows Vista OS Firewall do not allow remote connections. Hence after adding Teamwork Server to Windows 7 or Windows Vista services, you have to add the Teamwork Server port number 1100 in Windows Firewall Exceptions list. Only then all remote connections to Teamwork Server will be allowed.

To start Teamwork Server as a service on Red Hat Linux

1. Create a new service script file named "teamwork".
2. Copy the following script code and paste it into the file.

Script code

```
#!/bin/bash
#
# chkconfig: - 91 60
# description: MagicDraw TeamWork Server

### BEGIN INIT INFO
# Provides: teamwork
# Required-Start: $local_fs $network $named $remote_fs $syslog
# Required-Stop: $local_fs $network $named $remote_fs $syslog
# Short-Description: MagicDraw TeamWork Server
# Description: This script is used to start MagicDraw TeamWork Server
### END INIT INFO

RETVAL=0 TEAMWORK_HOME="/var/MagicDraw_Teamwork_Server/bin"
prog="teamwork_server_nogui"
prog_stop="stop_teamwork_server"
desc="MagicDraw Teamwork Server"
args="SERVICE"

check() {
if [ -f /var/lock/$prog ]; then
    if ps -p $(cat /var/lock/$prog 2>/dev/null) >/dev/null; then
        return 0
    fi
fi
return 3
}

status() {
check
if [ $? -eq 0 ]; then
    echo "${desc} is running..."
    return 0
fi
echo "${desc} is stopped"
return 3
}

start() {
check
if [ $? -eq 0 ]; then
    echo "${desc} is already started..."
    return 2
fi

echo -n "Starting $desc ($prog): "
$TEAMWORK_HOME/$prog $args &
```

```


RETVAL=$?
SCRIPT_PID=$!

COUNT=0
while [ "$COUNT" -le 15 ] && [ -z $JAVA_PID ]
do
    JAVA_PID=$(pgrep -P $SCRIPT_PID java)
    let COUNT=COUNT+1
    sleep 1
done
[ $RETVAL -eq 0 ] && echo $JAVA_PID >/var/lock/$prog echo
}

stop() {
echo -n $"Shutting down $desc ($prog): "
$TEAMWORK_HOME/$prog_stop
RETVAL=$?
[ $RETVAL -eq 0 ] && rm -f /var/lock/$prog
return $RETVAL
}

case "$1" in
    start)
        start
        RETVAL=$?
        ;;
    stop)
        stop
        ;;
    restart)
        stop
        start
        RETVAL=$?
        ;;
    status)
        status teamwork
        RETVAL=$?
        ;;
    *)
        echo $"Usage: $0 {start|stop|restart|status}"
        exit 3
    esac
exit $RETVAL

```

 This script can also be used in non-RedHat based GNU/Linux distributions.

3. Change the value of the TEAMWORK_HOME variable according to the path of the Teamwork Server installation bin folder.
4. Save the file and move it into the system directory "/etc/init.d". 5. In the command line, type the following commands:

```
cd /etc/rc3.d  
ln -s ../init.d/teamwork S99teamwork
```

5. You can also configure the service for runlevel using the following command: `chkconfig --level 3 teamwork on`
6. In the command line, type the following command:

```
service teamwork start
```

To change the server license key from the command line

1. Stop the Teamwork Server (see [Stopping Teamwork Server](#)¹¹).
2. Start the Teamwork Server from the command line. Add the following argument:
"`-changeKey -key:<path to the key file location>`".

This is the sample for the Windows OS:

```
teamwork_server_nogui.exe -changeKey -key:C:  
\MagicDraw_16_0_TeamworkServer_key.xml
```

The path to the key file should be fully qualified and without spaces.

3.2 Stopping Teamwork Server



Server users

Make sure all users are logged out before stopping the server.

To stop Teamwork Server

- In the `<Teamwork Server home>\bin` directory, run the `stop_teamwork_server.exe`.

To stop Teamwork Server on Red Hat Linux

- In the command line, type the following command: `service teamwork stop`

¹¹ <https://docs.nomagic.com/display/NMDOC/Starting+and+stopping+Teamwork+Server>

To remove Teamwork Server from the Windows services

1. Run `teamwork_server.exe` in the server bin folder. The **Teamwork Server** startup dialog opens.
2. Click the **Remove Windows Service** button.



Operating System Availability

This feature is only available on Windows operating systems.

3.3 Upgrading Teamwork Server

You can automatically or manually upgrade Teamwork Server. In the automatic upgrade, a new Teamwork Server version will be installed in place of the current one. Upgrade the server manually to keep the current server version without any changes. The manual upgrade allows you to install a new server version on a new location. You will be able to import your old projects and users and test the functionality of a new server version. After that you can remove a previous server version. Because version 16.9 commercial licenses are locked to a particular machine, you will be requested to activate the license and receive the commercial license dedicated to that machine after the upgrade. If you activated the license before upgrading with a new version, the license will be activated automatically.



You should have valid Software Assurance to upgrade Teamwork Server. For more about Software Assurance, see at <https://www.nomagic.com/support/sales-and-licensing/software-assurance-maintenance-contracts>.

3.3.1 Automatic upgrade of Teamwork Server

The easiest way to renew MagicDraw Teamwork Server is to upgrade it automatically. Using this feature, you will automatically receive upgrades with the newest versions and service packs.



Make sure server and the client versions are the same. We also recommend using the same JVM version for the server and client, as well as making a backup of the project folder before upgrading Teamwork Server.

To upgrade Teamwork Server automatically

1. Stop Teamwork Server (see [Stopping Teamwork Server\(see page 19\)](#)) and close the **Administrator's Console**.
2. Deactivate the current license for the Teamwork Server.
3. For the Windows operating system, remove the Teamwork Server NT service, if it is added. See the procedure on how [to remove Teamwork Server from the Windows services\(see page 19\)](#).



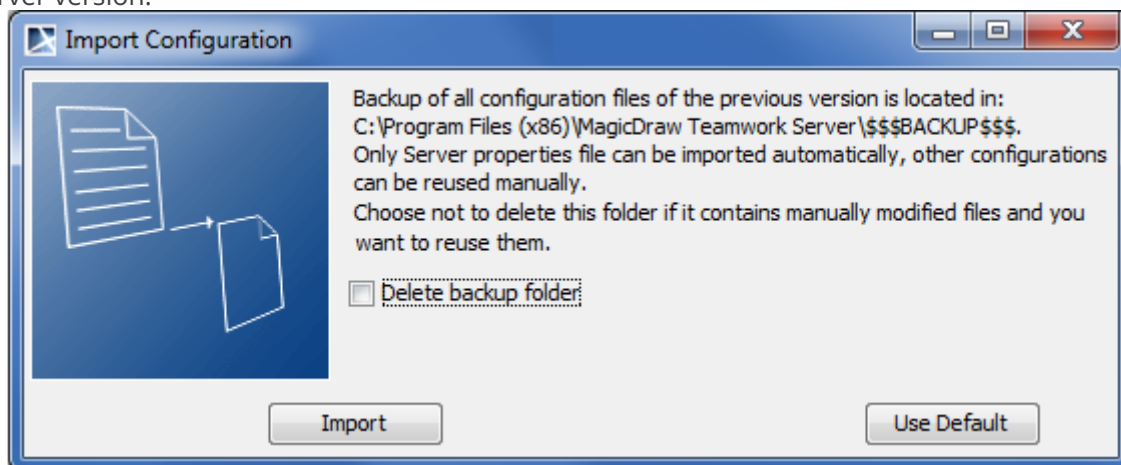
Skip step 3 for other operating systems.

4. Start the Teamwork Server GUI version (see how [to start the server using GUI\(see page 13\)](#)).

- When the dialog opens, click the Check for Updates button to check and download program upgrades and updates.

i If the **HTTP Proxy Server Connection** dialog opens, click **Use HTTP proxy server** if you want to use a proxy server. Enter the required values and click **OK** when you are done. Checking for updates starts.

- The **Update Information** dialog opens, displaying information about available updates. Click the **Update to New Version** button to start upgrading the server.
- When the automatic update finishes, the **Import Configuration** dialog opens. In this dialog, click the **Import** button to import all teamwork projects and users from the previous Teamwork Server version.



i We strongly recommend that you store the backup folder.

- After upgrading the server to version 16.9 or later, you will be asked to apply the commercial license dedicated for that particular machine. Please visit [Activating the Teamwork Server commercial license](#)¹² for instructions on getting the Teamwork Server license.

! Teamwork Server and its client must be of the same version. Otherwise, clients will not be able to connect to the server.

3.3.2 Manual upgrade of Teamwork Server

Installing a newer version does not detect or remove the current server version. You can import all projects and users to the new server.

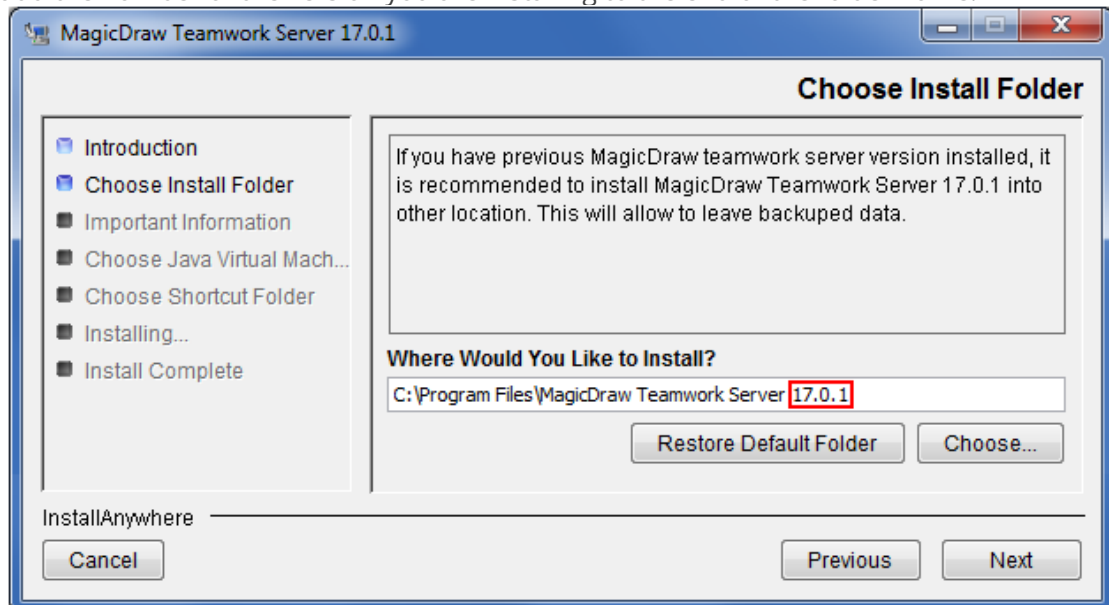
✓ For a description of importing procedures, please see [Importing projects and users from earlier versions of Teamwork Server](#)(see page 23).

The version of project server and client should be the same. We also recommend using the same JVM version for the server and client.

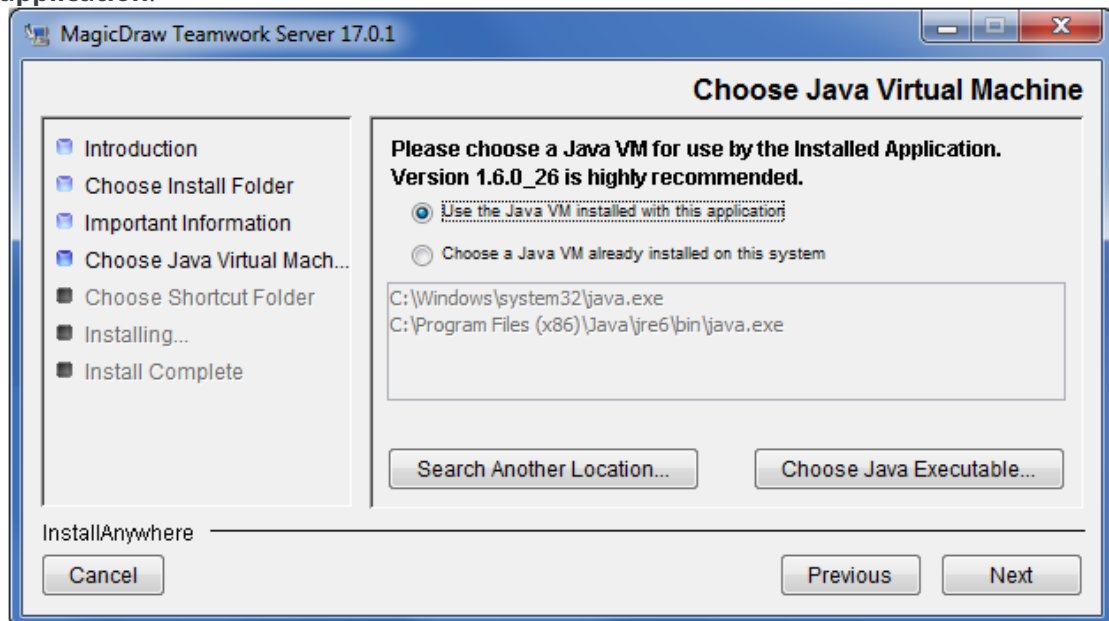
To upgrade Teamwork Server manually

¹² <https://docs.nomagic.com/display/NMDOC/Activating+the+Teamwork+Server+commercial+license>


1. Stop Teamwork Server, see [Stopping Teamwork Server](#)(see page 19).
2. Deactivate the current license for the Teamwork Server.
3. Remove Teamwork Server from Windows services, if it was added (see [remove Teamwork Server from the Windows services](#)(see page 20)).
4. Run the installation file. For successful installation please use the following recommendations:
 - a. For restoring backup data, install the new Teamwork Server version into a different location from the current Teamwork Server. For example, under **Choose Install Folder** add the number of the version you are installing to the end of the folder name.



- b. Under **Choose Java Virtual Machine**, click **Use the Java VM installed with this application**.



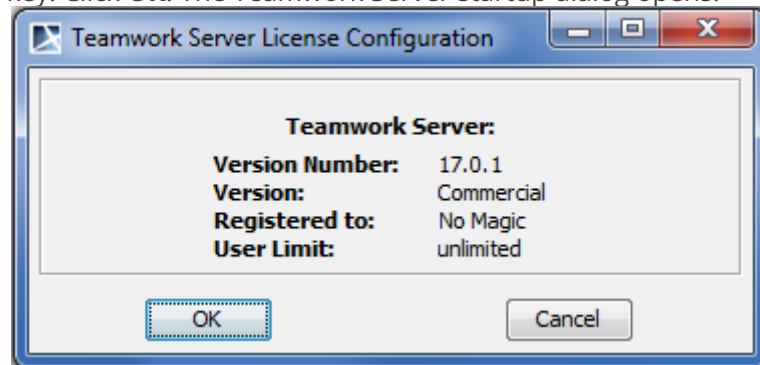
5. Start newly installed Teamwork Server. The Import Configuration dialog opens.

 The import time may take 90 minutes or more, depending on the quantity and size of the imported projects.

6. In the Teamwork Server License Manager dialog, enter the license key.

For more information about node locked license activation, see <https://www.nomagic.com/support/installation-and-use/teamwork-server-install#activating>.

- The Teamwork Server License Configuration dialog with license information opens after you have entered a license key. Click **OK**. The Teamwork Server startup dialog opens.



- Click the **Start Server** button. The server starts.
- If needed, add Teamwork Server to Windows services (see how [to add Teamwork Server to Windows services](#))(see page 13).
- Remove the old Teamwork Server version. Ensure the server and client versions are the same. We also recommend using the same JVM version for the server and client.

To upgrade Teamwork Server manually without GUI

⚠ We recommend making a backup of the project folder before upgrading the Teamwork Server.

- Stop Teamwork Server (see [stop teamwork server](#))(see page 19)).
- Deactivate the current license for the Teamwork Server. For more information, see https://www.nomagic.com/support/activation#deactivation_in_management.
- For the Windows operating system, remove the Teamwork Server NT service, if it was added (see how [to remove Teamwork Server from the Windows services](#))(see page 19). Skip this step for other operating systems.
- Extract the *MD_UML_<version number>_teamwork_server_no_installs.zip*.
- Start the new Teamwork Server without GUI (see how [to start Teamwork Server from the command line](#))(see page 13).

i You can only use **manual** Teamwork Server upgrade without GUI. This is not available with automatic updates.

3.4 Importing projects and users from earlier versions of Teamwork Server



Use one of the following to import projects and users from earlier versions:

- [Replacing the projects folder](#)(see page 24)
- [Modifying the muserver.projects_directory property in muserver.properties file](#)(see page 25)
- [Changing the repository location in Administrator's Console](#)(see page 25)

3.4.1 Replacing the projects folder

1. Stop Teamwork Server.
2. Copy and paste the projects folder to the newly installed projects folder of Teamwork Server.

See the location of the projects folder in the following table.

Teamwork Server Version	Location of the projects folder
17.0.3 or earlier	<i><Teamwork Server installation directory>\projects</i> on all operating systems
17.0.4 and 17.0.5	<i>C:\ProgramData\.magicdrawserver\<version number>\projects</i> on Windows 7/8 <i>C:\Documents and Settings\All Users\ApplicationData\.magicdrawserver\<version number>\projects</i> on Windows 2000/XP <i>C:\WINNT\Profiles\All Users\ApplicationData\.magicdrawserver\<version number>\projects</i> on Windows NT4 <i><install.root>/projects</i> on other operating systems <div> The projects folder is created automatically after starting Teamwork Server for the first time.</div>
18.0 or later	<i>C:\ProgramData\.magicdrawserver\projects</i> on Windows 7/8 <i>C:\Documents and Settings\All Users\ApplicationData\.magicdrawserver\projects</i> on Windows 2000/XP <i>C:\WINNT\Profiles\All Users\ApplicationData\.magicdrawserver\projects</i> on Windows NT4 <i><install.root>/projects</i> on other operating systems <div> The projects folder is created automatically after starting Teamwork Server for the first time.</div>


3.4.2 Modifying the `muserver.projects_directory` property in `muserver.properties` file

1. Stop Teamwork Server.
2. Go to the `.magicdrawserver\data` folder and open the `muserver.properties` file. The data folder is located in the same folder as the projects folder. For the projects folder location, refer the preceding table.
3. In the `muserver.projects_directory`, specify the path to the projects folder. For example:
`muserver.projects_directory=C:\ProgramData\magicdrawserver\17.0.5\projects`

3.4.3 Changing the repository location in Administrator's Console

You can indicate the `projects` folder that contains the project you want to import by changing the repository location.

