
# TEAMWORK SERVER

18.1

user guide

**No Magic, Inc.**

**2015**

# CONTENTS

# CONTENTS

# TEAMWORK SERVER

This UserGuide provides information about administrating MagicDraw Teamwork Server.


For information about working with Teamwork Server form the client side, that is MagicDraw, please refer to *MagicDraw UserManual.pdf*.

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 MagicDraw application is a client of Teamwork Server.

At www.http://www.nomagic.com/support/demos in the **Project management an collaboration** section, you will find the Teamwork viewlets to help you understand how to work with the Teamwork Server.


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

## Teamwork Server Concepts

Get acquainted with basic Teamwork concepts.

**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**

The default Administrator's account in Teamwork Server is:

Login name: Administrator

Password: Administrator

To prevent illegal access, it is advisable to change the Administrator's password.

For more information, see Section “Managing Teamwork Server” on page 22.

**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 shutdown 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 which enables visual grouping of projects in 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, ClearCase, or LDAP.

**Used Server Project**

A server project containing one or more shared packages. Used projects are created with a purpose to reuse them or to decompose projects into parts.

**Dependency between two elements**

A situation where one element (dependent element) refers to the data of the other element (independent element).

**Home server**

A server where project has been initially created.

**NEW! Domestic project**

A project created on the home server.

**NEW! Foreign project**

A project transferred from its home server after synchronization. The foreign project cannot be modified on the server, to which it was transferred, but it can be browsed, analyzed, selected for report generation, and used in other projects on this server. The foreign project can have domestic (editable) branches.

# Teamwork System Design

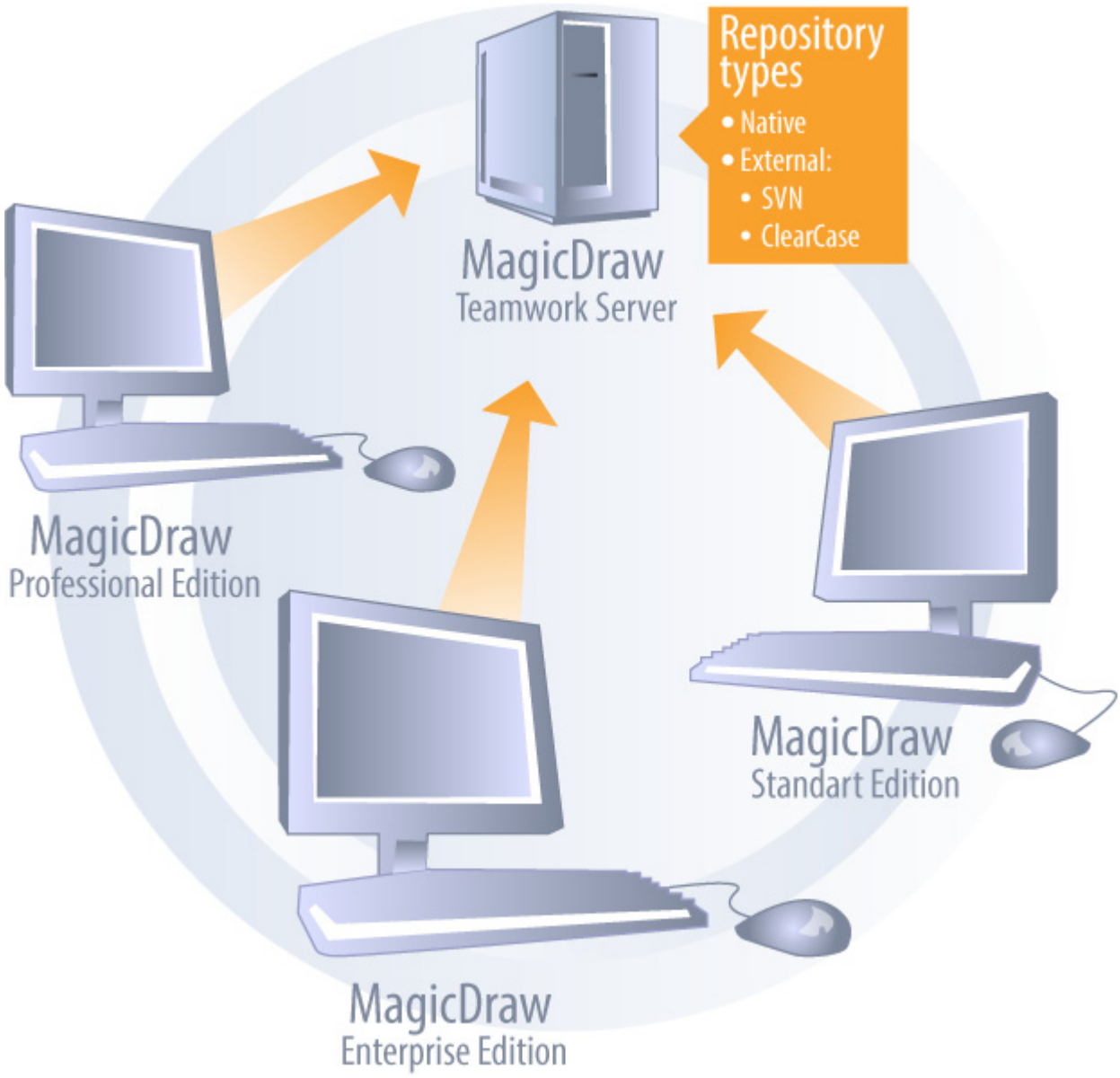



*Figure 1 -- Teamwork system design*

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

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

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 Section "Starting the Administrator's Console" on page 28). Data can migrate from one to another repository type. This functionality is also accessible from the Teamwork Administrator's console.

3 different types of repositories are supported in MagicDraw:

- Native (the default MagicDraw repository type).
- SVN (since v12.5).
- ClearCase (since v12.5).

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


For more information about specifying repositories, see the Administrator's Console dialog description, "Repository tab" on page 38.


For more information about importing or exporting project to the native repository, see the Administrator's Console dialog description, "Projects tab" on page 30.

## Native Repository

This is a default repository type. When the Teamwork Server is first installed and started, it is configured to use this type of repository. The Native repository is the only type of repository available in the version 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 into the server, it uses this user file to verify these users and their passwords. Users' rights to access different projects are also described using this file.

## SVN Repository

The Teamwork Server can be configured to use 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, that is 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, and 1.7.

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


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 not their passwords, their passwords are not stored. When a user tries to log on the Teamwork Server, server does not verify the password itself, but it logs on the SVN with the typed user name and password. All the project update/commit actions on the repository are performed by the server on the user's request. Hence, 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 that of the Native repository type, the users list with their crypted passwords is maintained in a repository file. The server authenticates users using this file. Actions in a repository are performed by the server 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 uses who performed them.


When a project file is committed into the SVN repository, the server stores additional 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 best performance, 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.

## ClearCase Repository

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

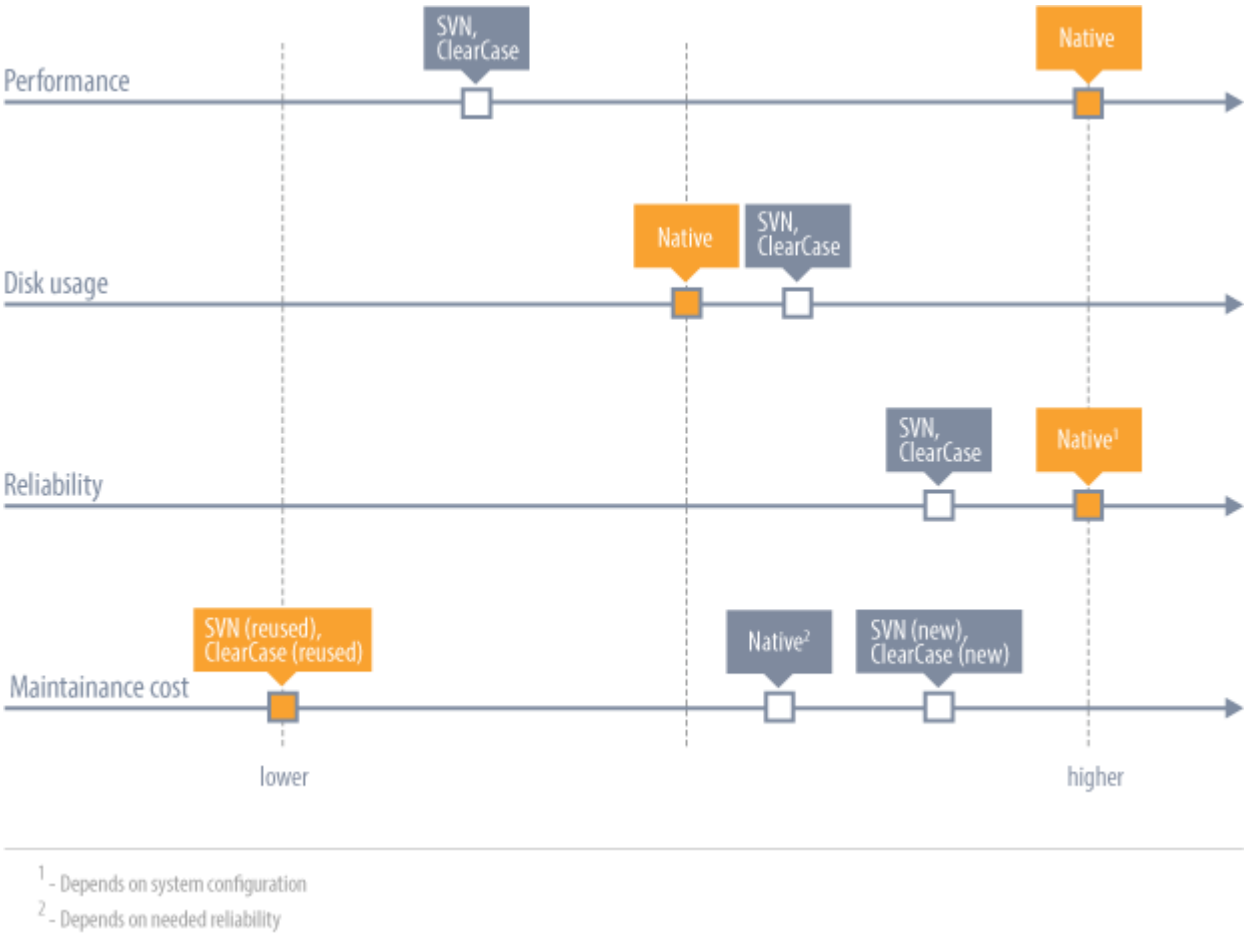
To use this repository type, the ClearCase client must be correctly installed on the computer where the Server runs. It must be able to launch cleartool executable - cleartool executable must be in PATH and have appropriate permissions. The supported ClearCase client versions are v7.0 or later (earlier versions might work, but were not tested).