To change the repository location and import projects

1. In the Teamwork Administrator's Console window, on the **Repository** tab next to the **Location** box, click the  button.
2. In the **Open** dialog, select the `projects` folder and click **Open**.
3. When you receive the warning that you must run a repository test to apply changes, click **Run Test**.
4. In the **Repository Test Passed** dialog, click **OK**.
5. When you receive the message that changes have been saved and the server should be restarted to apply the changed properties, click **OK**.
6. In the Teamwork Administrator's Console window, from **Menu**, select the **Restart Server** command.



Server users

Make sure all users are logged out before restarting the server.

7. In the Information dialog showing that the server is restarting and you should try to login again after few minutes, click **OK**.

3.5 Migrating from the SVN repository to the Native repository type

Migrating from the SVN repository to the Native repository type

1. Start the server on the SVN repository.
2. Open the **Administrator's Console**, trigger project export. Select a directory to permanently store the projects. This will be the directory in which the server will subsequently operate.

3. In the **Administrator's Console**, reconfigure the server for the Native repository. Specify the directory where you stored the projects as the directory to work with.
4. Restart the server to use this new repository.

**Server users**

Make sure all users are logged out before restarting the server.

5. Projects are now in the new Native repository.


3.6 Moving Teamwork Server

You can move your Teamwork Server from one computer to another. During this transfer, server configurations, projects, and user information will be moved. The instructions below show how to move Teamwork Server from computer A to computer B.

To move Teamwork Server

1. Stop Teamwork Server in computer A (see how to [stop teamwork server](#))(see page 19). Be sure to [back up all data](#) (see page 26) in case of unsuccessful data transfer.
2. Install Teamwork Server in computer B.
3. In computer A, copy the *.magicdrawserver* folder and paste it in the same location in computer B. If your projects were stored somewhere other than in the Project folder, copy the folder where Teamwork Server projects were stored and the path to that repository. In computer B, restore the path to the same repository and paste the folder with projects to that location. For more information about how to change the path to a repository, see [how to change a repository path](#)(see page 26).
4. Start Teamwork Server (see [Managing Teamwork Server](#))(see page 27). You are required to enter a license key when starting Teamwork Server for the first time. For more information about license activation, see <https://www.nomagic.com/support/installation-and-use/teamwork-server-install#activating>.
5. You can remove back up data in computer A (optional). Before removing back up data from computer A, make sure all data and server configurations are restored in computer B after moving.

To change a repository path

1. In the Teamwork Administrator's Console, click the **Repository** tab.
2. Near the **Repository Location** box, click the  button to select the repository path.
3. Click **Apply Changes**. For more information about repositories, see [Teamwork System Design](#)(see page 9).

3.7 Backups of server projects

To back up server projects, simply copy them from one location to another. Make sure you stop the server before copying files.

All server projects are stored in a projects directory in a server. You may periodically copy this directory from a running server to a backup server. However, there may be some issues with this approach:

- You can copy files that are in an inconsistent state (for example, one file contains new data while another has earlier data). One solution is to have copies in different times, reducing the risk of losing a consistent project state.
- A running server may have corrupted data, which may be copied to a backup server. In this case, you could have more than one backup so that the latest working backup may be used in case of data loss.
- The server hardware or software fails, so that you cannot start Teamwork Server at all. If this occurs, you can [install a new Teamwork Server](#)¹³ and simply move all data from a project's directory to other hardware and start the server there.

Remember that you can always restore your project from a previous stable version of your data.

Related pages

- [Importing projects and users from earlier versions of Teamwork Server](#)(see page 23)
- [Stopping Teamwork Server](#)(see page 19)
- [Starting Teamwork Server](#)(see page 13)

4 Managing Teamwork Server

As a Teamwork server administrator, you can:

- Set general Teamwork Server properties.
- Change administrator's and users' passwords.
- Manage users, user permissions, and how to assign projects to users.

Related Pages:

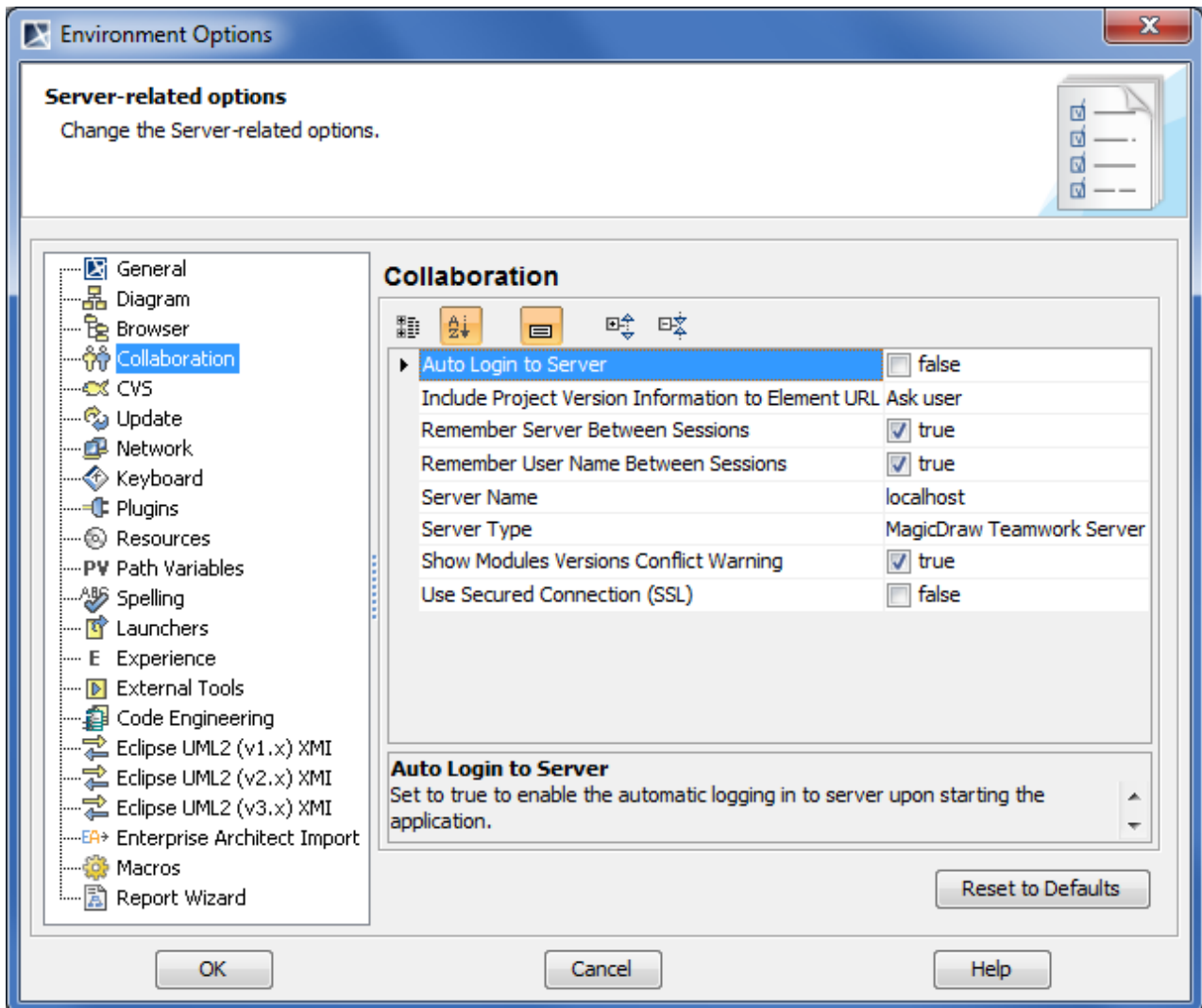
- [Managing Teamwork Server](#)(see page 27)
 - [Customizing Teamwork Server properties](#)(see page 28)
 - [Changing Administrator Password](#)(see page 29)
 - [User Management](#)(see page 30)
 - [User Permissions](#)(see page 34)
 - [Starting the Administrator's Console](#)(see page 36)
 - [Administrator's Console Dialog](#)(see page 36)
 - [Active Users tab](#)(see page 36)
 - [Projects tab](#)(see page 37)
 - [Log File tab](#)(see page 42)
 - [The Properties tab](#)(see page 43)
 - [Secured Connection tab](#)(see page 43)
 - [Repository tab](#)(see page 46)
 - [LDAP Integration tab](#)(see page 48)
 - [Administrating Server via Command Line Utility](#)(see page 49)
 -

¹³ <https://docs.nomagic.com/display/TWS190SP4/Teamwork+Server+installation>

- Data Migration between Different Repositories(see page 51)
- Changing the Teamwork Server Debugging Mode(see page 51)
- Configuring server properties(see page 52)

4.1 Customizing Teamwork Server properties

Set the general Teamwork Server properties in the Collaboration pane of the Environment Options dialog.



4.1.1 Environment Option dialog. Collaboration options group

An option description outlines the effect of changing the option value.

To read the option description

1. Click an option whose value you want to change.
2. Read the option description in the area below the options list. You are now ready to change the option value.



Show Description Mode

Make sure that the Show Description mode is turned on in the Environment Options dialog. To turn the Show Description mode on or off, click the **Show Description** button on the property group toolbar.



For more information about the Environment Options dialog, see [Customizing Environment Options](#)¹⁴.

Related Pages:

- [Managing Teamwork Server](#)(see page 27)
 - [Customizing Teamwork Server properties](#)(see page 28)
 - [Changing Administrator Password](#)(see page 29)
 - [User Management](#)(see page 30)
 - [Starting the Administrator's Console](#)(see page 36)
 - [Administrator's Console Dialog](#)(see page 36)
 - [Administrating Server via Command Line Utility](#)(see page 49)
 - [Data Migration between Different Repositories](#)(see page 51)
 - [Changing the Teamwork Server Debugging Mode](#)(see page 51)
 - [Configuring server properties](#)(see page 52)

4.2 Changing Administrator Password

The following instructions describe how to reset the Administrator's password.

To reset the Administrator's password

1. Stop Teamwork Server.
2. Open the folder storing the server projects. This location depends on your Teamwork Server version. For more information, read how to [replace the project folder](#)(see page 23).
3. In the open folder, open the *users.xml* file for editing.
4. Delete all letters and numbers between the `<password>``</password>` tags:

```
<users savedwithversion="17.0.2 beta">
  <user>
    <login>Administrator</login>
    <password>Leda23758be9e36e5e0d2a6a87de584aaca0193f</password>
```

Delete

5. Save any changes in the *users.xml* file.
6. Restart the Teamwork Server.

¹⁴ <https://docs.nomagic.com/display/MD190SP4/Customizing+environment+options>




Server users

Make sure all users are logged out before restarting the server.

7. Login to the Teamwork Server using the default Administrator's account:

- **Login name:** Administrator
- **Password:** Administrator

 To prevent unauthorized access, we advise changing the default Administrator's password.

For more information on changing user passwords, see [Working with Server Projects](#)¹⁵.

Related Pages:

- [Managing Teamwork Server](#)(see page 27)
 - [Customizing Teamwork Server properties](#)(see page 28)
 - [Changing Administrator Password](#)(see page 29)
 - [User Management](#)(see page 30)
 - [Starting the Administrator's Console](#)(see page 36)
 - [Administrator's Console Dialog](#)(see page 36)
 - [Administrating Server via Command Line Utility](#)(see page 49)
 - [Data Migration between Different Repositories](#)(see page 51)
 - [Changing the Teamwork Server Debugging Mode](#)(see page 51)
 - [Configuring server properties](#)(see page 52)

4.3 User Management

There are two types of users in Teamwork Server: users and administrators. Administrators in Teamwork Server have the ability to:

- Manage users.
- Create projects and assign users to them.
- Manage project versions and branches.
- Set user permissions for the system and projects (the read and edit modes are set by default).
- Remove users and projects from Teamwork.

Teamwork Server users have their own user accounts (including login names and passwords assigned by the administrator) and various types of permissions. Depending on where the user accounts are stored, users can be either:

- **Native** - the user's account data is stored locally.
- **External** - the user's account data is stored in the external database (Subversion and/or LDAP). Only the login name of an external user is stored locally.

You can create, edit, or remove both types of Teamwork users regardless of whether the integration with any external database (Subversion, LDAP) is enabled or disabled. The names of native and external users are unique per single server.

¹⁵ <https://docs.nomagic.com/display/MD185/Working+with+projects+from+Teamwork+Server>

Administrators can change a specific user's type by editing the user's account information, or convert a whole list of active Teamwork users by using the Teamwork Administrator's console. They can also convert an external user to a native one and vice versa

You will be connected to Teamwork Server once you complete the authorization process, which will prompt for your user ID (login name and password). Upon verification, you can work with the system.

i If there are two MagicDraw clients with the same login name, only one client is allowed to log into Teamwork Server at a time.

You can manage users in the Teamwork Administrator's Console or in MagicDraw UML after establishing a connection to Teamwork Server.

To add a new native user

1. From the **Collaborate** menu, select **Users**. The Edit Users dialog opens.
2. Click the **Add** button. The Add User dialog opens.
3. Enter the user's login name (full name for better identification) and password.
4. Click **OK**.
5. Select the types of system permissions for the user in the Permissions list, read [User Permissions](#)(see page 34).

To add a new external user

1. From the **Collaborate** menu, select **Users**. The Edit Users dialog opens.
2. Click the **Add** button. The Add User dialog opens.
3. Enter the user's login name and full name for better identification.
4. Select the External User check box.
5. Click **OK**.
6. Select the types of system permissions for the user in the Permissions list.

i As you cannot set a password for an external user in MagicDraw's Teamwork Server, use an appropriate tool to manage the external database (Subversion or LDAP) where the user's account is stored.

To convert a native user to an external one by editing the user's account information

1. From the **Collaborate** menu, select **Users**. The Edit Users dialog opens.
2. Click the **Edit** button. The Edit User dialog opens.
3. Enter the user's full name.
4. Select the External User check box.
5. Click **OK**.



- The password of a native user who has been converted to an external user will be retained. However, it will not be used in the user authentication.
- The user's native password will be enabled again only if the user is converted back to a native user.

To convert all native users to external

1. Start **Teamwork Administrator's Console**, see [Starting the Administrator's Console](#)(see page 36).
2. In the **Active Users** tab, click the **Convert Native Users to External** button, see [Active Users tab](#)(see page 36).
3. Click **Yes** to confirm your decision.



All converted users will be able to log into Teamwork Server only if they are available in the external user sources (LDAP or Subversion server to which your server is integrated).

You will be informed once the conversion has been completed. A Teamwork Server's user conversion can be:

- **Successful** - when all the users are converted from native to external. In this case the informational message is displayed, and you can check the list of all converted users in the server log.
- **Unsuccessful** - when the conversion failed. In this case an error message is displayed, and you can see the server log for more details.
- **Non-applicable** - when there are no users to convert from native to external. In this case an informational message is displayed.

For the information about the server log file, see [Log File tab](#)(see page 42).

To convert an external user to native by editing the user's account information

1. From the **Collaborate** menu, select **Users**. The Edit Users dialog opens.
2. Click the **Edit** button. The Edit User dialog opens.
3. Enter the user's full name.
4. Clear the **External User** check box.
5. Type and retype the password.
6. Click **OK**.



If the converted user used to be a native user, the password will be the same one used when he or she was a native.

To convert all external users to native

1. Start **Teamwork Administrator's Console**, read [Starting the Administrator's Console](#)(see page 36).
2. On the **Active Users** tab, click the **Convert External Users to Native** button, see [Active Users tab](#)(see page 36).
3. Click **Yes** to confirm your decision.



- All converted users will not be able to log into Teamwork Server as they do not have passwords; therefore, the administrator has to set up a password for each user after the conversion.
- If the converted user used to be a native user, the password will be reset to the same one used when he or she was a native.

You will be informed once the conversion has been completed. A Teamwork Server's user conversion can be:

- **Successful** - when all the users are converted from external to native. In this case an informational message is displayed, and you can check the list of all converted users in the server log.
- **Unsuccessful** - when the conversion failed. In this case an error message is displayed, and you can see the server log for more details.
- **Non-applicable** - when there are no users to convert from external to native. In this case an informational message is displayed.

For more information about the server log file, see [Log File tab](#)(see page 42).

To remove a user from Teamwork

1. From the **Collaborate** menu, select **Users**. The Edit Users dialog opens.
2. In the **Users** area, select the user and click **Remove**.

To assign a project to a user

1. From the **Collaborate** menu, select **Users**. The Edit Users dialog opens.
2. If you do not see the Teamwork projects list, click **More**. The list of available Teamwork project is displayed in the **Available Projects** area.
3. Select a project you want to assign to the selected user.
4. Click the << button to move the selected project to the **Assigned Projects** list.
5. Click **OK** when you are done.



- Once a user has been added to a project, the default user rights will be created allowing the user to access the project only according to the rights given.
- The system permissions have a higher priority over the project permissions. For example, a user whose system permissions allow model editing can edit all projects, even if the user does not have rights to edit the projects.

Related Pages:


- [User Permissions](#)(see page 34)

4.3.1 User Permissions

You can assign several types of permissions to Teamwork users to coordinate the work of the whole team. You can specify the types of user permissions in the Edit Users dialog.

There are two categories of permissions:

- System access – administrative permissions to access and manage users and projects.
- Project access – permissions to work on specific projects.

Permissions	Users can
Edit Model	Modify a Teamwork project.
Read Model	Open the content of any Teamwork project. <div> If the List not assigned projects permission is not selected, you will be able to open the project using its URL.</div>
Read including used projects (Legacy)	Access the used project data from the main project.
Assign user to project	Assign any user to any Teamwork project.
Edit project properties	Edit Teamwork project names and tags.
Administrator project	Manage (create, rename, and remove) project branches as well as migrate projects to later versions.
List not assigned projects	See all (assigned and not assigned) teamwork projects. If not selected, only projects that are assigned to the user will be listed.
Create project/ category	Create a new Teamwork project or category.
Rename category	Edit a category name.