All actions (update/commit) are performed in the ClearCase repository by the server on user request. If the server is started as NT service, actions in the repository will be attributed to the Local System user (unless a different user is specified in the 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 ClearCase repository with the ClearCase native tools. When looking at the project versions with MagicDraw client, all commit actions will be attributed to the users who performed them.


When a project file is committed into the ClearCase repository, the server stores additional auxiliary information about the project in an additional directory. 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 best performance the Teamwork Server should run on the machine where the repository is installed.

## Comparison of Teamwork Server Repositories




# Getting Started with Teamwork Server

## 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 when prompted. The key is no longer required the next time the server is run.


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

### 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.

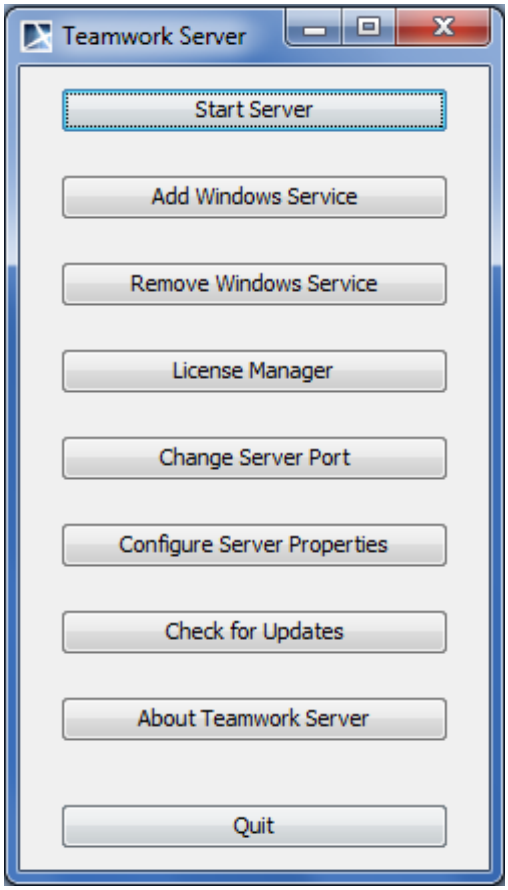



*Figure 2 -- Teamwork Server startup dialog*

2. Click the **Start Server** button.

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.
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.

NOTE

Restart the server to apply changes.

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.

## 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.

```
#!/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
```

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" on page 15).
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.

## Stopping Teamwork 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
```

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.


This feature is available only on Windows operating systems.

## Upgrading Teamwork Server

You can use automatic or manual upgrade of Teamwork Server.

In automatic upgrade, a new Teamwork Server version will be installed instead of current one. If you need to keep the current server version without any changes, please upgrade the server manually. The manual upgrade allows for installing a new server version on a new location. You will be able to import your old projects and users, to test a functionality of a new server version, and only after that to remove a previous server version.

Since the version of 16.9 commercial licenses are locked to the particular machine. You will be requested to activate the license and receive the commercial license dedicated for the particular machine after the upgrade. If you have already 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.

### Automatic upgrade of Teamwork Server

The easiest way to renew MagicDraw Teamwork Server is to upgrade it automatically. Using this feature the upgrade with newest versions and service packs is done automatically.


Make sure server and the client versions are the same. Also it is recommended to use the same JVM version for the server and client.


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

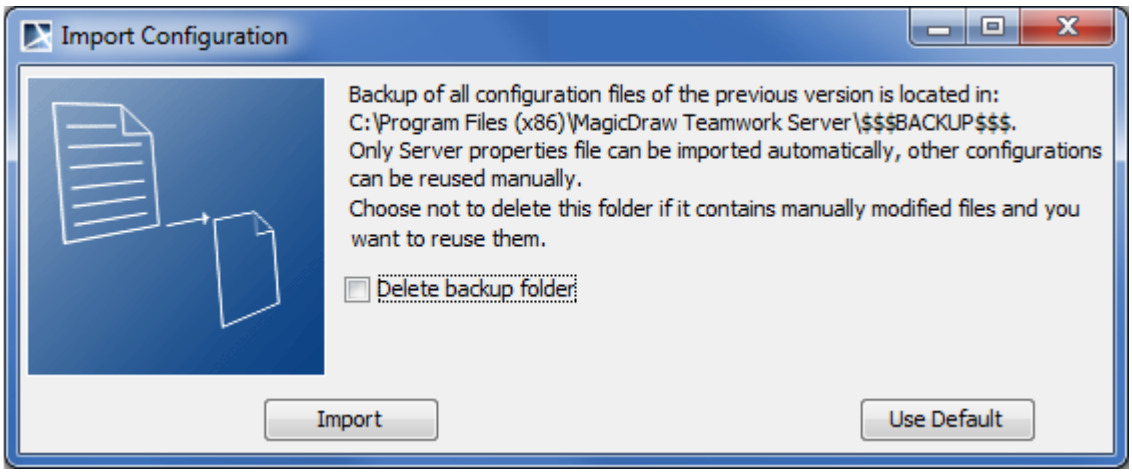
To upgrade Teamwork Server automatically

1. Stop Teamwork Server (see "Stopping Teamwork Server" on page 15) 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 "To remove Teamwork Server from the Windows services" on page 15. Skip this step for other operating systems.
4. Start the Teamwork Server GUI version (see "Starting the server using GUI" on page 10).
5. When the dialog opens, click the Check for Updates button to check and download program upgrades and updates.

NOTE

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

6. The **Update Information** dialog opens wherein information about available updates is displayed. Click the **Update to New Version** button to start upgrading the server.
7. 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.




IMPORTANT!

We strongly recommend to store the backup folder.

8. After upgrading the server to version 16.9 or later, you will be requested to apply the commercial license dedicated for the particular machine. Please visit the https://www.nomagic.com/support/installation-and-use/teamwork-server-install.html#activating for instructions on how to get the Teamwork Server license.

IMPORTANT!

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

## Manual upgrade of Teamwork Server

Installation of a newer version does not detect nor remove the current server version. All projects and users can be imported to the new server.

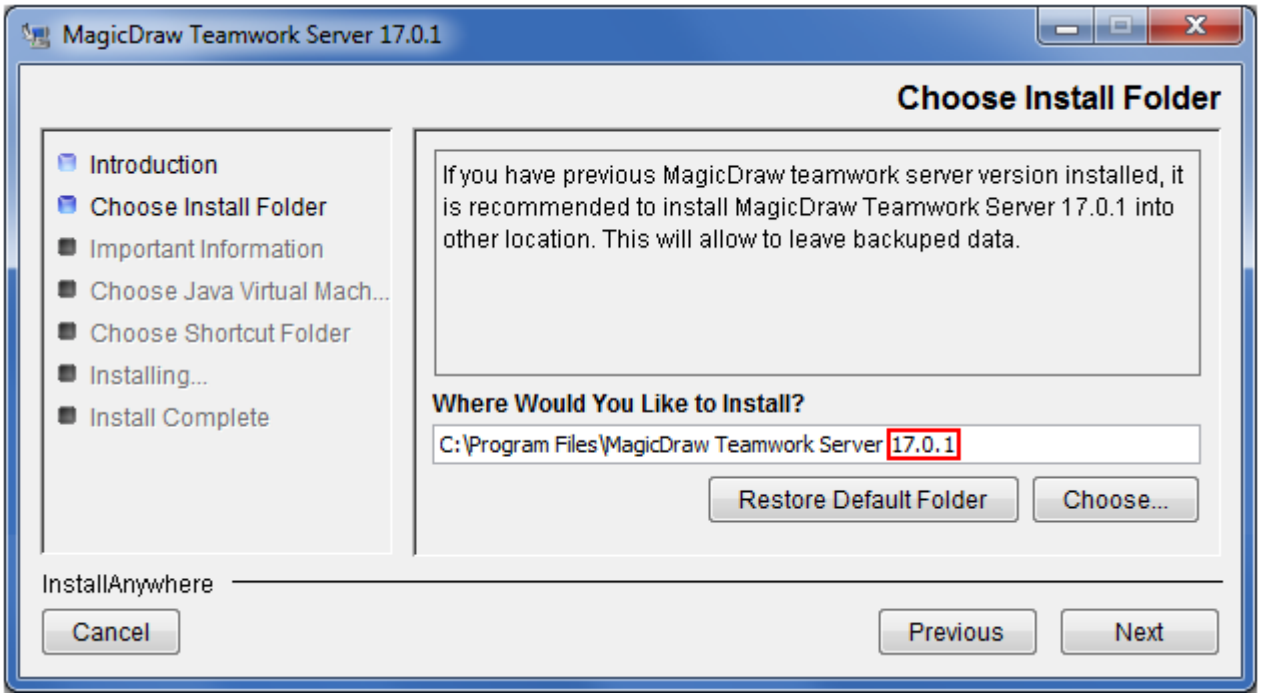
TIP!

For the description of importing procedures, see "Importing projects and users from earlier versions of Teamwork Server" on page 19.

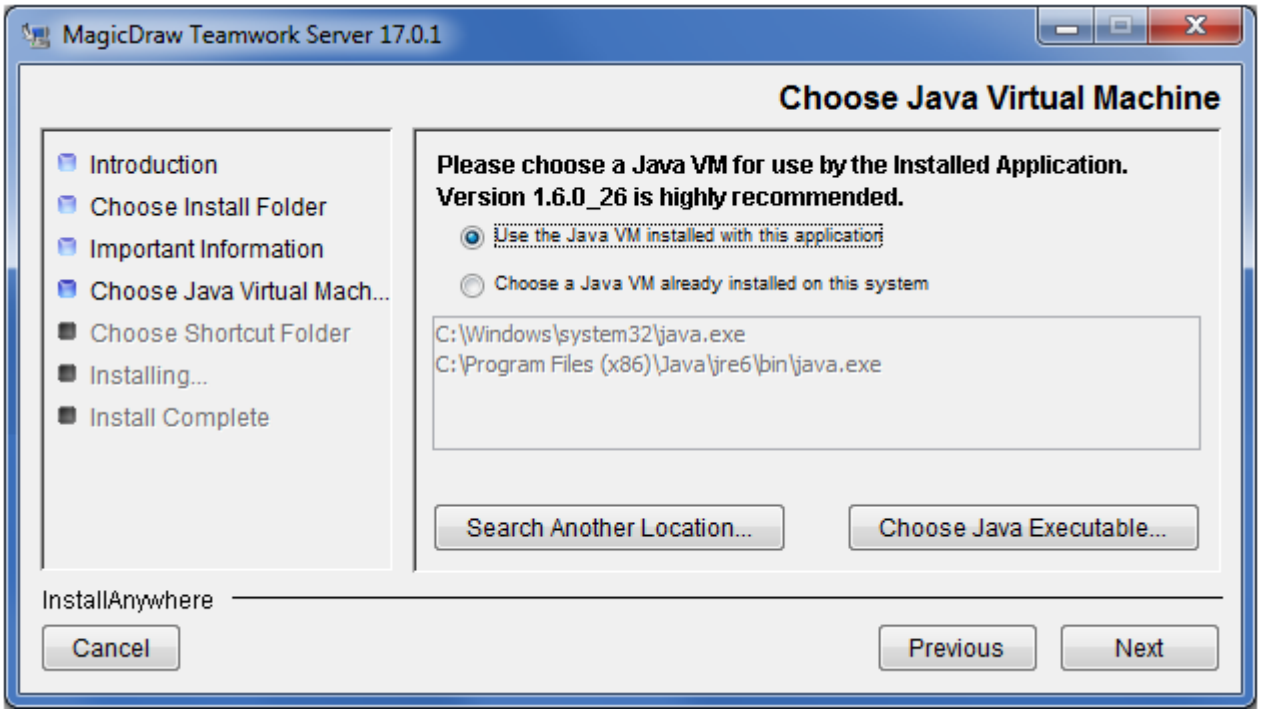
The version of project server and client should be the same and cannot be mixed up. Also it is recommended to use the same JVM version for the server and client.

To upgrade Teamwork Server manually

1. Stop Teamwork Server (see "Stopping Teamwork Server" on page 15).
2. Deactivate the current license for the Teamwork Server.
3. Remove Teamwork Server from Windows services, if it was added. See the procedure "To remove Teamwork Server from the Windows services" on page 15.
4. Run the installation file. For the successful installation please use the following recommendations:
    - To allow for restoring backup data, install a new Teamwork Server version into the other location than current Teamwork Server is installed. Under **Choose Install Folder**, add, for example, the version you are installing number to the end of the folder name.



- 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.

NOTE

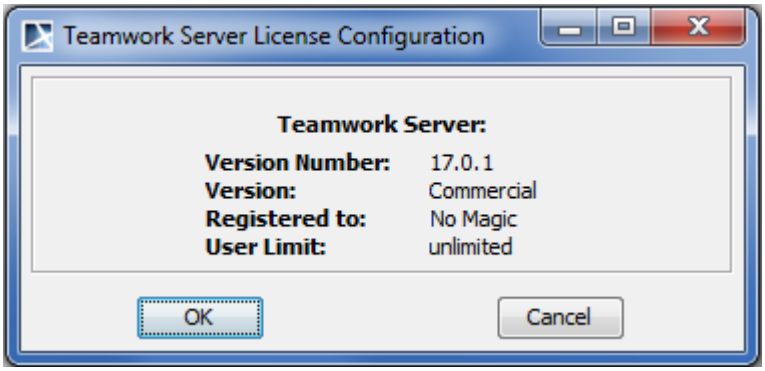