Permissions	Users can
Create user	Create a new user.
Remove project/ category	Remove a project or category from Teamwork Server.
Edit user properties	Edit user names and passwords.
Remove user	Delete user accounts from Teamwork Server. This permission will unlock all model elements locked by a deleted user in all projects.
Access user list	Allows user to see other Teamwork Server users.

To view the users' permissions


1. From the **Collaborate** menu, select **Users**. The Edit Users dialog opens.
2. A list of users and their permissions is presented in the **Permissions** area.

To edit user permissions

1. From the **Collaborate** menu, select **Users**. The Edit Users dialog opens.
2. In the **Users** area, select a user that permissions you want to edit.
3. Select the check box to give or clear it to remove the selected permission in the **Permissions** area.

To assign a user to a project

1. From the **Collaborate** menu, select **Projects**. The Edit Projects dialog opens.
2. If you do not see the unassigned users list, click **More**. The list of available users is displayed in the **Available Users** area.
3. Select the user you want to assign to the selected project.
4. Click the << button to move the selected user to the **Assigned Users** list.
5. Click **OK** when you are done.


 When a user is added to a project, default user rights are created, allowing the user to access the project according to the assigned rights.

4.4 Starting the Administrator's Console

The MagicDraw UML Teamwork Server Administrator's Console is used for Teamwork Server status observation and administrative control. The server holds information about active users connecting to the server and loaded projects. The administrator can shut down or restart the server, change its properties, and view log files (including debug information) for both the server and separate projects.

To start the Administrator's Console

1. Open the <MagicDraw Teamwork Server installation directory>\bin folder.
2. Use the *teamwork_administrator.exe* file.

 You can also start the Administrator's Console from the client side. The *teamwork_administrator.exe* file is located in <MagicDraw installation directory>\collaboration.

4.5 Administrator's Console Dialog

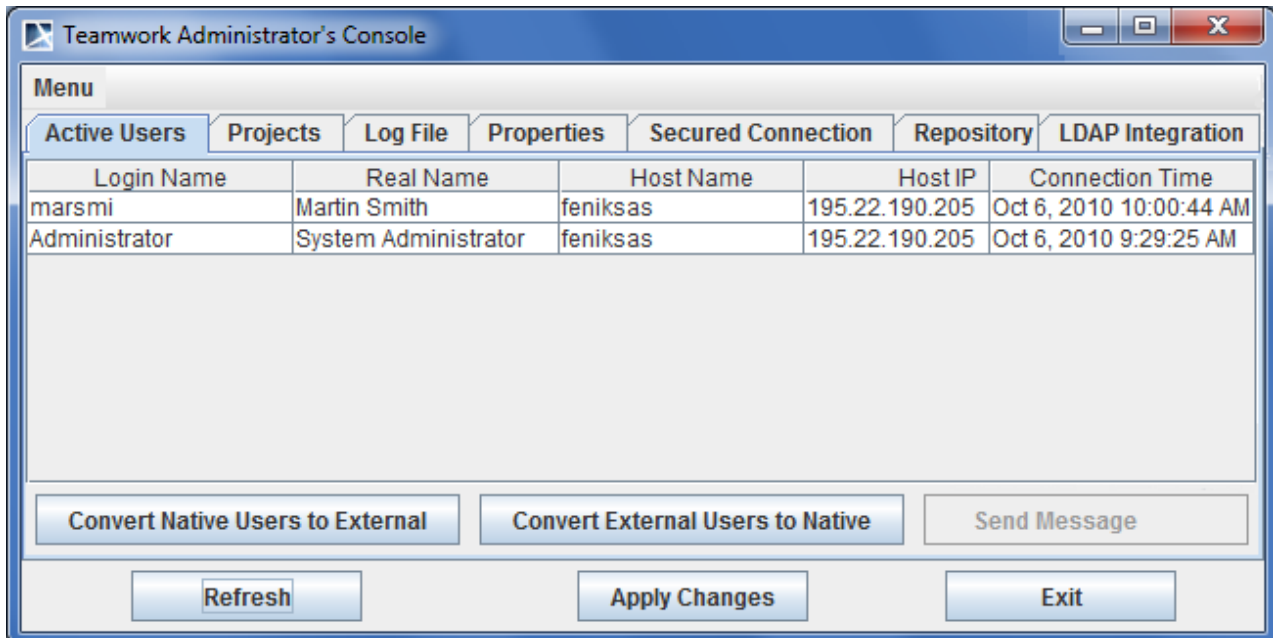
The Administrator's Console dialog is constructed of six tabs: Active Users, Projects, Log File, Properties, Repository, and LDAP Integration. All these tabs are described in the following sections.

Related Pages:

- [Active Users tab](#)(see page 36)
- [Projects tab](#)(see page 37)
- [Log File tab](#)(see page 42)
- [The Properties tab](#)(see page 43)
- [Secured Connection tab](#)(see page 43)
- [Repository tab](#)(see page 46)
- [LDAP Integration tab](#)(see page 48)

4.5.1 Active Users tab

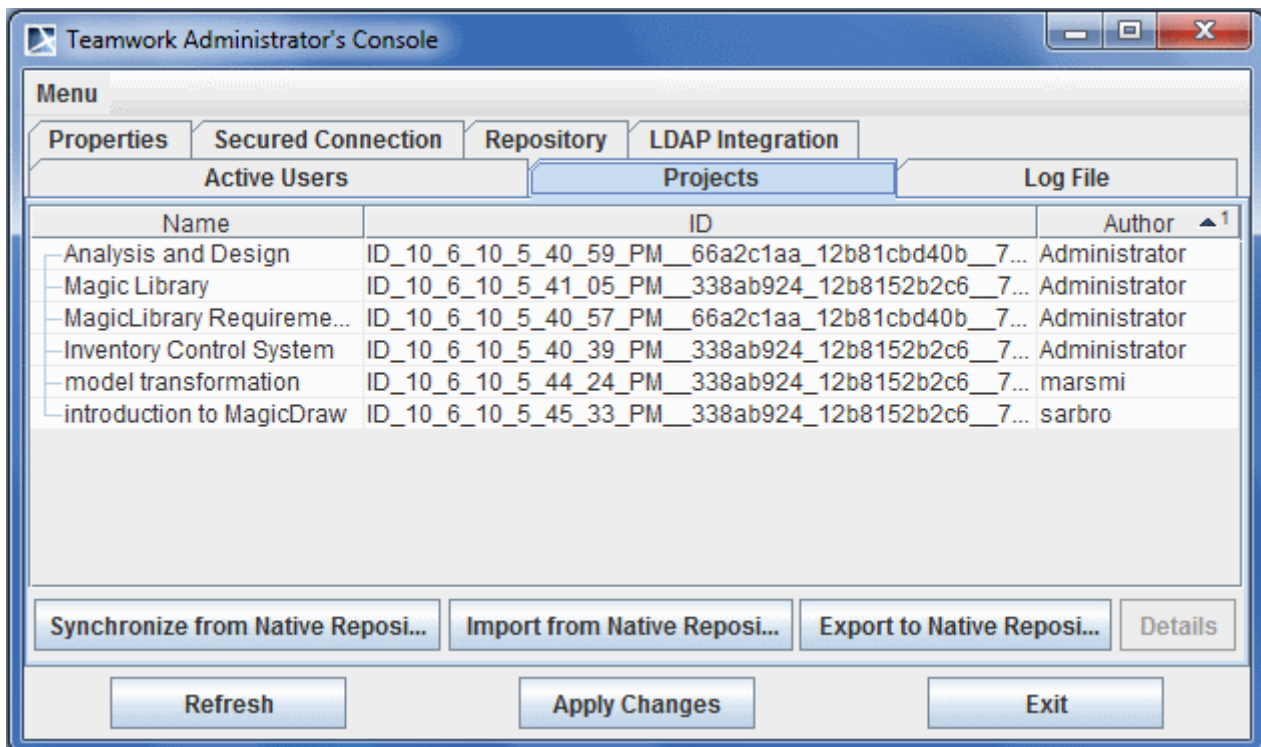
In the **Active Users** tab, the administrator is able to observe all the users who are currently connected to the Teamwork Server.



Teamwork Administrator's Console. Active Users tab

4.5.2 Projects tab

The **Projects** tab displays the names, authors, and status of all projects stored on the server.




Teamwork Administrator's Console. Projects tab

You can trigger import from and export to the native repository, as well as synchronization, in the **Projects** tab.

To synchronize projects on the current server with data from a selected location


1. Click the **Synchronize from Native Repository** button. The Select Native Repository dialog opens.
2. Select the directory where the data to synchronize is stored.

 This must be a directory that stores project versions in the Teamwork Server's Native repository format, such as a directory where you have exported the projects or a directory on which the server operated in the past.

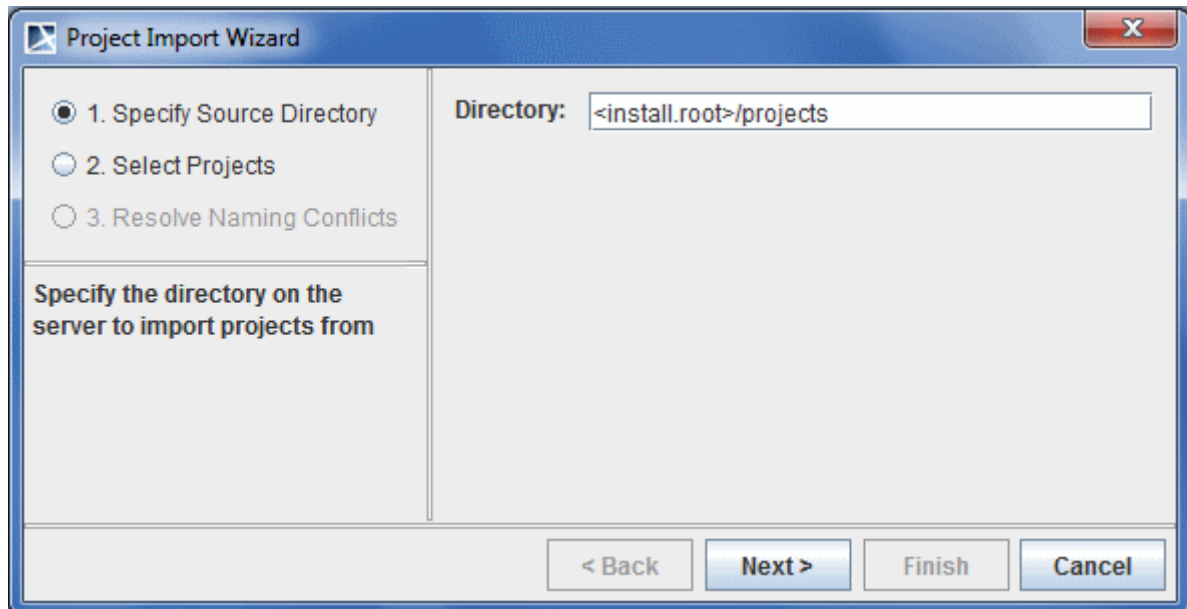
3. Click **Synchronize**. The contents of the selected directory are imported into the current server. After synchronization, the server assumes that the synchronized projects are foreign. These projects cannot be modified, but you can repeat the synchronization to update them with new data.

To import projects into the current server

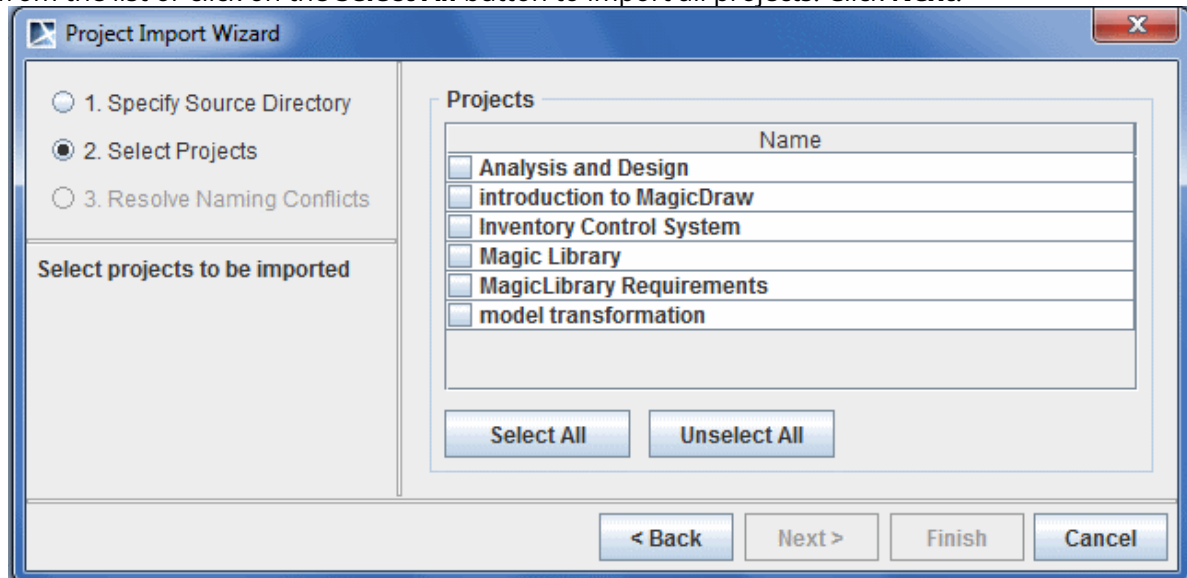
1. Click the **Import from Native Repository** button. The Projects Import Wizard opens.
2. Select the directory where the projects to import are stored.

 This must be a directory that stores project versions in the Teamwork Server's Native repository format, such as a directory where you have exported the projects or a directory on which the server operated in the past.

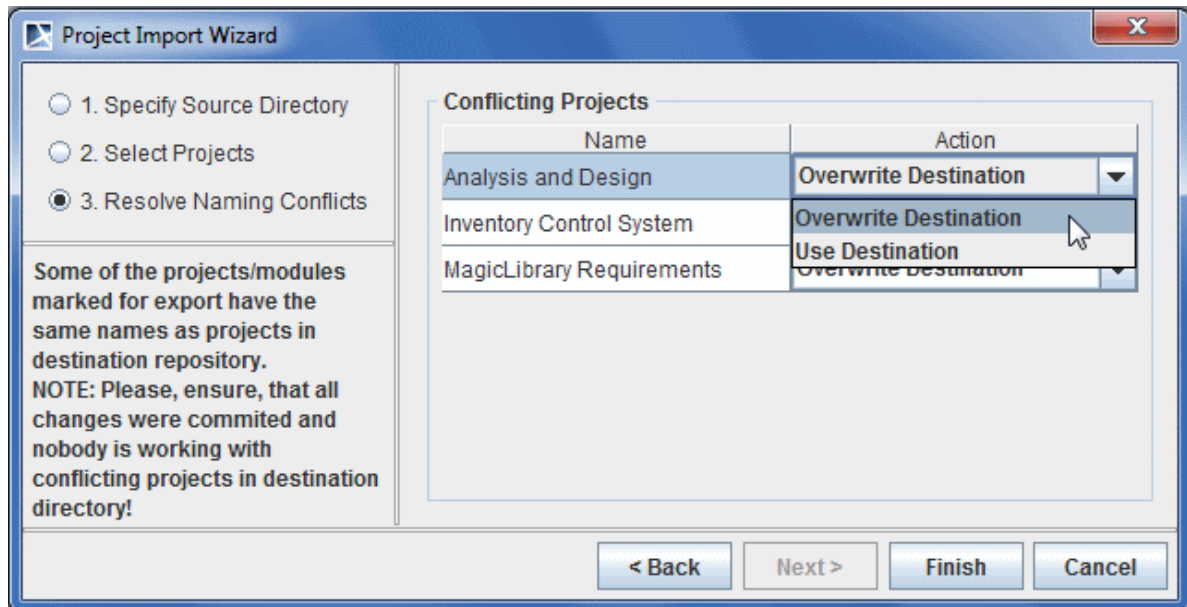
3. Click **Next**. In the **Directory** box, type the path to the backup folder to restore files from the backup made while upgrading the server.



4. The **Projects** list displays all the projects stored in the directory. Select specific projects to import from the list or click on the **Select All** button to import all projects. Click **Next**.



5. The **Resolve Naming Conflicts** tab appears. The import process will check if the selected projects/used projects/profiles already exist in the selected destination directory server (comparison is done by name). There are two ways to resolve naming conflicts:
- **Use destination.** The used project will not be imported. All other projects/used projects being imported will be modified to use the existing used project/profile in the current server.
 - **Overwrite destination.** Import selected profiles or used projects and commit them by overwriting the existing ones in the current server.