The import time can take more than 1 hour 30 minutes. Importing time depends on quantity and size of projects you are importing.

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

TIP!

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

7. 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.

8. Click the **Start Server** button. The server starts.
9. If it is needed, add Teamwork Server to Window services. See the procedure "To add Teamwork Server to Windows services" on page 12.
10. Remove the old Teamwork Server version.

IMPORTANT!

Make sure server and client versions are the same. Also it is recommended to use 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.

1. Stop Teamwork Server (see "Stopping Teamwork Server" on page 15.
2. Deactivate the current license for the Teamwork Server.


For more information, see http://www.nomagic.com/support/activation.html#deactivation_in_management

3. For the Windows operating system, remove the Teamwork Server NT service, if it was added (see "To remove Teamwork Server from the Windows services" on page 15). Skip this step for other operating systems.
4. Extract the MD_UML_<version number>_teamwork_server_no_installs.zip.
5. Start the new Teamwork Server without GUI (see "To start Teamwork Server from the command line" on page 12).


There is no automatic Teamwork Server upgrade without GUI, only manual upgrade.

## Importing projects and users from earlier versions of Teamwork Server

To import projects and users from earlier versions, use one of the following ways:

- Replacing the projects folder
- Changing the repository location in Administrator's Console
- Modifying the muserver.projects_directory property in muserver.properties file

### 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 location of the *projects* folder in the following table.

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

| Teamwork Server version | Location of *the projects* folder |
|---|---|
| **18.0 or later** | *C:\ProgramData\.magicdrawserver\projects* on **Windows 7/8**<br><br>*C:\Documents and Settings\All Users\ApplicationData\.magicdrawserver\projects* on **Windows 2000/XP**<br><br>*C:\WINNT\Profiles\All Users\ApplicationData\.magicdrawserver\projects* on **Windows NT4**<br><br>*<install.root>/projects* on **other** operating systems<br><br>Note that the *projects* folder is created automatically after starting Teamwork Server for the first time. |

## 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*

## Changing the repository location in Administrator's Console

By changing the repository location, you can indicate the *projects* folder that contains the project you want to import.

To change the repository location and import projects

1. In the Teamwork Administrator's Console window, **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, informing that In order to apply changes, you need to run repository test, click **Run Test**.
4. In the **Repository Test Passed** dialog, click **OK**.
5. When you receive the message, informing that changes have been saved and to apply changed properties, server should be restarted, click **OK**.
6. In the Teamwork Administrator's Console window, from **Menu**, select the **Restart Server** command.
7. In the **Information** dialog, informing that the server is restarting and try to login again after few minutes, click **OK**.

## Migrating from the SVN/ClearCase repository to the Native repository type

Migrating from the SVN/ClearCase repository to the Native repository type

1. Start the server on the SVN/ClearCase repository.
2. Open the **Administrator's Console**, trigger project export. Select a directory to dump the 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 a directory to work with.
4. Restart server in order to use this new repository.
5. Projects are now in a new Native repository.

## 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.

Let's say, we are moving Teamwork Server from the computer A to the computer B. In order to do that, follow the instructions below.