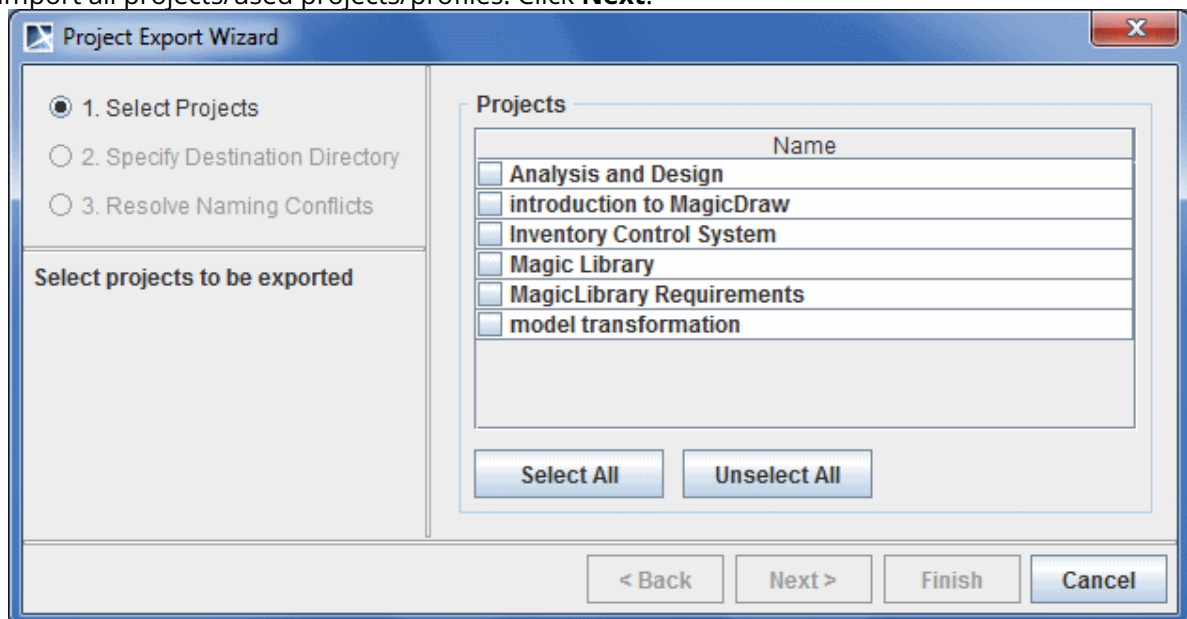


6. Click Finish.

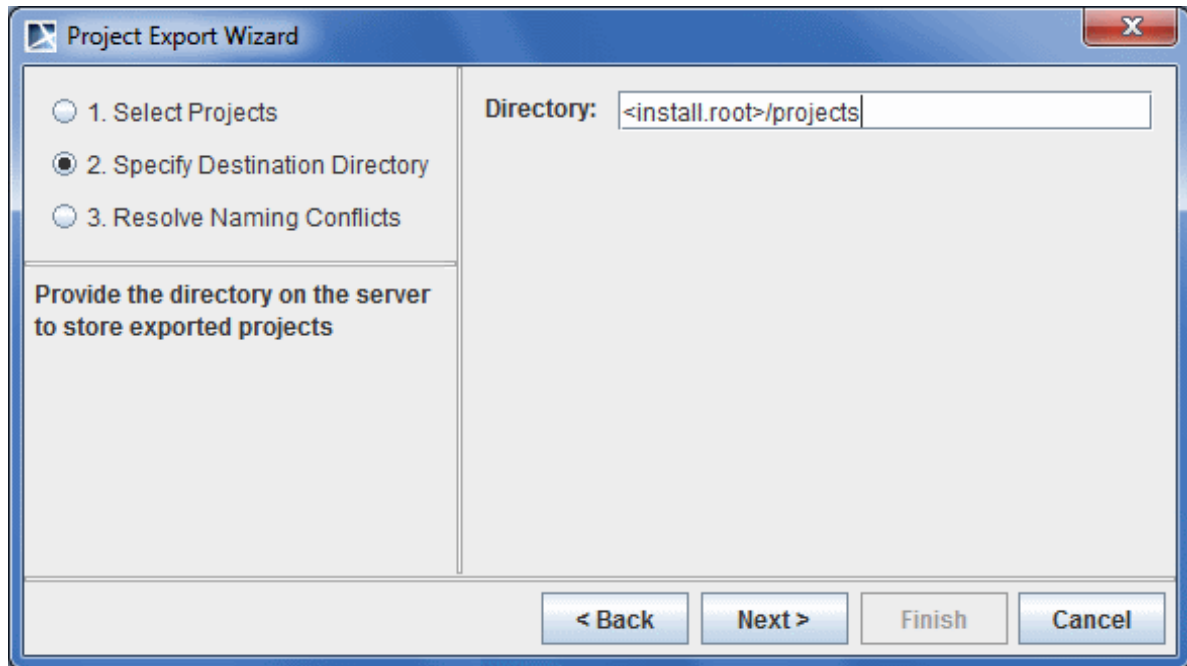
The contents of the selected directory are imported into the current server. After the import, the current server assumes the ownership of the imported projects. They can be modified, but the import cannot be repeated.

To export projects from the current server

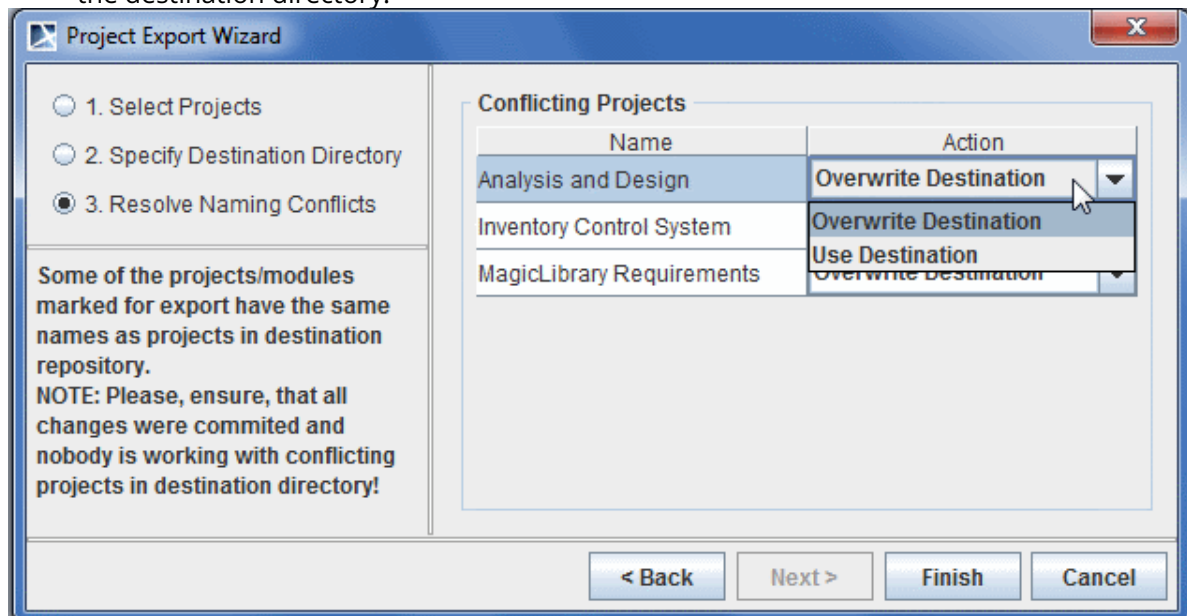
1. Click the **Export to Native Repository** button. The Project Export Wizard opens.
2. Select projects/used projects/profiles to export from the list, or click on the **Select All** button to import all projects/used projects/profiles. Click **Next**.



3. Select the directory to export the selected project. This can be a new directory or a directory where the Native repository is stored.



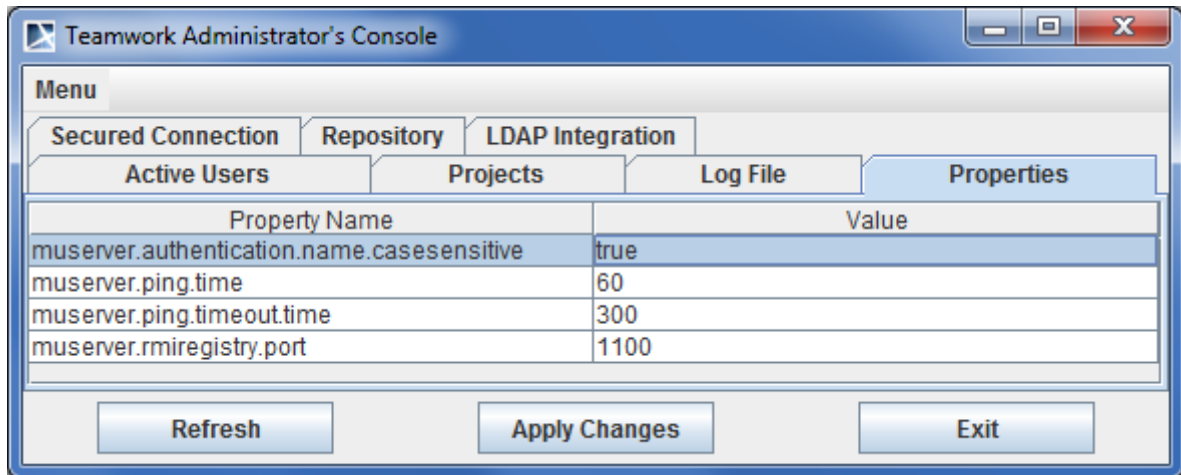
4. The resolve naming conflicts in the destination directory opens. If you select a new directory to export the project, there will be no naming conflicts. However, if the projects/used projects/profiles to be export already exist in the selected destination directory (comparison is done by name), naming conflicts will occur.
5. There are two methods to resolve naming conflicts:
 - **Overwrite Destination.** Overwrite the projects/used projects/profiles currently existing in the destination directory with the exported projects/used projects/profiles.
 - **Use Destination.** Do not replace the projects/used projects/profiles currently existing in the destination directory.



To view the details of a project

1. Select a project and from the Teamwork Administrative's Console dialog and click **Details**.
2. The Project dialog opens.

4.5.4 The Properties tab



Teamwork Administrator's Console. Properties tab

The **Properties** tab displays the following Teamwork Server properties:

- **`muserver.authentication.name.casesensitive`** specifies if the authentication is case sensitive or not. If the property is set to true, the entered login name must exactly match the registered one, and is case sensitive.
- **`muserver.ping.time`** records how often the Teamwork Server pings a client. After sending a ping, the server waits for the client to answer. If the client does not respond during the interval of time specified in the `Muserver.ping.timeout.time` property, the user is logged out from the Server.
- **`muserver.ping.timeout.time`** displays the number of seconds the Teamwork Server will wait for a client's response once the server sends its ping. If no answer is received during this interval, the user is logged out from the Server
- **`muserver.rmiregistry.port`** specifies the port where communication between MagicDraw clients and Teamwork Server occurs (RMI traffic over TCP/IP).

Restart the server to apply changes.



Logging out

Make sure all users are logged out before stopping the server.

4.5.5 Secured Connection tab

Our modeling tool allows you to safely transfer data using a secured connection (SSL).



If the SSL connection is established in the server side, you should also use the SSL connection in the client side when connecting to the server.

To use the SSL connection, you need two types of certificates, one for the server and one for the client. Certificates must be in Java Key Store format.

The server certificate is automatically placed in the <Teamwork Server installation directory>\cert folder after the SSL configuration is done.

Locate the client certificate manually. You should create a folder named *certs* and place into it these two files:

1. A client certificate named *cert.jks*.
2. A file named *cert.pass* wherein the certificate password is typed.

In this case, a modeling tool and Teamwork Administrator's Console applications are Teamwork server clients. Both should have the client certificate. Thus, the folder *certs* should be placed in two [Configuration file](#)¹⁶s locations:

- <Teamwork Administrator's Console installation directory> for the Teamwork Administrator's Console application
- <LOCAL_APPDATA1>\.magicdraw\<version number> on Windows OS or <user.home>\.magicdraw\<version number> on other OS for the MagicDraw application (It can be located in the folder <MagicDraw installation directory> either but the user home folder is the default one)


If the Teamwork Administrator's Console is not installed in a separate location, its installation directory is the same as Teamwork Server installation directory or the MagicDraw installation directory (if installed together).

You can get certificates from your system administrator or [generate certificates](#)(see page 44) by yourself. To work with multiple servers without need to launch different modeling tools, create more than one public key and import it to the client certificate, named *cert.jks*.

To generate certificates

We recommend using the KeyTool IUI application for generating certificates. This is a free tool that can be downloaded from the Internet.

1. Run the KeyTool IUI application.
2. Create empty files for storing certificates:
 - a. Select **Create > Keystore**.
 - b. Create an empty keystore file for the server. Do the following:
 - i. In the Keystore file dialog, set the location of the file and type a file name.
 - ii. In the Keystore password dialog, type the password for the server keystore file and click **OK**.
 - c. Create an empty keystore file for the client. Do the following:

 Do not repeat this step, if you already have the keystore file for the client. Multiple public keys should be imported to the already existing client file *cert.jks*. Skip this step and go to the step [#3](#)(see page 44)

- i. In the Keystore file dialog, set the location of the file and type a file name. In the next steps, create a new folder *certs* and save the file named *cert.jks* in it for easier certificate transfer.
 - ii. In the Keystore password dialog, type the password for the client keystore file and click **OK**.
3. Create a RSA keypair for the server:

¹⁶ <https://docs.nomagic.com/display/MD190SP4/Configuration+files>

- a. Select **Create > Keystore's entry > Private key, with vers. #3 > RSA**.
- b. In the Keystore file dialog, the Source section, open the created server keystore file and type a password.
- c. Provide the required information in the Target section and click OK.
- d. The dialog for creating a new alias will open. Type a new private key entry's alias name and a password for it. Click OK.
- e. You will see the created alias. Close the dialog.
4. Exclude a public key from the keypair to provide it to the client:
 - a. Select **Export > Private's key first certificate in chain > As simple certificate file**.
 - b. In the Keystore file dialog, the **Source** section, open the server keystore file and type its password.
 - c. Create a file where the key will be exported. In the Certificate file dialog, the **Target** section, set a location and type a file name for the client certificate. Click **OK**.
 - d. The dialog for selecting an alias will open. From the list, select the alias created in step 3.d and type its password. You will be able to see the created certificate.
5. Import a public key into the client certificate:
 - a. Select **Import > Keystore's entry > Trusted certificate > Regular certificate**.
 - b. In the **Source** section set the certificate file created in step 4 as a regular certificate file.
 - c. In the **Target** section:
 - i. for one client, set the client keystore file *cert.jks* as a keystore file and click **OK**.
 - ii. for multiple clients, import multiple certificates to the already existing *cert.jks* file.
 - d. The dialog will open asking you to enter a new alias name. Enter the alias name created in step 3 and click **OK**.
 - e. Some pop-up windows about the generation process will open. Close all of them after reviewing.

Generated certificates are now ready to use. Paste them into the correct location. Enable the secured connection (SSL) for server server side.

To enable the secured connection (SSL)

-
1. Run the Teamwork Server Administrator's Console.
 2. In the **Secured Connection** tab, select the **Enable Secure Connection** check box.
 3. Click **Browse** to add the server certificate.
 4. Enter the JKS password.
 5. Click the **Apply Changes** button.
 6. Restart Teamwork Server.



Server users

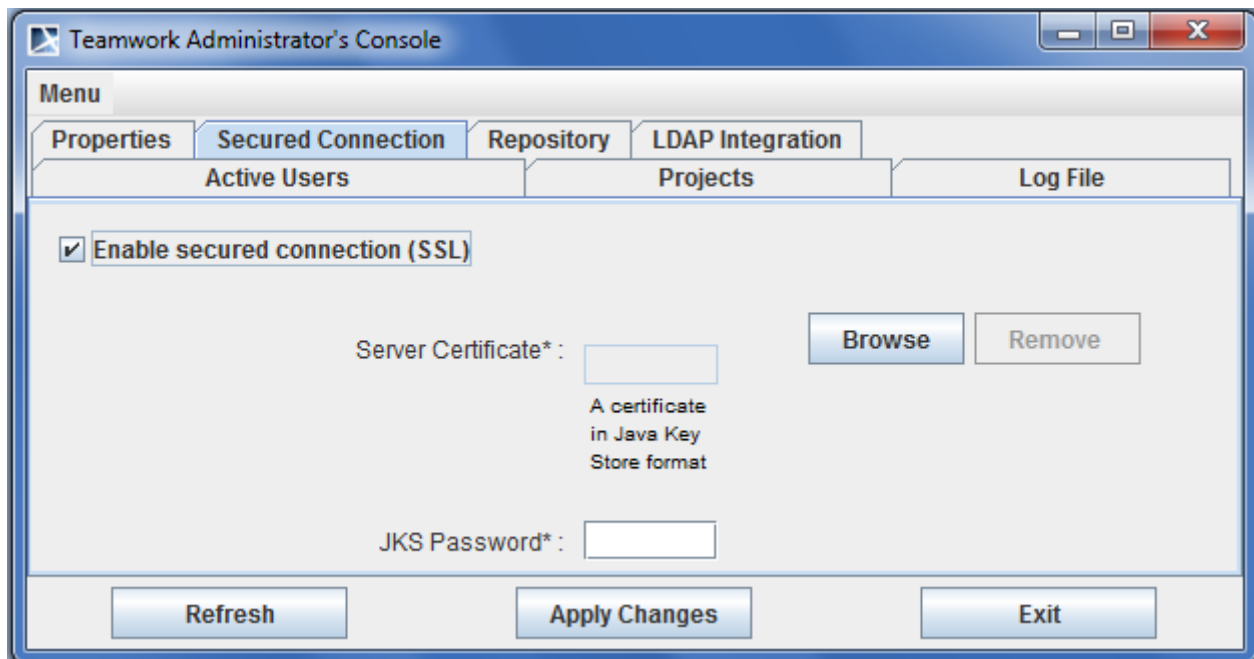
Make sure all users are logged out before restarting the server.

If you want to change the password for the certificate, you need to regenerate the certificate with a different password.



Only the administrator can configure Teamwork Server options. The administrator should be disconnected from Teamwork in MagicDraw while using the Teamwork Administrator's Console.

To remove a certificate, click the **Remove** button.



Teamwork Administrator's Console. Secured Connection tab

4.5.6 Repository tab

The Repository tab is very important. It contains information about the repository of the server. This tab determines where and how Teamwork Server stores projects and their version information, user lists, etc.

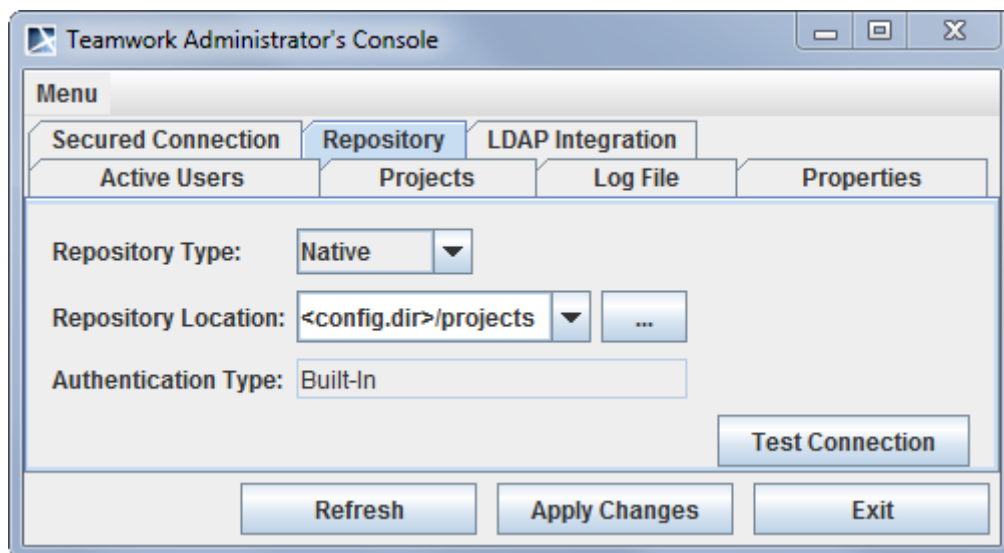
- For more information about repositories, see [Teamwork System Design](#)(see page 9).
- Restart the Server to apply changes.

There are two different types of repository to choose from in the Repository type combo box. This section describes the configuration fields in detail. See [Teamwork System Design](#)(see page 9) for general information.

The Repository tab layout changes with the type of repository selected.

Native Repository

The Native repository type is the simplest one to configure. There is only one editable parameter, the **Repository Location**.



Teamwork Administrator's Console. Repository tab for Native repository type

SVN Repository

The screenshot shows the 'Teamwork Administrator's Console' window with the 'Repository' tab selected. The 'Repository Type' is set to 'SVN'. A warning message states: 'ATTENTION! Integration with Subversion downgrades Teamwork Server performance considerably in comparison to the Native repository. After integration, reliability of the Teamwork server will directly depend on reliability of the link between the Teamwork and Subversion server. Availability of the Subversion server will directly influence accessibility of models stored in Teamwork Server.' The 'Repository URL' is 'svn://drambllys/trunk/vidjur'. The 'Project Cache Directory' is '<install.root>/projects'. The 'Server Configurations Location' is 'projects'. The 'System Profile Location' is 'projects'. The 'Authentication Type' is 'Remote Login'. The 'Administrator login name' is 'admin'. The 'Administrator password' is masked with dots. There is a checkbox for 'Auto-add unknown users to Teamwork if they log-in successfully' and a 'Test Connection' button. At the bottom are 'Refresh', 'Apply Changes', and 'Exit' buttons.

Teamwork Administrator's Console

Menu

Properties Secured Connection **Repository** LDAP Integration

Active Users Projects Log File

Repository Type: SVN

ATTENTION! Integration with Subversion downgrades Teamwork Server performance considerably in comparison to the Native repository. After integration, reliability of the Teamwork server will directly depend on reliability of the link between the Teamwork and Subversion server. Availability of the Subversion server will directly influence accessibility of models stored in Teamwork Server.

Repository URL: svn://drambllys/trunk/vidjur

Project Cache Directory: <install.root>/projects

Server Configurations Location: projects

System Profile Location: projects

Authentication Type: Remote Login

Administrator login name: admin

Administrator password: ..

☐ Auto-add unknown users to Teamwork if they log-in successfully

Test Connection

Refresh Apply Changes Exit

Teamwork Administrator's Console. Repository tab for SVN repository type

When configuring Teamwork Server to use the SVN repository, the SVN client executable must be installed on the computer running Teamwork Server. The SVN client executable must be accessible on the system's PATH and have appropriate permissions to execute. The SVN repository itself can be located on the same or a different computer than the one running Teamwork Server.

4.5.7 LDAP Integration tab

For more information about LDAP Integration, see [LDAP Support](#)(see page 53).

4.6 Administrating Server via Command Line Utility

4.6.1 Administrating Server via Command Line Utility


You can perform administrative tasks, such as users, projects, and permissions management, using a command line utility. The utility is stored in <Teamwork Server installation directory>\bin. It is called `teamwork_console.exe` for Windows and `teamwork_console` for Unix and Unix-like operating systems.

The ability to access the server administrative functions via the command line utility facilitates the scriptable management of Teamwork Server. This enables automation of routine administrative tasks, such as permission management. Single commands can be given through the command line utility parameters. Multiple commands can be given in bulk through the input stream. Command results are provided through exit status codes, enabling conditional script execution.

To perform administrative tasks using the command line utility

1. Start Teamwork Server.
2. Open the command line interface.
3. At the command line, type the following command:

cd <Teamwork Server installation directory>\bin

 **cd** C:\Downloads\MagicDraw_180_sp2_teamwork_server_no_install\bin

4. At the command line, type the following command:
teamwork_console.exe -u username -p password servername

 **teamwork_console.exe** -u Administrator -p Administrator localhost

You are successfully logged into the server and can now perform server administration tasks.

The following table includes the command line utility commands for performing administrative tasks and their brief descriptions.

Command	Description
mkuser	Creates an external user.
rmuser	Deletes the given user.
lsuser	Lists all users or tests if the given user exists.
mkproj	Creates an empty project.
rmproj	Deletes the given project.
lsproj	Lists all projects or tests if the given projects exist.
stperm	Sets the specified permission of the specified user on a given project.

Command	Description
clperm	Clears the specified permission of the specified user on a given project.
lsperm	Lists permissions of the specified user on a given project.

For detailed command descriptions, refer to the help section, printed after typing any of the following commands at the command line:

- **teamwork_console.exe --help**
- **teamwork_console.exe -h**

Any of the commands in the preceding table can be typed at the command line immediately after the *servername* parameter.

Example: **teamwork_console.exe -u Administrator -p Administrator localhost lsuser**

✓ If your Teamwork Server runs on Linux or another Unix-based operating system, you can avoid typing a long command every time by using the following command once:
alias tc="/teamwork_console -u username -p password servername"
 In the sequel, you can type "**tc**" instead of "**teamwork_console -u username -p password servername**", for example, "**tc lsuser**".

4.6.2 Giving commands on the input stream

The commands for performing supported administrative tasks can be given on the input stream. This is indicated by specifying the *"-"* parameter immediately after the *servername* parameter.

Example: **teamwork_console.exe -u Administrator -p Administrator localhost -**

In this case the command line utility connects to the specified server with the given credentials, performs the commands and exits.

Example:

This is a shell script for creating a project and assigning users to it:

```
#!/bin/bash
#shorthand
TC='/MagicDraw_TeamworkServer_installation_directory/bin/ teamwork_console -u
Administrator -p Administrator localhost'
#create a blank project in a category, store output in a variable,
#exit on fail
PID=$( $TC mkproj "TestPr" "Training Material and Demos" ) || { echo "Failed to
create project" >&2; exit 1; }
#pump multiple set permission commands into the teamwork console$TC - << EOF
stperm user1 RD $PID
```

```
stperm user1 RR $PID
stperm user1 WR $PID
EOF
```

4.7 Data Migration between Different Repositories

Two different types of repositories are supported in MagicDraw v12.5 and above. The server uses the Native repository type by default when first installed. Switching to a different repository type inevitably causes the project migration task to increase.

It is possible for the Teamwork Server to import and export projects, triggered from the Administrators console (read Administrator's Console Dialog). *Import & Export is only possible in the Native repository type format.* The Native repository type is a kind of intermediate form for information interchange.

Note that the terms "import" and "export" are used relatively to the currently running server (the one to which administrator's console is attached):

- import: importing data into the current server from the designated directory.
- export: exporting data from the current server into the designated directory.

Migrating the server from the Native repository to the SVN repository

1. Start the Server on the Native repository (projects/versions stored in a directory).
2. Open the Administrator's Console. Reconfigure the server to work with the SVN repository.
3. In the **Administrator's Console**, reconfigure the server to work with the SVN repository.
4. Restart the server to use the new repository.



Server users

Make sure all users are logged out before restarting the server.

5. The Server starts. The only things in the repository are the profiles needed to work with MagicDraw.
6. Login again to the **Administrator's Console** and trigger the project import. Select the directory to import from, the same directory where the export was performed.
7. Projects are now in a new SVN repository.

4.8 Changing the Teamwork Server Debugging Mode

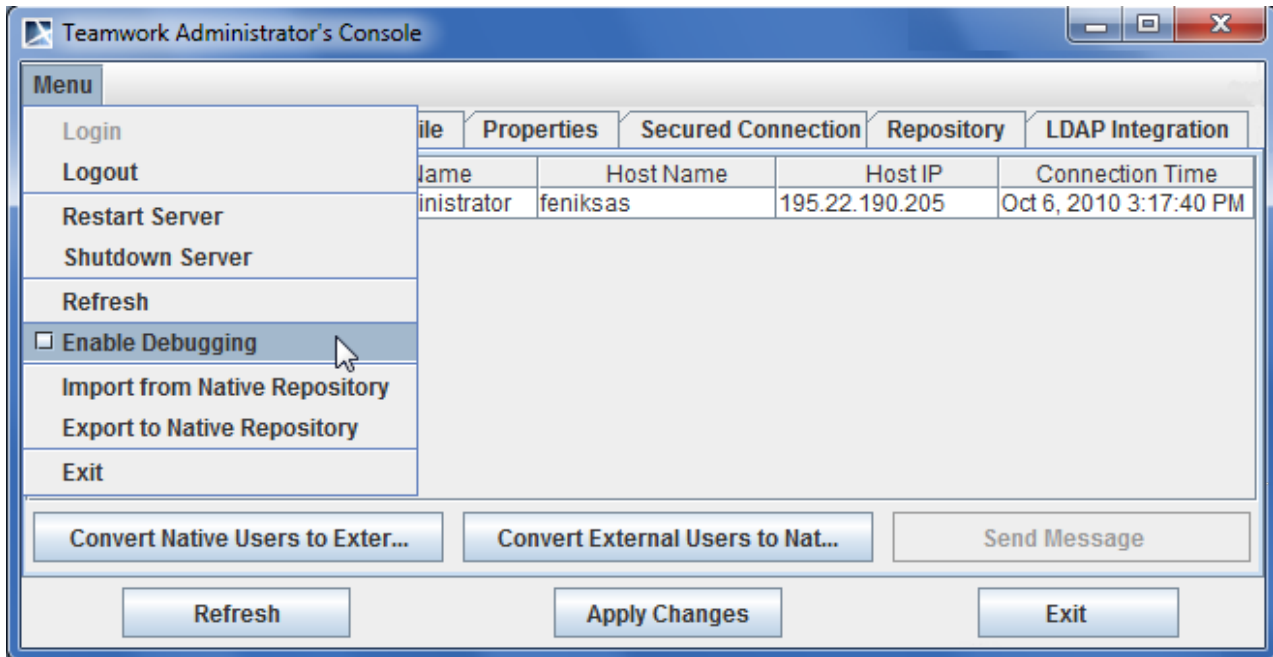
You can change the Teamwork Server debugging mode without restarting.

To enable Teamwork Server debugging

1. Start Teamwork Server Administrator's Console.
2. Select **Enable Debugging** from the **Menu**.



Enable Debugging is disabled in the Offline Administrator's Console mode.



4.8.1 Teamwork Administrator's Console - Enable Debugging

4.9 Configuring server properties

You can configure the server properties in the `teamwork_server.properties` and `wrapper.conf` files manually.

i The procedure is valid for Windows only as other operation systems store configuration files and projects in the installation directory by default.

To store configurations and projects in the Teamwork Server installation directory

1. Stop Teamwork Server.
2. Open the `<Teamwork Server installation directory>\teamwork_server.properties` file.
3. In the `JAVA_ARGS` line, add the argument `-DWINCONFIG=false` and save the file.
4. Do the same changes in `stop_teamwork_server.properties`, `teamwork_administrator.properties`, and `teamwork_server_nogui.properties` files.
5. Open the `<Teamwork Server installation directory>\service\conf\wrapper.conf` file.

wrapper.conf file configuration

Do not modify any of the properties when an application using this configuration file has been installed as a service or daemon. Please uninstall the service before modifying this file. The service must then be reinstalled.

6. Locate property name *wrapper.java.additional.5*, and in the next line add the argument *wrapper.java.additional.6=-DWINCONFIG=false* and save the file.
7. Start Teamwork Server.

Related pages:

- [Managing Teamwork Server](#)(see page 27)

5 LDAP Support

LDAP Integration allows Teamwork Server to authenticate its users against LDAP servers. LDAP Integration enables pass-through MagicDraw authentication against LDAP servers by passing client's authentication information to LDAP servers.

LDAP Integration supports Simple User+Password and SASL authentication, SSL/TLS protocols, and several LDAP servers configured for a single integration.

Related Pages:

- [Enabling LDAP Integration](#)(see page 53)
- [Connection Settings](#)(see page 54)
- [Authentication Settings](#)(see page 55)
- [User Data Retrieval Settings](#)(see page 61)
- [Connection Testing](#)(see page 64)
- [Subversion and LDAP Integration Working at the Same Time](#)(see page 64)
- [Converting Certificates to JKS Format](#)(see page 64)
- [Integrating Teamwork Server with a SSL-Enabled Active Directory](#)(see page 65)

5.1 Enabling LDAP Integration

To enable LDAP Integration

-
1. Start Teamwork Administrator's Console. Refer to [Starting the Administrator's Console](#)(see page 36).
 2. Click the **LDAP Integration** tab.
 3. Click **Enable LDAP Integration**. LDAP integration settings become active.
 4. After enabling the integration, specify all mandatory setting values. Mandatory settings are marked with the star sign at the end of the name. There are three groups of settings in the **LDAP Integration** tab:
 - [Connection Settings](#)(see page 54)
 - [Authentication Settings](#)(see page 55)
 - [User Data Retrieval Settings](#) (see page 61)
 5. Click **Apply Changes** when you are done. You will be required to [test the connection](#)(see page 64) with the LDAP server.

5.2 Connection Settings

Connection Settings specifies network and security settings for connecting to LDAP servers.

The screenshot shows the 'Teamwork Administrator's Console' window with the 'LDAP Integration' tab selected. The 'Connection Settings' section is active, showing the following fields and controls:

- Enable LDAP Integration:** A checked checkbox.
- Test Connection:** A button to test the LDAP connection.
- Server Address(es)*:** A text field containing 'main, sunldap.nomagic.com:14408'. Below the field, a note states: 'Several addresses can be entered by separating them with spaces. The default port number is 389. If you wish to use a different port, enter addresses in the form of server:port, e.g. example.com:636'.
- Server Timeout*:** A text field containing '500'. Below the field, a note states: 'Server timeout in milliseconds'.
- Encryption Protocol*:** A dropdown menu currently set to 'None'.
- Buttons:** 'Refresh', 'Apply Changes', and 'Exit' buttons are located at the bottom of the console.

5.2.1 Teamwork Administrator's Console, LDAP Integration tab. Connection Settings

The following table describes **Connection Settings**.

Setting Name	Description
Server Address(es)	<p>A list of servers separated by spaces. Each entry holds the server address and server port. If unspecified, the 389 port is used. At least one server address must be specified. Usually a master server and its slaves (replicas) are specified for round-robin authentication.</p> <p>Teamwork Server authenticates against servers in the order they are listed in the Server Address(es). If authentication does not succeed with the first server in the list, the second server is used, and so on until authentication is successful.</p> <p>Authentication failure for the whole integration is considered an inability to authenticate against any of the specified servers.</p> <p>A single server in the specified list is queried within the period of time specified in the Server Timeout setting.</p>
Server Timeout	<p>A time duration specifying the maximum period of time in milliseconds to successfully authenticate to a single server. If authentication is unsuccessful within this period of time, the next server in the server list is queried. The default value for this option is 500 milliseconds.</p>
Encryption protocol	<p>A list of protocols. You can use SSL or TLS protocols for the encryption. Select None if you do not need to use an encryption protocol. The selected protocol applies to every server specified in the LDAP server list. For example, if the SSL encryption is specified, communications to all servers specified in the Server Address(es) list will be encrypted using the SSL protocol.</p> <p>If an encrypted connection is used, a Server Root CA certificate and password may be required. The Server Root CA certificate should be stored in a Java Key Store (JKS) format. For more information about converting certificates to JKS format, see Converting Certificates to JKS Format(see page 64).</p>

5.3 Authentication Settings

LDAP Integration supports the two most popular LDAP authentication methods:

- Simple User+Password: [Authentication settings for the Simple User+Password authentication type](#)(see page 56).
- SASL: [Authentication settings for the SASL authentication type](#)(see page 60).

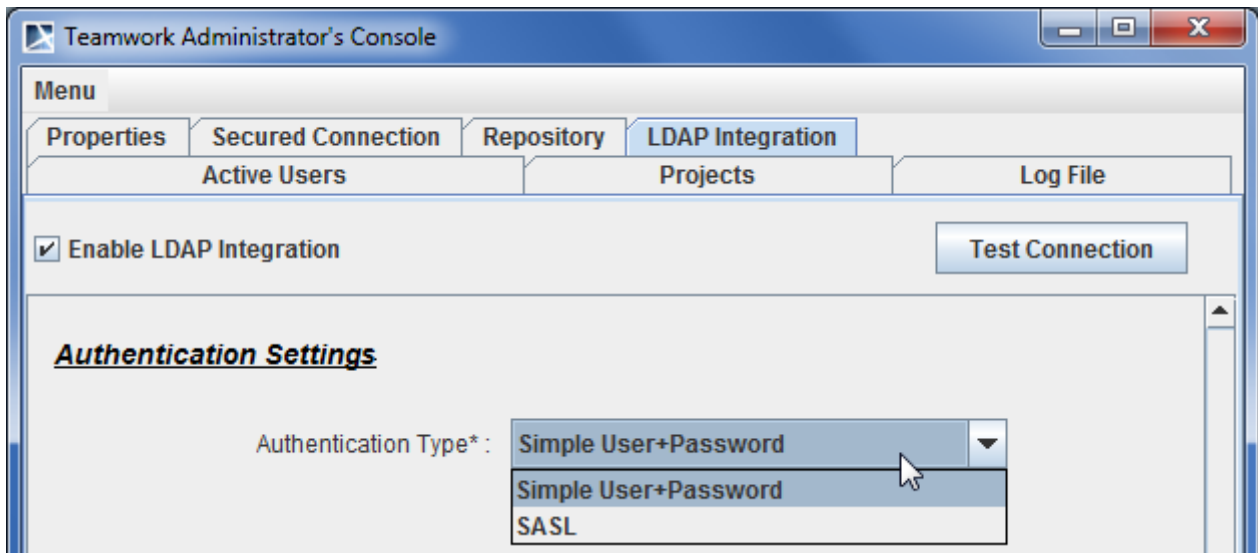
Teamwork Server transforms user credentials entered in MagicDraw to LDAP authentication credentials using the templates in authentication settings. After successful authentication to LDAP, a special user for each authenticated LDAP user is created in Teamwork Server. They differ from ordinary users in that they have no passwords. Authentication to LDAP server(s) is used to complete authentication. You

can perform various actions for these users, such as setup permissions, remove users, and other common actions.