To move Teamwork Server

1. Stop Teamwork Server in the computer A (see "Stopping Teamwork Server" on page 15).
2. In a computer A, back up all data in case unsuccessful data transfer.
3. Install Teamwork Server in the computer B.
4. In the computer A, copy the *.magicdrawserver* folder and paste them in the same location in the computer B.


If your projects were stored other where than in the Project folder, copy that folder wherein Teamwork Server projects were stored and the path to that repository. In the 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 pats to a repository, see the procedure "To change a repository path" on page 21.

5. Start Teamwork Server (See "Managing Teamwork Server" on page 22).


Starting Teamwork Server for the first time, you will be required to enter a license key. For more information about license activation, see at http://www.nomagic.com/support/installation-and-use/teamwork-server-install.html#activating.

6. You can remove back up date in the computer A (optional).


Before removing back up data from the computer A, make sure that all data and server configurations are restored in the computer B after moving.

To change a repository path

1. In the **Teamwork Administrator's Console**, click the **Repository** tab.
2. Near to the **Repository Location** box, click the  button, to select the repository path.

3. Click **Apply Changes**.


For more information about repository, see "Teamwork System Design" on page 7.

# Managing Teamwork Server

You can manage Teamwork Server properties, teamwork projects, and users in the MagicDraw environment.

Section "Customizing Teamwork Server properties" on page 23 describes how to set general Teamwork Server properties.

Section "Changing Administrator Password" on page 24 describes how to change the administrator's password.

Section "Users Management" on page 24 describes how to manage users, user permissions, and how to assign projects to users.

Section "Starting the Administrator's Console" on page 28 describes how to start the Administrator's console.

Section "Administrator's Console Dialog" on page 28 describes the Administrator's console dialog tabs.

Section "Data Migration between Different Repositories" on page 46 describes how to migrate data between three different repositories.

Section "Changing Teamwork Server Debugging Mode" on page 47 describes how to change the Teamwork Server debugging mode.

## Customizing Teamwork Server properties

In the **Collaboration** pane of the **Environment Options** dialog, set the general Teamwork Server properties.

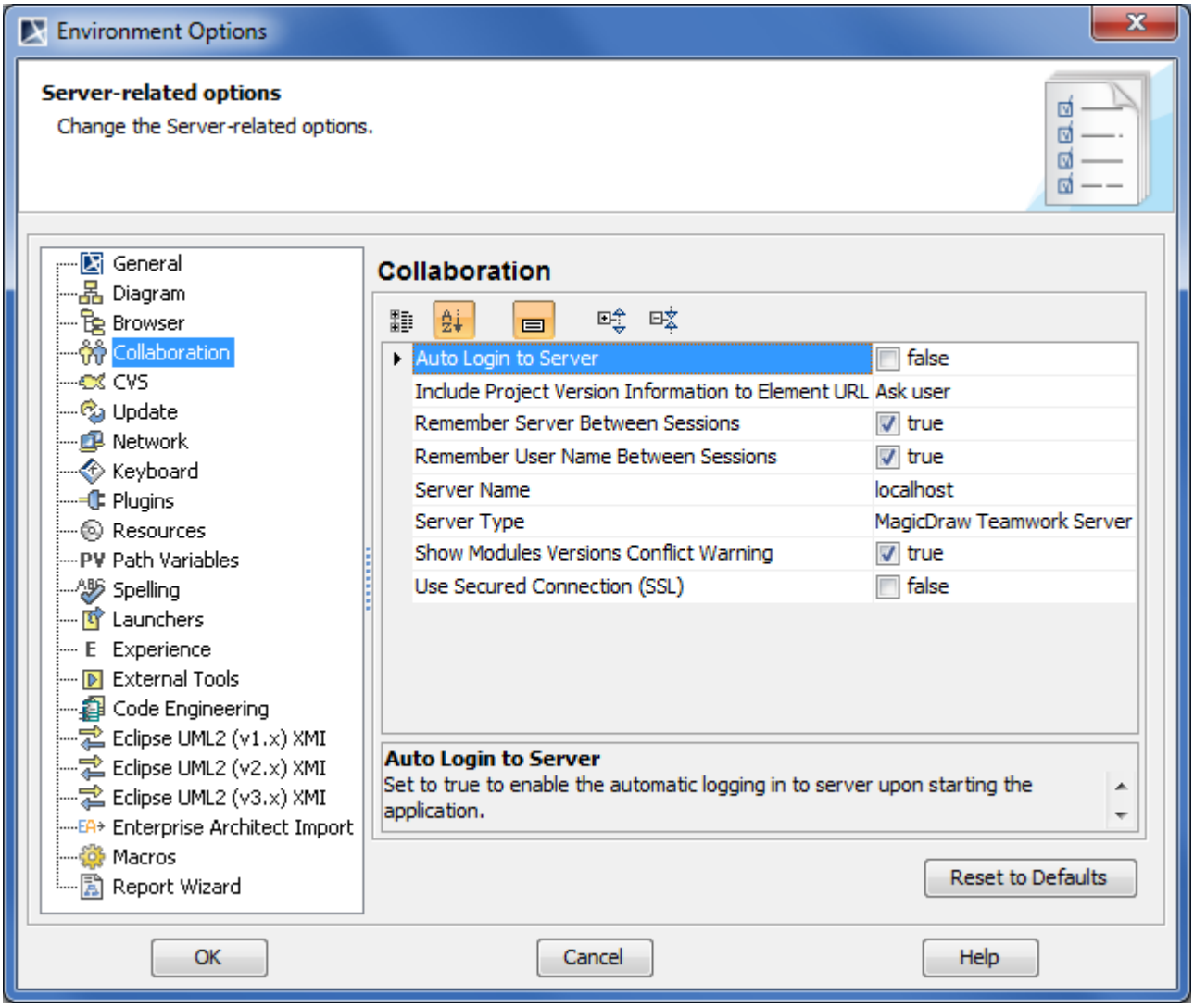



*Figure 3 -- Environment Option dialog. Collaboration options group*

For better understanding an option you can read its description that tells what is the effect of changing the option value.

To read the option description

**IMPORTANT!**

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.

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

**TIP!**

For more information about the **Environment Options** dialog, see "Customizing Environment Options" in the MagicDraw UserManual.pdf.

## Changing Administrator Password

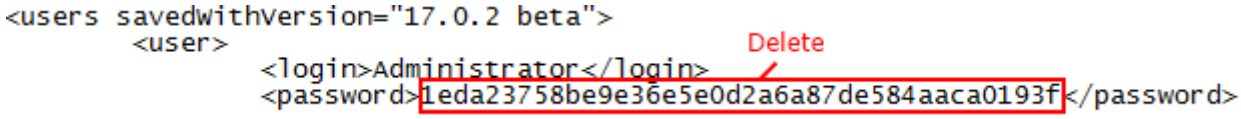
If you forgot the Administrator's password, see the following procedure on how to reset the Administrator's password.

To reset the Administrator's password

1. Stop Teamwork Server.
2. Open the folder which stores server projects. This location depends of your Teamwork Server version.


For more information, see "Replacing the projects folder" on page 19.

3. In the open folder, open the *users.xml* file for editing.
4. Delete all the letters and numbers that are between the *<password></password>* tags. See the following example:




5. Save the changes made in the *users.xml* file.
6. Restart the Teamwork Server.
7. Login to the Teamwork Server using the default Administrator's account:
**Login name**: Administrator

**Password**: Administrator


To prevent the illegal access, it is advisable to change the default Administrator's password.