You can automatically create a proxy user account for Teamwork Server. Select **Auto-add unknown users if they login successfully** checkbox. Authenticated user credentials without passwords will be stored in Teamwork Server on successful login.

To select an authentication type for LDAP integration

- In the **Authentication Type** list, select a desired protocol.



5.3.1 Teamwork Administrator's Console, LDAP Integration tab. Selecting Authentication Type

5.3.2 Authentication setting for the Simple User + Password authentication type

You can select the following options using the Simple User+Password authentication type:

- **Use User DN template**
- **Retrieve User DN by using an LDAP query**

Authentication using a user DN template has the following characteristics:

- A hard-coded template is filled in with the user login supplied on logging in to Teamwork Server.
- The user DN is used to login to LDAP server.

Authentication using retrieved user DN occurs in the following order:

1. A query template is filled in with the login name entered by the user.

2. An anonymous bind or specific User DN and password is used to connect to the LDAP server.
3. The LDAP server is queried for the User DN using the query produced in the step #1, **Search Base** and **Search Scope** settings values.
4. The LDAP server returns User DN by the query.
5. Teamwork disconnects from the LDAP server.
6. Teamwork tries to login to LDAP using the returned User DN and password supplied by the user during the login process.

Teamwork Administrator's Console

Menu

Properties Secured Connection Repository **LDAP Integration**

Active Users Projects Log File

☒ Enable LDAP Integration Test Connection

Authentication Settings

Authentication Type*: Simple User+Password

☐ Use User DN template

User DN*: uid=\$(login),ou=People,dc=example,dc=com
Example: cn=\$(login),dc=example,dc=com

☒ Retrieve User DN by using an LDAP query

Query*: (&(givenName=\$(login))(|(ou=Accounting)|(ou=...
Example: (uid=\$(login))

Search Base*: dc=NoMagic,dc=com
Example: dc=example,dc=com

Search Scope*: Subtree

Anonymous Bind: ☐

Bind DN*: uid=mdtest,ou=People,dc=nomagic,dc=com
Example:
cn=admin,dc=people,dc=example,dc=com

Bind Password*:

☒ Auto-add unknown users if they login successfully

Refresh Apply Changes Exit

Teamwork Administrator's Console, LDAP Integration tab. Authentication Settings (Simple User+Password)

The following table describes Authentication **Settings**.

Setting Name	Description
Settings that are active when the Use User DN template is selected	
User DN	<p>User DN stores a template, used to map the user's authenticating against Teamwork Server to LDAP distinguished names when authenticating. The template recognizes a single keyword \$(login). An example of the template:</p> <p><i>cn=\$(login), dc=example, dc=com</i></p>
Settings that are active when the Retrieve User DN by using an LDAP query is selected.	
Query	<p>The LDAP query for retrieving User DN, for example:</p> <p><i>uid=\$(login)</i></p>
Search Base	<p>DN from which a search should begin, for example:</p> <p><i>dc=example, dc=com</i></p>
Search Scope	<p>Specifies whether the search must be restricted only to the directly-owned DNs or performed in the whole subtree.</p> <p>Choose one of the following:</p> <ul style="list-style-type: none"> • One level • Subtree
Anonymous Bind	<p>A mode of bind, specifying whether the user connects to LDAP server with a specific user or anonymously to find the User DN corresponding to the user trying to log in to Teamwork. You must have this type of user if you do not have anonymous access.</p>
Bind DN	<p>Specific User DN for connecting to the LDAP server and performing queries. This element is active when Anonymous Bind is not selected.</p>

Setting Name	Description
Bind Password	A specific password for connecting to the LDAP server and performing queries (you must have this type of user if you do not have anonymous access). This element is active when the Anonymous Bind is not selected.

5.3.3 Authentication settings for the SASL authentication type

The screenshot shows the 'Teamwork Administrator's Console' window with the 'LDAP Integration' tab selected. The 'Authentication Settings' section is visible, showing the following configuration:

- Enable LDAP Integration:** ☒ (A 'Test Connection' button is located to the right of this checkbox.)
- Authentication Type*:** SASL (selected from a dropdown menu)
- Authentication Identity*:** \$(login) (Example: \$(login))
- Authorization Identity:** \$(login)@NoMagic (Example: \$(login))
- Realm:** (Empty text field)
- Mechanism:** (Empty text field, Example: DIGEST-MD5)
- Auto-add unknown users if they login successfully:** ☒

At the bottom of the console, there are three buttons: 'Refresh', 'Apply Changes', and 'Exit'.

Teamwork Administrator's Console, LDAP Integration tab. Authentication Settings (SASL)

Authentication Settings are described in the following table.

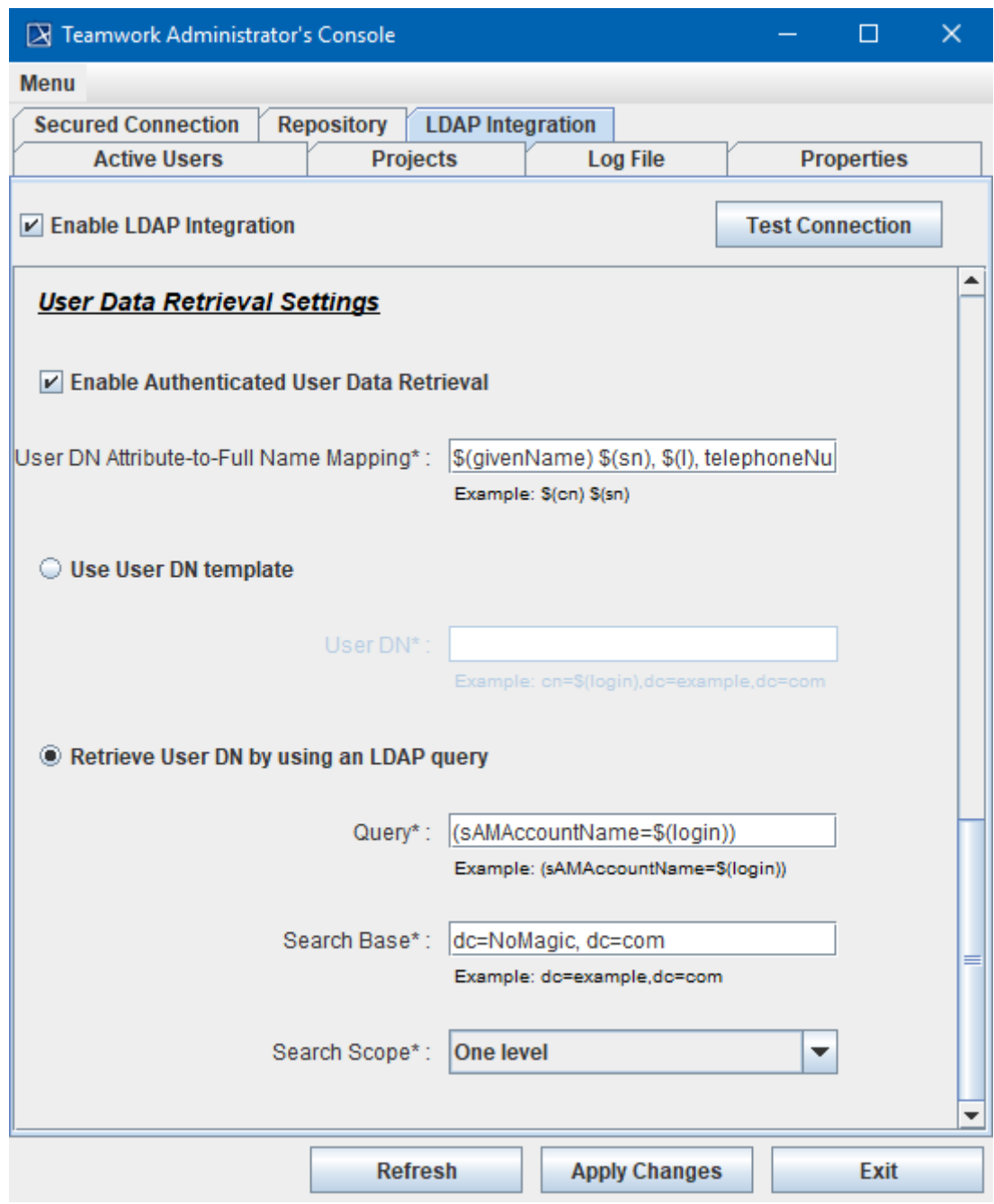
Setting Name	Description
Authentication Identity	<p>Login name supplied by the user is transformed to an authentication identity when authenticating. The authentication Identity is a mandatory template. The template recognizes a single keyword \$(login). An example of a template:</p> <p>\$(login)</p>
Authorization Identity	<p>Login name supplied by the user is transformed to an authorization identity when authenticating if the Authorization Identity template is specified. The template recognizes a single keyword \$(login).</p> <p>An example of a template:</p> <p>\$(login)</p> <p>or</p> <p>\$(login)@example</p>
Realm	Specifies the realm of an authentication identity for the SASL bind.
Mechanism	<p>Specifies the SASL mechanism to be used for authentication. An example:</p> <p>DIGEST-MD5</p>

5.4 User Data Retrieval Settings

Authenticated users usually have access to User DN attributes in the LDAP database. If user information retrieval is enabled and User DN attributes are accessible to the authenticated user, Teamwork Server retrieves their values and sets them for the corresponding external users.

If the **Simple User+Password** authentication type is enabled (either by using a static User DN template or by querying the LDAP server(s) for User DN), the User DN is retrieved in the same way. This connection is further reused for retrieving user information when the user logs in to the LDAP server.

Teamwork Server creates an external user with the login name specified by the user upon authentication if the user information retrieval is disabled or User DN attributes are not accessible to the authenticated user.



5.4.1 Teamwork Administrator’s Console, LDAP Integration tab. User Data Retrieval Settings

User Data Retrieval Settings are described in the following table:

Setting Name	Description
User DN Attribute-to-Full Name Mapping	<p>After a specific User DN is found, the name of a local user created on the authentication is generated using the Full Name Mapping template for this User DN. The Full Name Mapping template supports placeholders in the form of $\\$(attribute)$, where attribute is an attribute of DN.</p> <p>An example:</p> <p>$\\$(cn) \\(sn)</p> <p>This will form the Name of the created user out of two LDAP attributes - <i>cn</i> and <i>sn</i>.</p>
Settings that are active when the Use User DN template is selected.	
User DN	<p>User DN specifies a template used to search for a specific DN by the supplied login name. An example:</p> <p>$cn=\\$(login), dc=example, dc=com$</p>
Settings that are active when the Retrieve User DN by using an LDAP query is selected.	
Query	<p>An LDAP query for retrieving User DN, for example:</p> <p>$sAMAccountName=\\$(login)$</p> <p>$\\$(login)$ is a username a user types when connect to a server¹⁷ from the modeling tool.</p> <p><i>aSAMAccount</i> is used for authorization to Windows OS and Active Directory as a directory service is set by default. You need to contact your system administrator if other OS or parameters are used.</p>
Search Base	<p>DN where the searching should begin, for example:</p> <p>$dc=example, dc=com$</p>

¹⁷ <https://docs.nomagic.com/display/MD190SP4/Starting+a+collaboration+session>

Setting Name	Description
Search Scope	<p>Specifies whether the search must be restricted to the directly owned DNs only or performed in the whole subtree.</p> <p>Choose one of the following options:</p> <ul style="list-style-type: none"> • One level • Subtree

5.5 Connection Testing

After specifying the LDAP Integration setting values, you can test the connection to the LDAP server.

To test LDAP Integration configuration

1. In the **LDAP Integration** tab of Teamwork Administrator's Console, click **Test Connection**. The **Test Connection** dialog opens.
2. Type the user's login name, password, and click **OK**.
3. The message with connection results appears.

5.6 Subversion and LDAP Integration Working at the Same Time

If Teamwork Server uses Subversion for storing projects and LDAP Integration for authentication, Subversion must recognize (or authenticate) the same users and passwords used to authenticate against the LDAP server. The Administrator is the user specified in the LDAP Integration configuration.

When Teamwork is integrated with Subversion only, the client's authentication information is passed to the Subversion server. When Teamwork Server is integrated with Subversion and LDAP, client's authentication information is passed to both Subversion server(s) and LDAP server(s), but only successful authentication to the LDAP server(s) successfully logs the user into Teamwork Server.

5.7 Converting Certificates to JKS Format

Teamwork Server recognizes certificates stored in *Java KeyStore (JKS)* format. If certificates are in *PEM* format, they must be converted to *JKS* format. You can use OpenSSL (<http://www.openssl.org/>) and *Sun Java KeyTool* (included with every Java distribution) for this purpose. For example, if we have a certificate in a file called *cert.pem*, then the following commands will convert it to *JKS* format:

```
openssl x509 -in cert.pem -out cert.der -outform der
keytool -importcert -alias mycert -file cert.der -keystore truststore.jks
```

5.8 Integrating Teamwork Server with a SSL-Enabled Active Directory

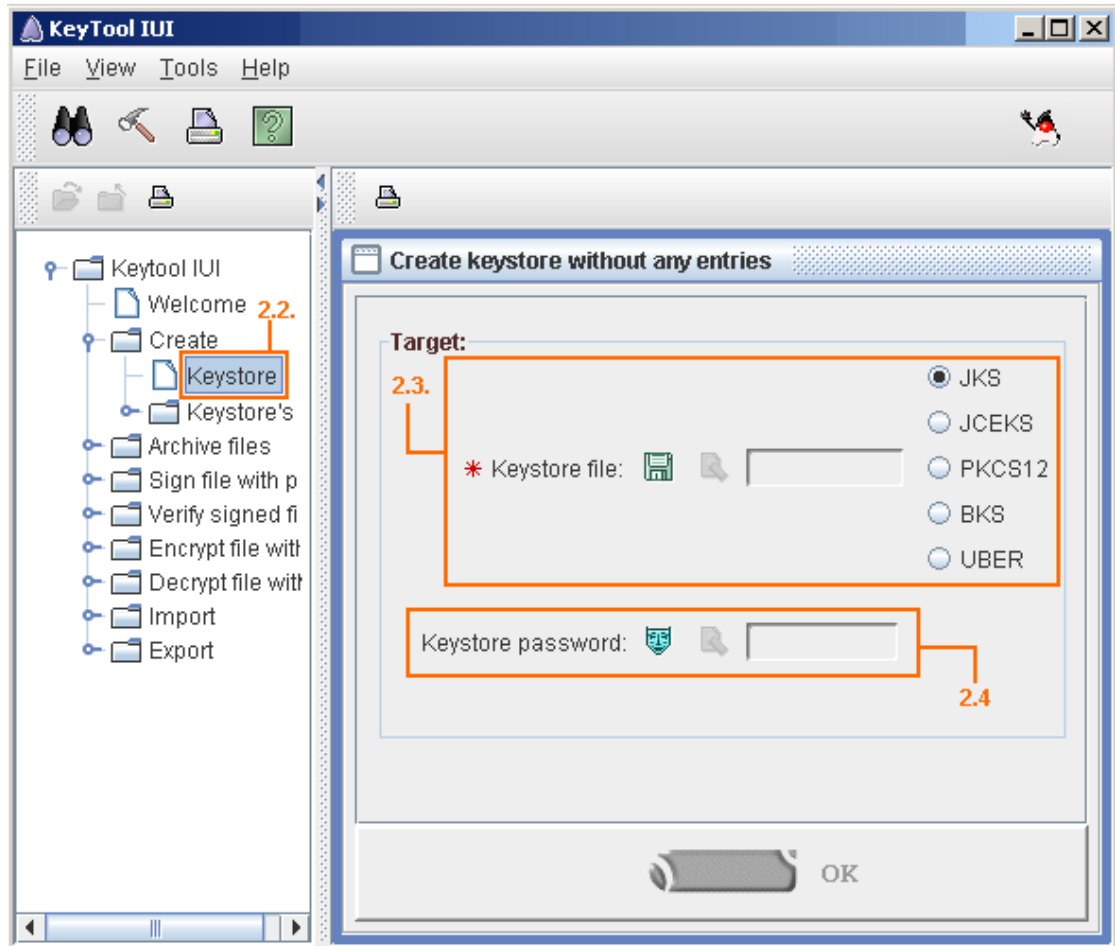
You can integrate Teamwork Server with SSL-enabled Microsoft Windows 2000 and Microsoft Windows 2003 Active Directory.

To successfully complete this integration, the following requirements should be met:

- Windows Server Active Directory should have SSL enabled. This includes a valid Certificate Authority (CA) and a valid certificate for Active Directory (AD) server certificate (for more information on installing and configuring Certificate Services for Windows Server, see Microsoft documentation).
- Any SSL-aware LDAP client should be able to connect to your AD server port 636 with SSL enabled. It should also have access to its contents (for more information on setting SSL-enabled connections to AD, refer to the specific LDAP client documentation).

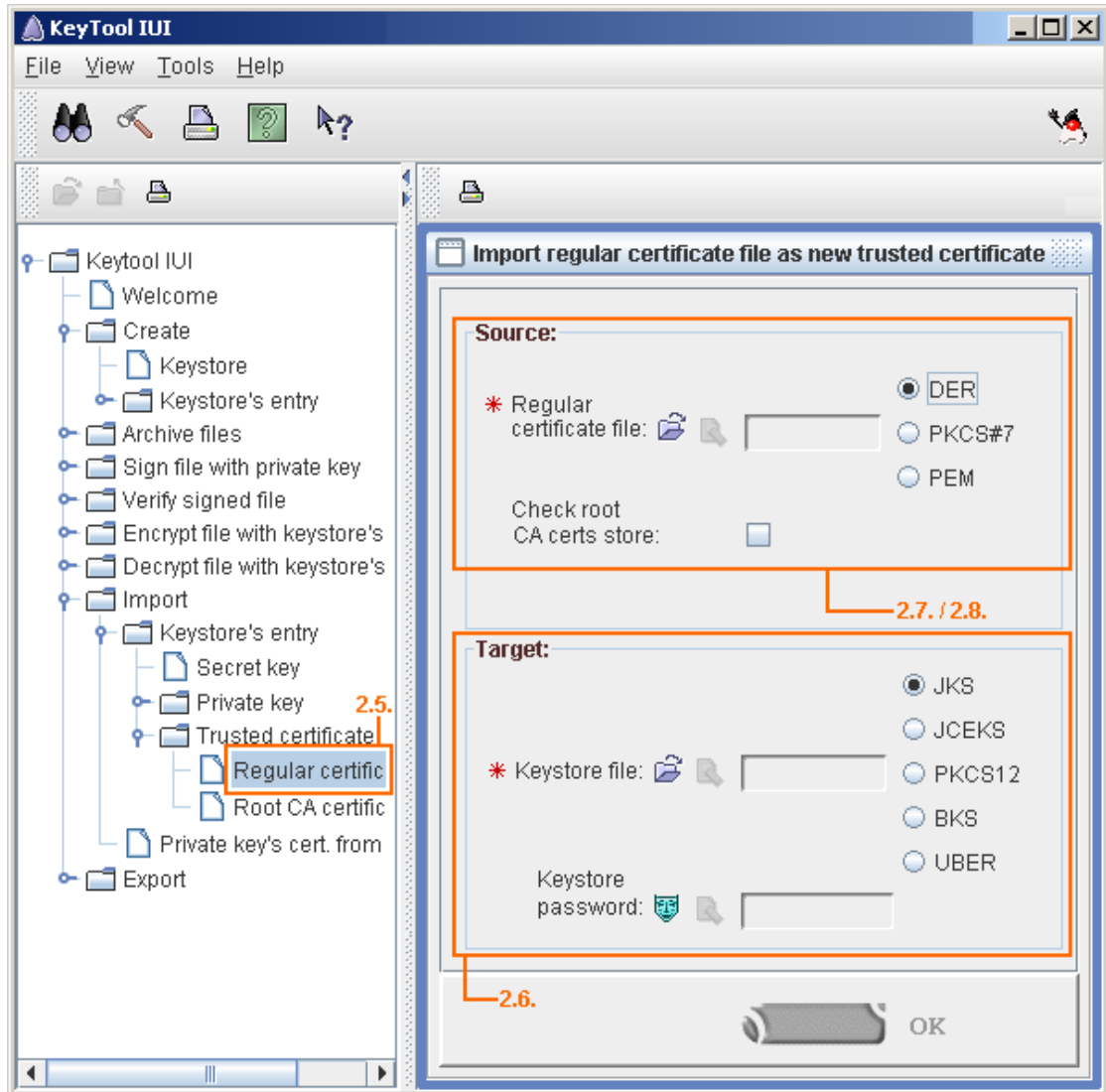
To create a Java KeyStore with the included CA and AD server certificates

1. Export the CA and AD server certificates to the DER encoded binary.cer files using the Certificates Snap-in on the Microsoft Management Console.
 - Do not include private keys while exporting.
2. The following steps outline how to import the CA and AD server certificates (.cer files) to Java KeyStore (.jks file), using the KeyTool IUI:
 - a. Run the KeyTool IUI.
 - b. Double-click **Create** on the tree, and then click **Keystore** to create a new keystore file.
 - c. Choose the JKS format and save a new keystore file.
 - d. Set the password for the keystore.



5.8.1 Creating a new keystore file with the KeyTool GUI (steps 2.2, 2.3, and 2.4)

- e. On the tree, double-click **Import**, **Keystore's Entry** and **Trusted Certificate**, and then click **Regular Certificate** to import the .cer files into Java KeyStore.
- f. Select the created keystore file as the target.
- g. Select the exported CA certificate file (.cer) as the source. Enter the keystore password and click **OK**. Enter "CAAlias" as the alias for the CA certificate and click **OK**.
- h. Select the exported AD server certificate file (.cer) as the source. Enter the keystore password and click **OK**. Enter the full name of the AD server as the alias for the AD certificate and click **OK**.



5.8.2 Importing certificate files into the keystore file (steps 2.5, 2.6, 2.7, and 2.8)

Now you have a Java KeyStore containing both certificates.

The subsequent steps for the Teamwork Server integration with SSL-enabled Active Directory are the same as for the integration with any other LDAP server. This procedure is described in [Enabling LDAP Integration](#)(see page 53).

6 Connecting Teamwork server via SSH encrypted tunnel

This section describes SSH-encrypted MagicDraw UML and Teamwork Server connectivity.


Requirements for both server and client include:

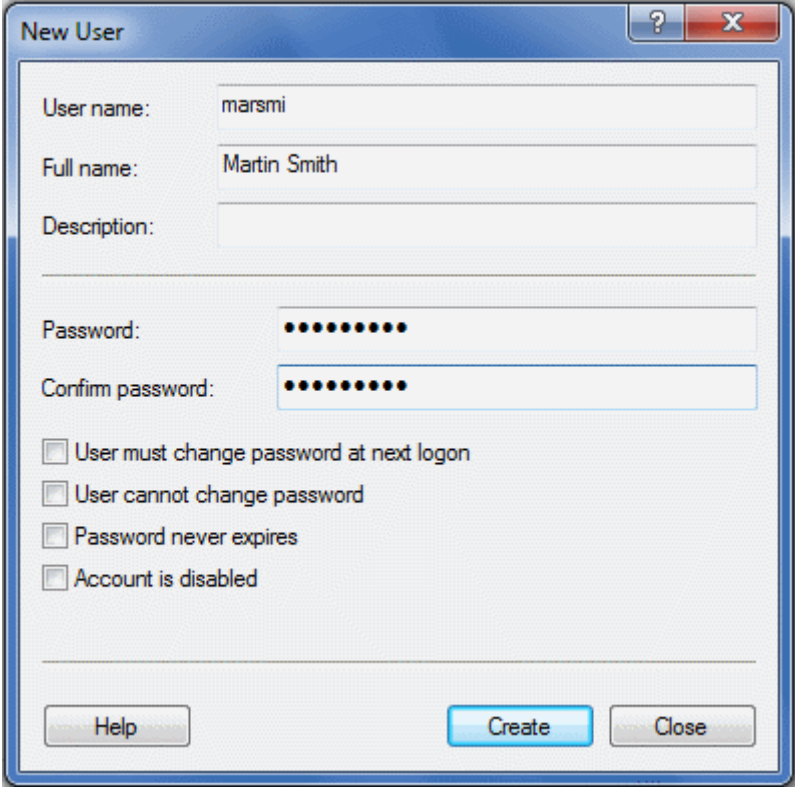
1. Windows 2000/XP/Vista or Windows Server family OS.

2. Local administrative rights to create local users for tunneling and to run SSH service.
3. OpenSSH server and client binary installation files. You may obtain it free of charge from http://sourceforge.net/project/showfiles.php?group_id=103886&package_id=111688#¹⁸.

6.1 Configure the Teamwork Server side

1. Log in to the Teamwork Server with administrator privileges.
2. Install OpenSSH with default settings. This will install the OpenSSH server and client on your Teamwork Server machine. A warning about editing password and group file will appear while installing. Click **OK**.
3. Create a local user for SSH tunnel. To do this correctly, click on **My Computer**, then select **Manage**. In the Local Users and Groups section right-click on **Users** and choose **New User**. The New User dialog opens.
 - a. Enter a new username to log in into SSH service to establish tunneling. For example, *tunnel*.
 - b. Enter the user password according to your local system policy.
 - c. Clear the **User must change password at next logon** check box.
 - d. Click **Create**. The local user will be created.

 This is not the same as the MagicDraw Teamwork Server user used to check out and commit UML models from/to the server. Use Teamwork Administrator to manage Teamwork users.

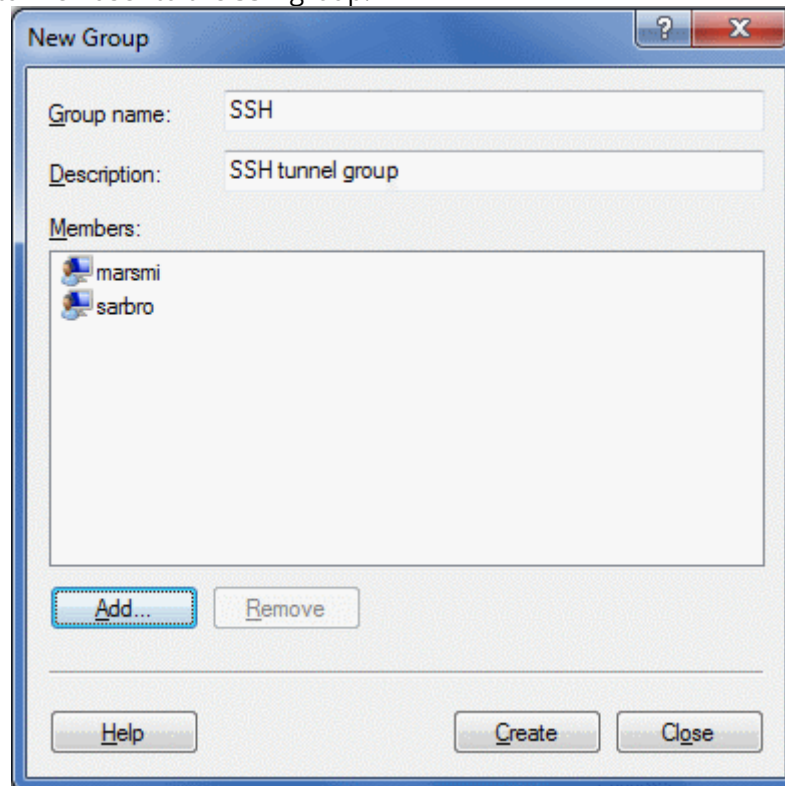


¹⁸ http://sourceforge.net/project/showfiles.php?group_id=103886&package_id=111688

6.1.1

The New User dialog


4. Create a local group for SSH tunnel users. To do this right-click on **My Computer** and then select **Manage**. In the Local Users and Groups section, right-click on **Group** and choose **New Group**. The New Group dialog opens.
 - a. Enter a new groupname, for example, *SSH*.
 - b. Add the "tunnel" user to the SSH group.



6.1.2

The New Group dialog

5. Create SSH-aware local password file with 'tunnel' user entry. Any users in this password file will be able to log on with SSH. To create the SSH-aware local password file run command prompt (click "Start"-> "Run", then type "cmd" and click Enter) and then type the following commands:
 - a. `cd C:\Program Files\OpenSSH\bin`
 - b. `mkgroup -l >> ..\etc\group`
 - c. `mkpasswd -l -u tunnel >> ..\etc\passwd`
6. Start OpenSSH Server service from your control panel. To do this right-click on **My Computer** and then select **Manage**. In the Services and Application section, under the Services item, right-click on the **OpenSSH Server** service and choose **Start**.
7. Test the SSH server.
 - a. Type `"ssh tunnel@localhost"` from your command prompt.

 The following warning appears:
The authenticity of host 'localhost (127.0.0.1)' can't be established. RSA key fingerprint is
xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx. Are you sure you want
to continue connecting (yes/no)?

- a. Type yes and press **Enter**. You must type the full word "yes," not just "y."
- b. Enter the password you created in step 3.
- c. A warning about nonexistent home directory appears. Please ignore it.
- d. Now you are logged in into localhost via SSH service and you can see the shell prompt.
- e. After the SSH server testing, exit the server by typing exit.

Another way to test if the SSH port (port 22) is opened on the server:

In the command prompt go to the C:\Program Files\OpenSSH\bin and type the 'netstat -na' command. You will get the list of all connections. The state of the port should be "LISTENING" while the SSH server is running.

6.2 Client Side Configuration

1. Install OpenSSH client from the same installation file you downloaded. Uncheck the "Server" option when asked.
The SSH client package is installed to "C:\Program Files\OpenSSH" by default and added into your PATH variable.
2. Establish an SSH tunnel by logging into the Teamwork server SSH service from the command prompt:
`ssh -L localhost:teamworkserver:teamworkserverport username@teamworkserver`
For example, the following command will establish an SSH encrypted tunnel from client port 1100 to server port 1100. When connecting to localhost:1100, the packets are encrypted and sent to twserver:1100, where the actual Teamwork server resides:
 - a. `ssh -L 1100:twserver:1100 tunnel@twserver`
localhost may be any unused port on your workstation.
The Teamwork server port is the port the Teamwork server is running (usually 1100)
You are logged as user "tunnel" to the SSH service on Teamwork server machine. Leave the session open, as killing it also kills the SSH tunnel used for the MagicDraw Client.
3. Open MagicDraw Client. Use localhost:localhost when connecting to the Teamwork server. In this case, localhost is 1100, as we used it when creating the tunnel.

i Using any value other than "localhost" or "127.0.0.1" will fail, even if connected to the actual machine name, resolved by DNS. This is because the tunnel starts on a loopback interface of your workstation for security reasons.

7 Server Synchronization and Multisite Deployment

Related Pages:

- [Synchronization Overview](#)(see page 70)
- [Using Synchronization](#)(see page 72)
- [Running Synchronization](#)(see page 76)

7.1 Synchronization Overview

The ability to synchronize data between several Teamwork Servers is available as of version 17.0.3. One Teamwork Server can now expose and make projects from different servers accessible, along with owned projects. Server users can access remote projects from different locations in read-only mode and can perform any read-only actions.

There can be more than two servers in the pool. Each server can separately synchronize data from many other servers.

7.1.1 Usage Scenarios, Multisite Deployment

The main usage scenario for the inter-server synchronization capability is a multisite Teamwork Server deployment.

Very often a large company has several different sites, often in different countries, where projects are developed. Usually, the site has very good internal connectivity - LAN-class speeds (1Gbit/s, or at least multiMbit/s) are available on the site. However, connectivity between the sites is not as good - WAN-class speeds (several Mbit/s) are available between sites.

At times, developers at one site are responsible for the development of subsets of projects, while developers at other sites only need to use (but not modify) the artifacts produced by the first team.

In earlier versions, the company had to choose one site to deploy a single Teamwork Server and set up remote connections for developers joining the other sites.

The server performance (project open, commit, update, element locking times) for onsite developers were good. However, offsite developers suffered from degraded performance due to reduced connectivity parameters. This is especially important when managing large projects.

As of this version, the company can deploy many Teamwork Servers - one per site. Offsite developers can connect to their local server and work at LAN-speeds. Synchronization between servers can then be set up.

You can choose the synchronization frequency as necessary - hourly, daily, weekly

You can also set the specific time to perform the synchronization. For example, you may choose to synchronize at night, when internet traffic is reduced.

With synchronization set up, developers can access the projects of other teams on their local server. Any action that does not change the project is acceptable. Users can open the remote projects, browse, search, analyze then, generate reports, and, most importantly, integrate remote projects as used projects/libraries into their projects.

For example, if a team at one site is responsible for developing a requirements project, the team at another site can take this requirements project and incorporate it into their implementation project(s)

Please note that if developers at different sites edit the same project, one server will be designated as the home server of the project. Any developer who needs to edit the project must log onto the home server of the project to edit it.

There are additional, less frequent scenarios when synchronization between the servers can be handy. For example, if a small number of developers generate a project set that must be readable by a large number of users, the company may want to set up a small main development server where developers will not be affected by network traffic, as well as a large server for handling all read-only traffic (possibly in a different network security zone, etc.).

The licensing scheme is straightforward. Each Teamwork Server deployment is licensed separately. Synchronizing multiple sites is NOT counted as one big deployment, but as many separate

deployments. There are no additional charges (such as a separate "enterprise" edition of the server) for using the synchronization feature.

7.1.2 Characteristics

The synchronization process characteristics are:

- Synchronized items are Projects and Categories. The entire project history is synchronized with all commit data, including all branches and comments. The user's data is NOT synchronized. Each server has its independent user list, and the administrator of each server assigns permissions to each server.
- Each project has Server-Affinity, with one writer and (potentially) many readers. Each project has a home server. Projects can be edited only on the home server. All other servers can only read this project (open the project, use it as a read-only used project/ library of the other projects and so on). Having a single writer prevents incompatible modification conflicts between the servers without introducing any additional burden, such as inter-server locks.
- Synchronization is Incremental. The synchronization process can be run many times. Each synchronization process updates the target server to the latest data from the source server. The amount of the data transferred is proportional to any changes since the last synchronization.
- Synchronization is Asynchronous. Servers continue to run asynchronously. Changes on one server are propagated to other servers only during the synchronization procedure. The synchronization procedure can be arbitrarily delayed from the moment changes occur (for example only monthly synchronization).
- Synchronization is Batched; it is not a continuous process. It has clearly defined start and stop points. However, the frequency of synchronization can be arbitrary (for example every 15 minutes).
- Synchronization is Non-Intrusive. While synchronization is running, users can continue to work on both the source and the target servers. It is not necessary to restart the target server after finishing synchronization.