For more information on how to change the user's password, see the "Working with Server Projects" section, in MagicDraw UserManual.pdf.

## Users Management

You can create two types of users in Teamwork Server: normal users that may have different permissions and users with administrator rights. Users with administrator rights can do the following actions in Teamwork Server:

- 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.

The Teamwork Server users have their own user accounts (including login names and passwords given by the administrator) and various types of permissions. According to the storage place of the user accounts, 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/ClearCase 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, ClearCase, LDAP) is enabled or disabled. The names of native and external users are unique per single server.

You can convert an external user to a native one and vice versa. Users with administrator rights 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.

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


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 Teamwork Administrator's Console or in MagicDraw UML when connection to Teamwork Server is established.

To add a new native user


Create a New User in Teamwork

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 (see "User permissions" on page 27).

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.


As you cannot set a password for an external user in MagicDraw's Teamwork Server, use an appropriate tool for managing the external database (Subversion, ClearCase, or LDAP) wherein 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


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

1. Start **Teamwork Administrator's Console** (see "Starting the Administrator's Console" on page 28).
2. In the **Active Users** tab, click the **Convert Native Users to External** button (see "Active Users tab" on page 29).
3. Click **Yes** to confirm your decision.

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” on page 34.

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.


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.

6. Click **OK**.

To convert all external users to native


- 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.

1. Start **Teamwork Administrator's Console** (see "Starting the Administrator's Console" on page 28).
2. On the **Active Users** tab, click the **Convert External Users to Native** button (see "Active Users tab" on page 29).
3. Click **Yes** to confirm your decision.

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 the information about the server log file, see "Log File tab" on page 34.

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. Click **More**, if you do not see the Teamwork projects list. The list of available Teamwork projects 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.

## User permissions

You can give several types of permissions to the 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 – user's administrative permissions to access and manage users and projects.
- Project access – user's permissions to work on specific projects.

| Permissions | Users can |
|---|---|
| **Edit model** | Modify a Teamwork project. |
| **Read model** | Open the content of any Teamwork project.<br>**NOTE:** If the **List not assigned projects** permission is not selected, you will be able to open the project using its URL. |
| **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. |
| **Administer 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 a category. |
| **Rename category** | Edit a category name. |
| **Create user** | Create a new user. |

| Permissions | Users can |
|---|---|
| **Remove project / category** | Remove a project or a 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 the 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. Click **More**, if you do not see the unassigned users list. 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 rights given.