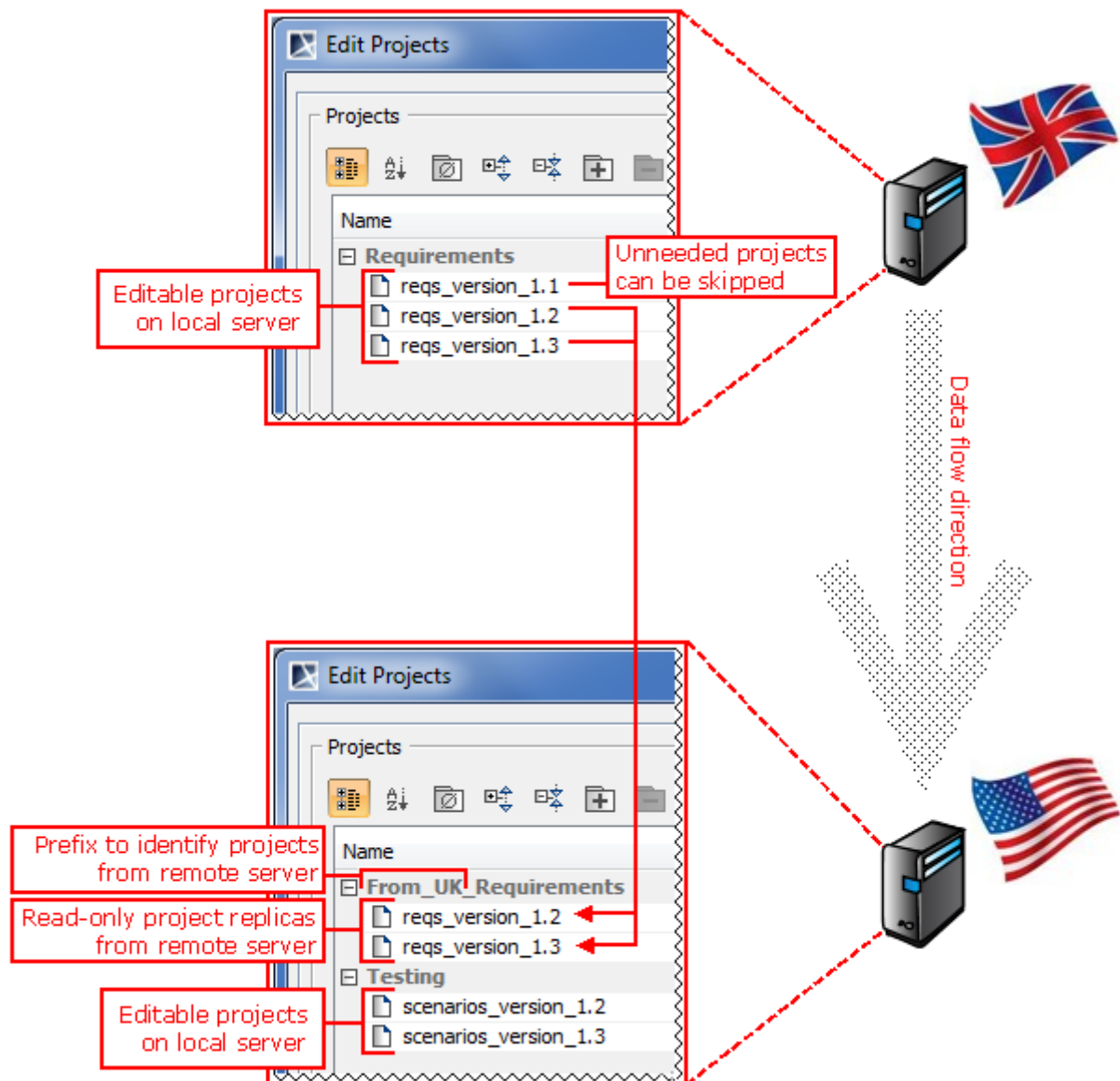
Logging out

Make sure all users are logged out before restarting the server.

- Synchronization is Separate in each Direction. Synchronization between each source and target server pair is independent of any different server pair. This includes synchronization in the opposite direction. If server S1 is synchronized from server S2, this does not automatically mean that S2 is synchronized from S1. Synchronization in the opposite direction can be set up on a different schedule or disabled.
- Target server is Active. The target server (the one receiving data) drives the synchronization process. It connects to the source server, determines what updates need to be transferred, fetches the data and incorporates it into its repository.
- Synchronization is Direct only. Currently, the synchronization procedure can only take projects directly from their home server. Transferring projects through an intermediate server chain (server S1'server S2'server S3) is not supported.
- Synchronization can be Partial. The administrators can specify that all the projects of the source server should be synchronized to the target server OR specify a subset of projects to synchronize.

7.2 Using Synchronization

As described earlier, the synchronization process is launched between two given servers: source (providing the information); and target (receiving the information). The target server is the driving server; it actively connects to the source server and pulls the necessary data. Synchronization can be bi-directional, but for each direction, a separate synchronization must be launched. Both directions are independent of each other and can be present or absent as necessary. Several synchronization processes can run in parallel on the same target server, pulling data from several different source servers.



The synchronization process takes the project history data (including the project and category names, all the commits, comments, and branches) from the source server and uploads it to the target server. This process is incremental, so only the changes from the last synchronization are transferred. Target server users can then access this data in the same manner as local projects of the target. These projects are listed among other projects, can be opened, or attached as used projects/libraries to other projects. The only limitation is that they cannot be edited and no further versions can be committed.

Projects can only be edited on the home server. Project permissions of the source server are not synchronized - the target server administrator specifies the different access rules for the remote projects. Note that specifying write permissions on a remote project has no effect.

A remote project can be removed from the target server; however, please note that it will reappear after the next synchronization unless it is also excluded from the synchronized project set of the source server. The rename operations are handled in the same way; it is possible to rename the remote project on the server, but the name will be re-synchronized to the name in the source during the next synchronization.

Synchronization can be a long-running process, but it is not intrusive. The end users of the source and target servers do not feel its effects (no need to stop or restart servers during or after synchronization) except possibly some slight performance degradation.

The results of the pending synchronization appear on the target server gradually during synchronization. It is possible to see a particular remote project having 100 versions, then 110 after some time and 117 after synchronization is complete. However, this should not cause problems for clients on the target; partial version data will never be offered. For example, when there are 116 versions of a remote project declared, but the 116th version is only half-way downloaded, the 116th version will not be exposed.

Related Pages:

- [Distinguishing between Local and Remote Projects](#)(see page 74)
- [Controlling the Synchronization Scope](#)(see page 74)
- [Controlling Synchronization Initiation](#)(see page 75)
- [Deleted and Inaccessible Projects](#)(see page 75)
- [Securing the Connection between Servers](#)(see page 75)

7.2.1 Distinguishing between Local and Remote Projects

Remote and local projects on the server are almost indistinguishable. To differentiate between them, the synchronization process can prepend the arbitrary prefix to the remote category names.

During synchronization, the administrator (optionally) provides the prefix string and synchronization process modifying the names of the categories coming from the remote server. When synchronizing from multiple source servers, each different source can have a different prefix.


For example, in the target server of a GB site, projects coming from the server of the American site can have the "USA_" prefix added to their category names, and projects of the German site can have "DE_" prefix. If there are projects under the category "Requirements" on the German site, these projects will be located under "DE_Requirements" on the GB site.

7.2.2 Controlling the Synchronization Scope

It is possible to configure which projects are synchronized, i.e., select all or only a particular subset of the source server projects to be synchronized to the target.

The target server specifies the user/password of the source server when the target server initiates the synchronization process. The synchronization process then takes all accessible projects from the source server. The source server administrator can control which projects are accessible by managing

user permissions on his server. If the specified synchronization user has permission to read all the projects on the source server, all projects will be synchronized during the process. If the specified user only has permissions to access a particular subset of various projects, only these projects shall be synchronized during the process.

 It is highly advisable to create a special dedicated user on the source server for synchronization purposes and not reuse the real existing user accounts. Only read permissions should be given to this account.

Note that different synchronized project sets can be specified on the source for each different target. Just use different users for each target and define the project set as described above.

7.2.3 Controlling Synchronization Initiation

Not every user can trigger the synchronization procedure (on the target server).

The user must have at least the following system-wide permissions:

- Create project/category
- Edit project properties
- Rename category

It is not enough to have project-specific permissions.

7.2.4 Deleted and Inaccessible Projects

During synchronization, there may be situations where a project was previously located in the source server but is no longer available. The project may have been deleted on the source server or no longer accessible (by the source server administrator's decision).

The synchronization process does not and cannot automatically delete these unavailable remote projects from the target server, because other local projects may still be using this remote project. The missing remote project is not deleted from the target server but moved into a special category - ATTIC.

The target server administrator should inspect this category from time to time. If there are any projects, the target server administrator should inspect whether they are used in the local projects, perhaps using the Project Usage Map.

For more information, read [Project Usage Map](#)(see page 75) for removing unnecessary projects.

7.2.5 Securing the Connection between Servers

Since the connection between the source and target servers is usually going through the WAN, it is advisable to secure this connection, so that third parties cannot access the project data in transit.

There are two options to accomplish this:

1) Set up the SSL connection. In this case, the source server must have the public/private key pair configured in a keystore. The target server must have the public key (in certificate form) of each source server it wants for connection.

 For more information, see [Secured Connection tab](#)(see page 43).

2) Set up the VPN connection between the servers so that both servers are in the same virtual subnet. This solution is server-neutral. Nothing needs to be done on the server(s).

7.3 Running Synchronization

The synchronization procedure can be started in two different ways: 1) using a build in the scheduler in the Teamwork; and 2) using the command line utility. As stated earlier, the synchronization procedure must be triggered separately for each synchronization direction in bi-directional setup. The synchronization target server (the server receiving the data) is the active server, the one driving the synchronization process. When started, the synchronization process runs until completion or until an unrecoverable error appears (such as network problems) or until interrupted by the user (when using command line utility). You can monitor the synchronization progress and outcome in the server log.

Related Pages:

- [Triggering Synchronization with an Internal Scheduler](#)(see page 76)
- [Triggering Synchronization Using the Command Line Utility](#)(see page 77)

7.3.1 Triggering Synchronization with an Internal Scheduler

Teamwork Server has a built-in mechanism to trigger tasks at various periods and with various parameters.

This mechanism can be used for triggering synchronization between servers.

The tasks are specified by *.properties files located in the schedule subdirectory in the Teamwork Server install directory. Each *.properties file describes one task to be run at the defined period(s). There is an example file (synchronization.properties.example) which can be examined and copied/renamed to produce the synchronization task(s). For example, you can create two files (syncDE_GB.properties and syncDE_US.properties) to describe two synchronizations, one for synchronization from a remote server on the GB site and another for synchronization from the remote server on the US site.

Teamwork Server constantly monitors the schedule subdirectory for any changes in the *.properties files. Any changes take effect immediately. All other files with different endings, such as the synchronization.properties.example outlined above, are ignored.

When scheduled tasks are run, they are run on behalf of the system user of the Teamwork Server. This is the reason it is not necessary to specify the user on the target server when running synchronization using the scheduler; however, you must specify the user on the target server when running synchronization from the command line utility.

Properties files are specified in Java property file format.

Properties files carry the following task parameters:

- **com.nomagic.teamwork.synchronization.Task.** This parameter describes the type of task to be run. Currently, the synchronization task is the only type of task that can be scheduled. This task type is specified by the `com.nomagic.teamwork.synchronization.SynchronizationJob` value of the property.
- **com.nomagic.teamwork.schedule.cron.** This parameter specifies the frequency of task running. The parameter syntax corresponds to the Unix cron task definition syntax. This syntax allows very flexible descriptions of task running frequency and pattern. For example, the value: "0 30 01 ? * MON-FRI" triggers a synchronization task at 01:30 AM Monday to Friday. For more information about this parameter, please see the documentation of Quartz framework at <http://quartz-scheduler.org/files/documentation/Quartz-2.1.x-Documentation.pdf>. See the chapter "CronTrigger Tutorial." Alternatively, you can read the manual of your Unix system.
- **com.nomagic.teamwork.synchronization.SourceServer.** This parameter specifies the synchronization source server and port. It is a mandatory parameter for a synchronization task.
- **com.nomagic.teamwork.synchronization.SourceUser.** This parameter specifies the user on the synchronization source server (the server providing the data) used to access and fetch the project data.
- **com.nomagic.teamwork.synchronization.SourcePassword.** This parameter describes the password of the user on the synchronization source server used to access and fetch the project data. Since the password is stored in plain text, please ensure that the *.properties file has the correct and safe permissions.
- **com.nomagic.teamwork.synchronization.Use Secure Connection.** This parameter specifies that SSL connection shall be used between the target server and source server for data transfer. SSL certificates must be set up correctly on both the source and target servers. For more information, see [Secured Connection tab](#)(see page 43).
- **com.nomagic.teamwork.synchronization.CategoryPrefix.** This parameter specifies the prefix prepended to each remote category name in the target (this) server. For example, if the category name in the source server is "Requirements," and the prefix is "REMOTE/," in the target server this category will be named "REMOTE/Requirements."

Once triggered by the scheduler, synchronization runs to completion (or to unrecoverable error). The progress of synchronization can be monitored in the server log file. There is no facility to stop these jobs by user command (except for full server stop). If the synchronization job takes longer than the schedule period, and the next scheduled event happens before the previous job has finished, the event is suppressed. The second synchronization is NOT started in parallel; the running synchronization is allowed to continue and finish normally.

7.3.2 Triggering Synchronization Using the Command Line Utility

If running the synchronization through the scheduler mechanism is not flexible enough (perhaps you want to use your OS for scheduling tasks or run synchronization manually, on demand), there is a separate command line utility for triggering the synchronization process.

The utility is located in the bin subdirectory of the Teamwork Server installation directory. The utility is called `synchronize.exe` for Windows deployments and `synchronize` for Unix deployments. The utility has an accompanying `synchronize.properties` file.

This utility can be run from any computer (not just the servers), but most frequently it will be run on the target server. When the utility is run, it connects to the target server and initiates the synchronization procedure on it. There is no additional load on the computer running the utility; the procedure is run on the server side.

The utility outputs the synchronization progress messages.

A utility exists when the synchronization process is complete. It provides the exit codes (0 if synchronization is successful and >0 if errors are found). The exit code can then be used for shell scripting purposes (such as restarting synchronization in case of failure, etc.).

Please note that since the synchronization procedure is running on the server, exiting (or crashing or killing), the utility will not stop the synchronization process on the server. However, the utility does accept the Ctrl+C keyboard signal and stops the synchronization process on the server cleanly.

The utility takes 6 or 7 parameters. The last one, prefix, is optional. Parameters are mostly the same as the scheduled job, but there are additional parameters for specifying the target server to connect to (since the connection layout is now utility =>target server =>source server).

Here is a short synopsis:

```
synchronize (-h|--help)
```

A help message.

```
synchronize [options] <targetserver>[:<targetport>] <targetuser> <targetpass>  
<sourceserver> [:<sourceport>] <sourceuser> <sourcepass>  
[<prefix>] <targetserver>, <targetport>, <targetuser>, <targetpass>
```

Server and user credentials for the synchronization target(usually local) server. This utility will connect to this server and initiate the synchronization procedure. The target server will then start the synchronization procedure and pull the necessary data from the source server.

```
<sourceserver>, <sourceport>, <sourceuser>, <sourcepass>
```

Server and user credentials for the synchronization source (usually remote) server. The target server will connect to this server using the supplied credentials and pull the necessary data.

```
<prefix>
```

Options:

```
-h, --help
```

This help message.

```
-q, --quiet
```

Quiet synchronization. No information is displayed on the system output stream during synchronization. Errors are still reported on the system error stream.

```
--ssl
```

Implies --targetssl and --sourcessl.

```
--targetssl
```

Use SSL connection between utility and target server.

```
--sourcessl
```

Use SSL connection between target server and source.

Example

```
synchronize tws.company.de sysop p@Ssw0rd tws.company.ca:1102 sync ssApcnys CA_
```

This command connects to the tws.company.de server with the sysop user p@Ssw0rd password and initiates the synchronization procedure, pulling the information from the tws.company.ca:1102 server:port, using the sync user ssApcnys password. Categories from the .ca server will get the CA_ prefix on the .de server (e.g. the category "Requirements" from the ca server will be named "CA_Requirements" on the .de server).

If necessary, you can make copies of the utility with the pre-defined parameters. To do that, you can copy the **synchronize.exe** and **synchronize.properties** to files with different names and modify the **APP_ARGS** parameter inside the copied *.properties file - specifying the necessary arguments.

For example, you can copy synchronize.exe to synchronizeDE_GB.exe and copy synchronize.properties to synchronizeDE_GB.properties. Inside the synchronizeDE_GB.properties, you can specify DE and GB server parameters in the APP_ARGS line the same way you would specify parameters on the command line. After this you get a single executable (with all parameters pre-specified), that can be run with one click.

8 Transferring Projects between Isolated Servers

Isolated Teamwork Servers, which run in consciously disconnected or secured environments, can still interchange projects.

MagicDraw Teamwork Server 18.1 supports transferring project data from one Teamwork Server to another using any external storage device, such as a CD, DVD, hard disc, or flash memory device. The updated version of the shared project can be transferred back to the sharing server and smoothly merged with the original project version. Furthermore, the same project version can be given to several contributors simultaneously, and their contributions to the model can be successfully merged as well.

To manage the data transfer between isolated servers, utilize the functionality of the **Projects** tab in the Teamwork Administrator's Console.

Here is the workflow for interchanging projects between two isolated servers:

	Sharing Server	Contributing Server
1 .	Export selected projects A, B, and C to location X, a directory on the server. For more information, refer to To export projects from the current server (see page 37).	
2 .	Copy the exported data to an external storage device, such as a CD.	
3 .	Deliver the external storage device to the contributing server.	

	Sharing Server	Contributing Server
4 .		Copy data from the external storage device to location Y, a directory on the server.
5 .		Synchronize the server repository with data from location Y. For more information, refer to To synchronize projects on the current server with data from selected location (see page 37). After the synchronization finishes, the server gets 3 new read-only projects: A, B, and C.
<p>If you need to update the projects on the contributing server with new data from the sharing server, repeat steps 1 to 5.</p> <p>If you need to modify these projects on the contributing server and transfer the changes to the sharing server, perform steps 6 to 12.</p>		
6 .		Create a branch Contrib for the projects A, B, and C.
7 .		Modify projects A, B, and C on the branch Contrib.
8 .		Export projects A, B, and C to location Z on the server. Projects are exported with all their branches, including the branch Contrib.
9 .		Copy the exported data to an external storage device, such as a CD.
10 .		Deliver the external storage device back to the sharing server.
11 .	Copy the data from the external storage device to location W on the server.	

	Sharing Server	Contributing Server
1 2 .	<p>Synchronize the server repository with data from location W. For more information, refer to To synchronize projects on the current server with data from a selected location(see page 37). After the synchronization finishes, projects A, B, and C get a new read-only branch <i>Contrib</i> on the server. To review the transferred changes and incorporate them into the trunk, merge the projects from the new branch with the projects in the trunk. For more information, refer to Model Merge¹⁹. Another solution applicable in simple cases is setting the branch version as the latest version on the trunk. For more information, refer to Project Versions dialog²⁰.</p>	
<p>If you need to update the projects on the sharing server with new data from the contributing server, repeat steps 8 to 12.</p>		

¹⁹ <https://docs.nomagic.com/display/MD185/Model+Merge>

²⁰ https://docs.nomagic.com/download/attachments/17665106/project_versions_dialog.png?api=v2&modificationDate=1493363259636&version=1

Index