## 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 shutdown or restart the server, change its properties, and view log files (including debug information) for 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.

## 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.

## Active Users tab

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

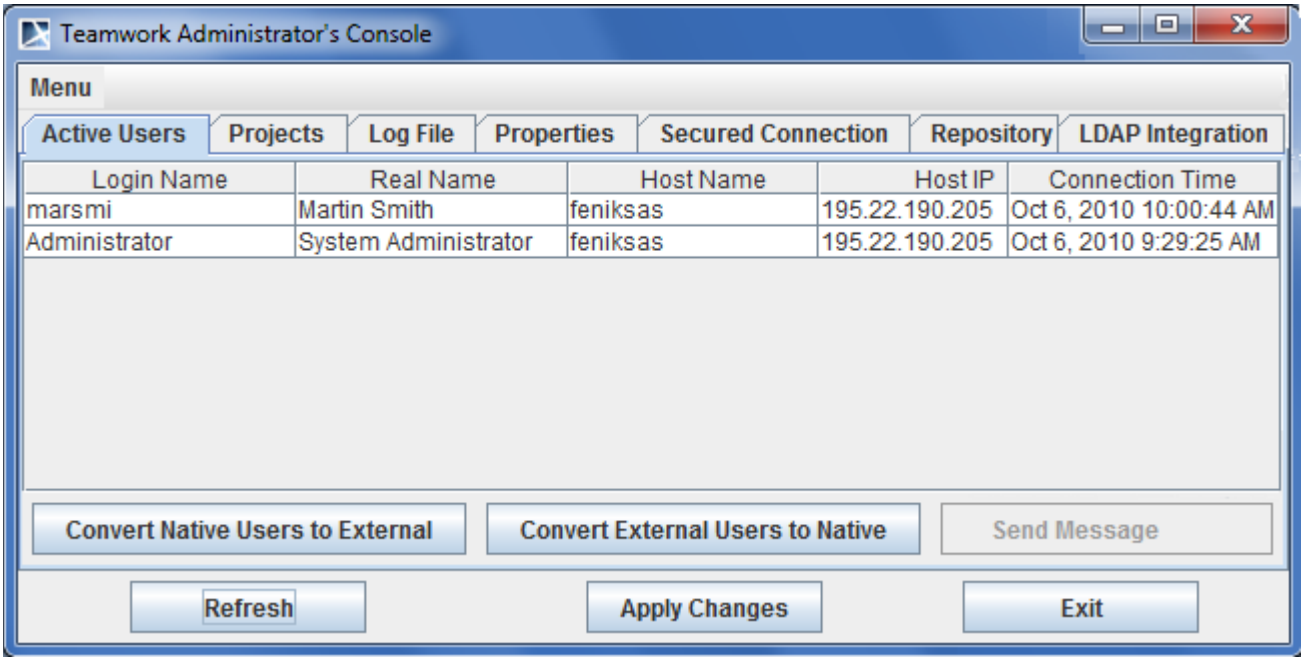



*Figure 4 -- Teamwork Administrator's Console. Active Users tab*

The GUI elements of the **Active Users** tab are described in the following table.

| Element name | Description |
|---|---|
| **Columns of the Active Users list** | |
| Login Name | The Teamwork user's login name. |
| Real Name | The user's full name. |
| Host Name | The host name. |
| Host IP | The host IP address. |
| Connection Time | The date and time the user connects to Teamwork Server. |
| **Buttons of the Active Users list** | |
| **Convert Native Users to External** | Starts the conversion of all connected native users to external users.<br>For more information about this conversion, see section “To convert all native users to external” on page 26. |
| **Convert External Users to Native** | Starts the conversion of all connected external users to native users.<br>For more information about this conversion, see section “To convert all external users to native” on page 26. |
| **Send Message** | Opens the **Send Message to Remote User** dialog for sending message to a specific user. The button is available, when a user is selected. |

## Projects tab

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

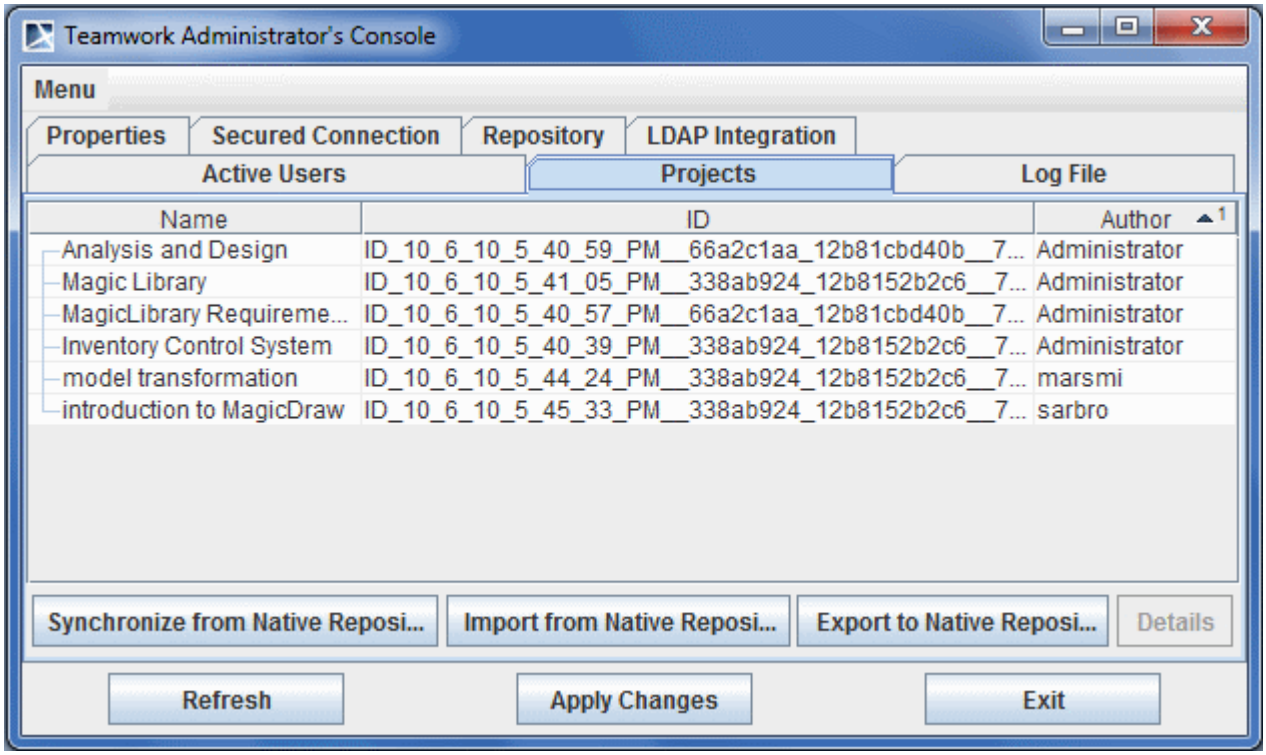



*Figure 5 -- Teamwork Administrator's Console. Projects tab*

Import from and export to native repository as well as synchronization can be triggered in the **Projects** tab.

**NEW!** To synchronize projects on the current server with data from 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, for example, a directory, to which you have exported the projects (see the procedure "To export projects from the current server" on page 32), or a directory on which the server operated in the past.

3. Click **Synchronize**.

The contents of the selected directory is imported into the current server. After the synchronization, the server assumes that the synchronized projects are foreign. Therefore, these projects cannot be modified, but the synchronization can be repeated to update them with new data.

To import projects into the current server

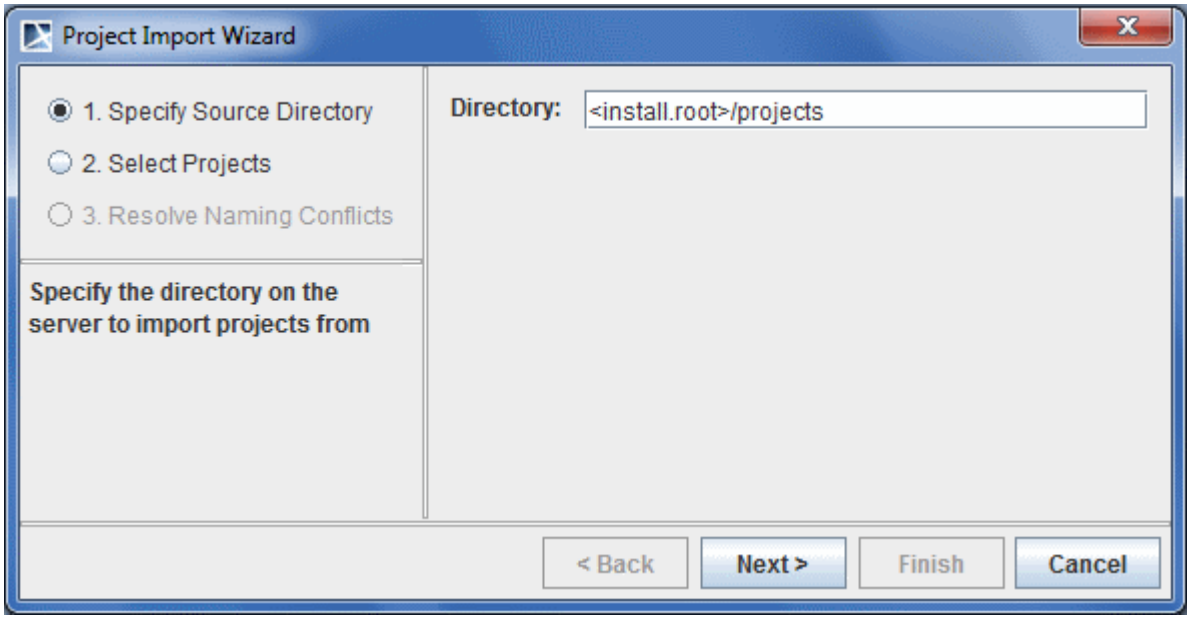
1. Click the **Import from Native Repository** button. The **Projects Import Wizard** opens.
2. Select the directory where the projects to import are stored.


This must be a directory that stores project versions in the Teamwork Server's Native repository format, for example, a directory, to which you have exported the projects (see the procedure "To export projects from the current server" on page 32), or a directory on which the server operated in the past.

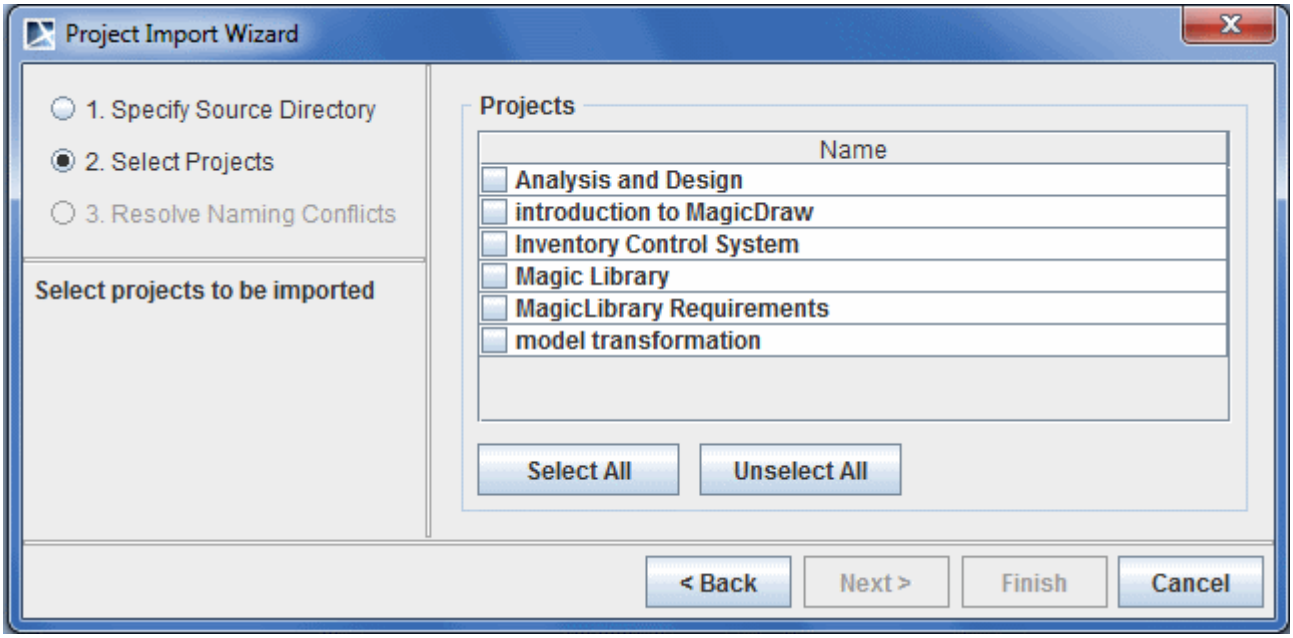
3. Click **Next**.

NOTE

To restore files from the backup that was made while upgrading the server, in the **Directory** box, type the path to the backup folder.

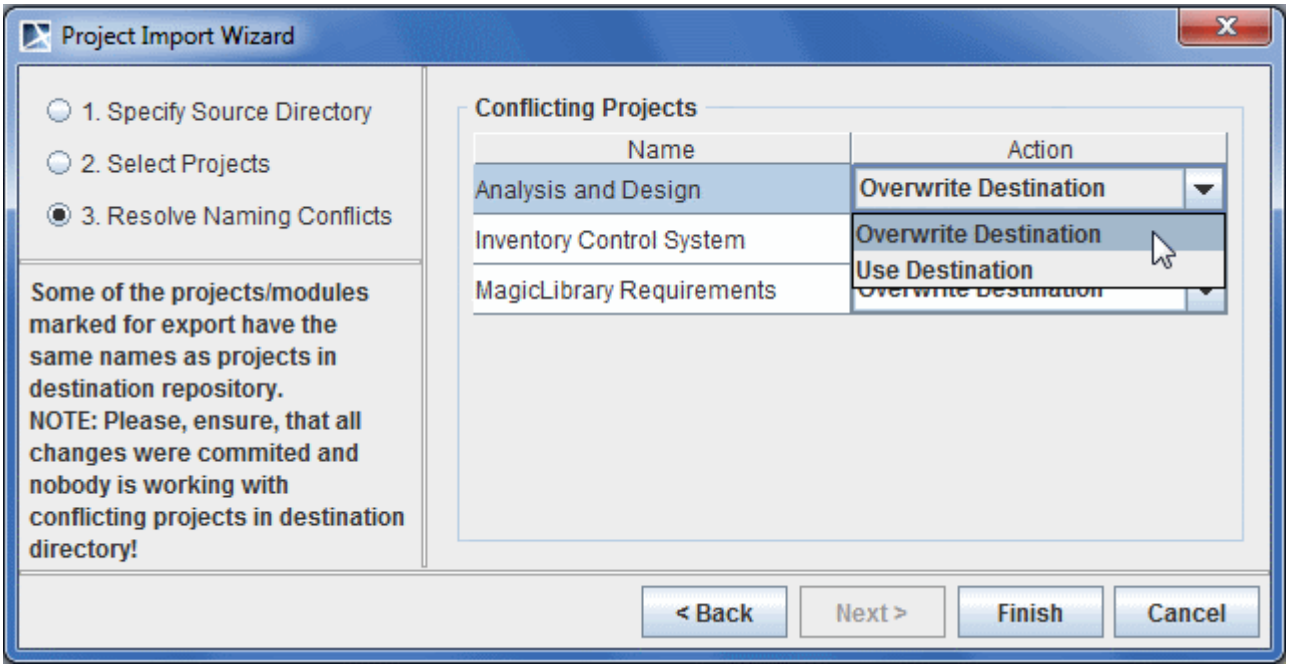



4. In the **Projects** list, all the projects that are stored in the directory are displayed. Select any projects to import from the list or click on the **Select All** button to import all projects. Click **Next**.




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 and all other projects/used projects, which are being imported, will be modified to use the already existing used project/profile in the current server.

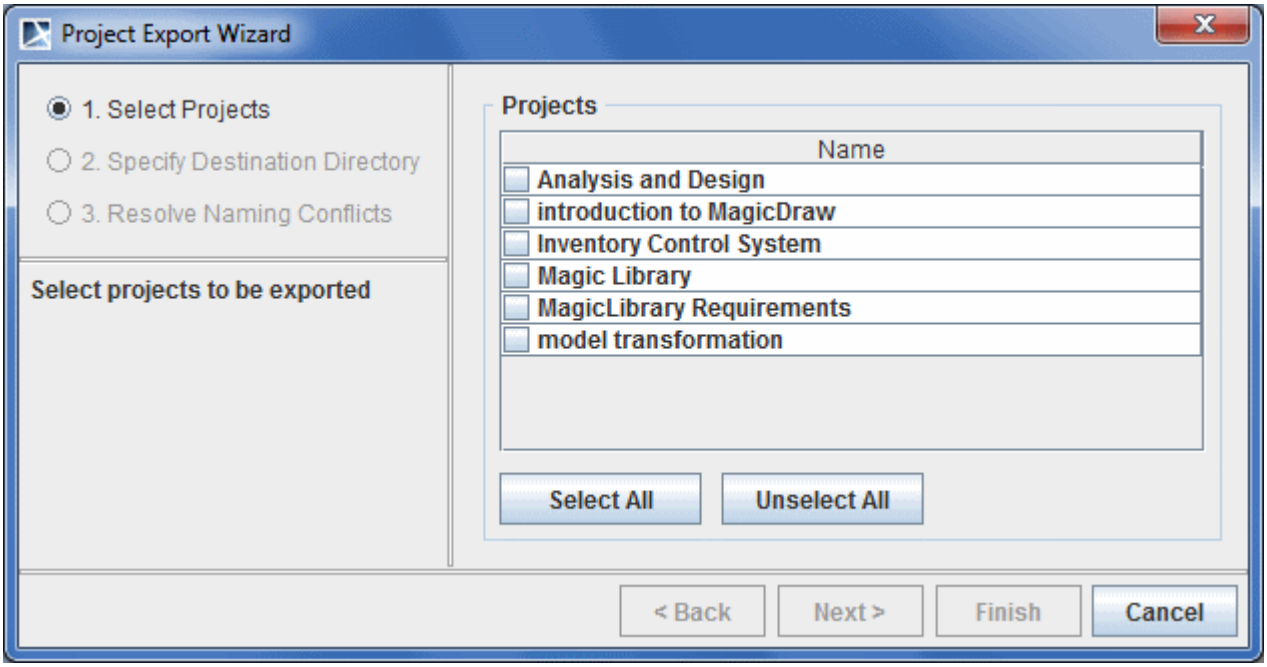
- **Overwrite destination**. Import the selected profiles or used projects and commit them by overwriting the existing ones in the current server.



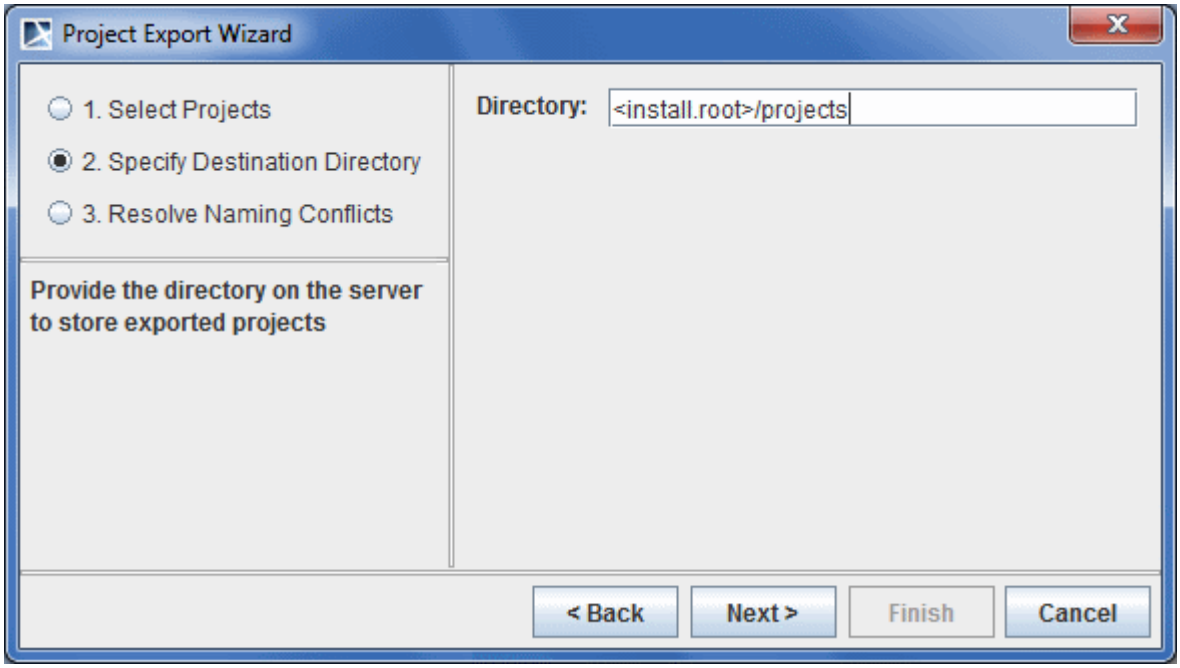

6. Click **Finish**.
The contents of the selected directory is imported into the current server. After the import, the current sever assumes the ownership of the imported projects. Therefore, they can be modified, but the import cannot be repeated.

To export projects from the current server

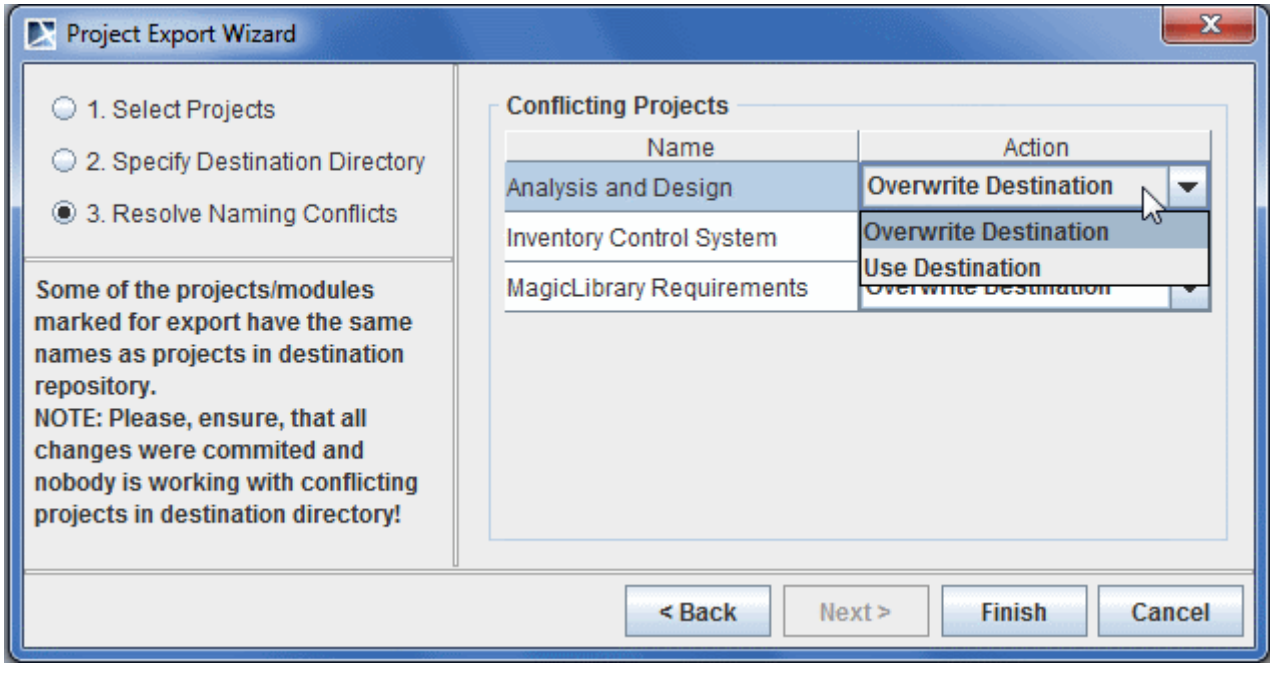
1. Click the **Export to Native Repository** button. The **Project Export Wizard** opens.
2. Select projects/used projects/profiles to export from the list, or click on the **Select All** button to import all projects/used projects/profiles. Click **Next**.



3. Select the directory to export the selected project. This can be either 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. The projects/used projects/profiles, to be export already exist in the selected destination directory (comparison is done by name), the naming conflicts will occur.
5. There are two actions to resolve the naming conflicts:
   - **Overwrite Destination**. Overwrite the projects/used projects/profiles that already exist in the destination directory with the exported projects/used projects/profiles.
   - **Use Destination**. Do not replace the projects/used projects/profiles that already exist in the destination directory.



To view the details of a project

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

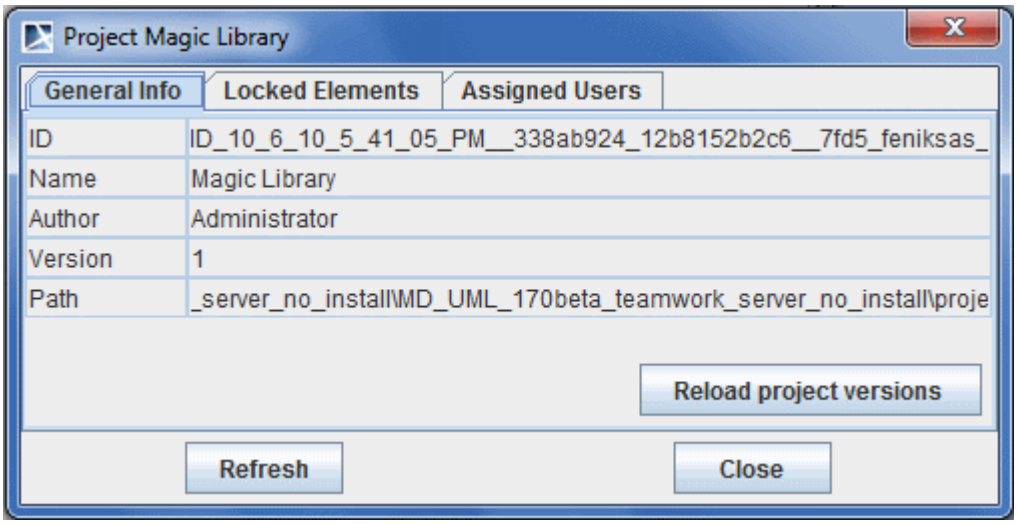



*Figure 6 -- Project dialog*

## Log File tab

If errors occur on the Teamwork Server, use the **Log File** tab to view the error message.

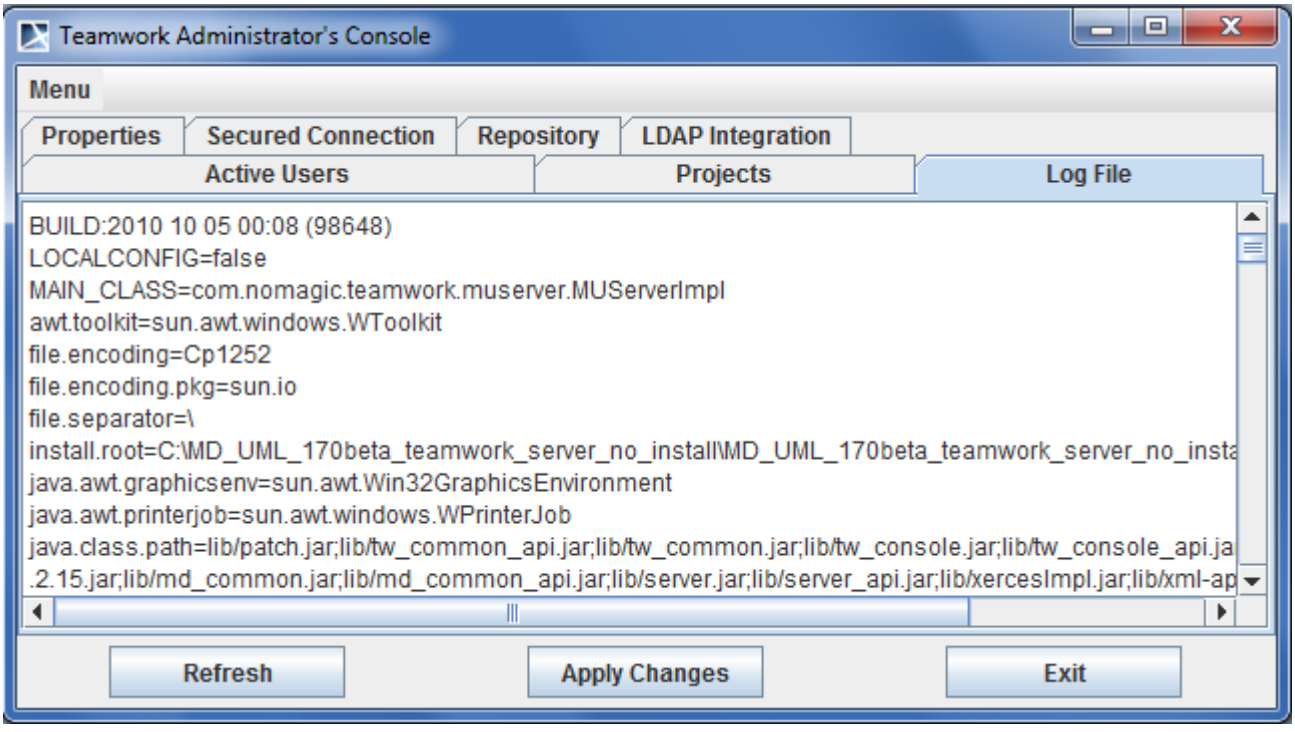



*Figure 7 -- Teamwork Administrator's Console. Log File tab*

## Properties tab

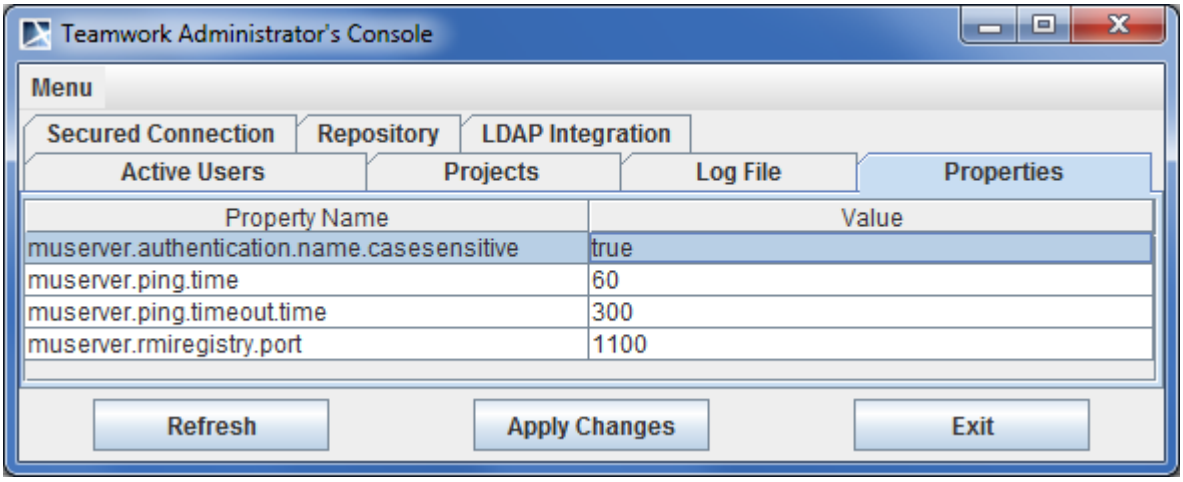



*Figure 8 -- Teamwork Administrator's Console. Properties tab*

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

- **muserver.authentication.name.casesensitive** specifies if the authentication is case sensitive or not. This means, that if the property is set to *true*, entered login name must fully match with registered one (for example, when the registered login name contains capital letters, user must enter the login name with capital letters).
- **muserver.ping.time** records how often the Teamwork Server pings a client. After sending a ping signal, 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 sent its ping. If no answer is received during this interval, the user is logged out from the Server.
- **muserver.rmiregistry.port** specifies the port on which communication between MagicDraw clients and Teamwork Server occurs (RMI traffic over TCP/IP).

NOTE

Restart the server to apply changes.

## Secured Connection tab

Since version 17.0, you can transfer data in a more secure way using the secured connection (SSL).

NOTE

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.

In order to use the SSL connection, two types of certificates are needed: one for the server and one for the client. Certificates must be in a Java Key Store format.

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

The client certificate should be located 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 applications MagicDraw and Teamwork Administrator's Console are Teamwork server clients. Both of it should have the client certificate. Hence the *certs* folder should be placed in two locations:

- <Teamwork Administrator's Console installation directory> for the Teamwork Administrator's Console application
- <LOCAL_APPDATA[1]>\.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 the 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 them by yourself.


For more information about generating certificates, see procedure "To generate certificates" on page 37.

To enable the secured connection (SSL)


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

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.


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

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

1. To find out the location of MagicDraw configuration files you may open **Help > About MagicDraw** and select the **Environment** tab in the MagicDraw application. Then click the **Configuration Files** link.

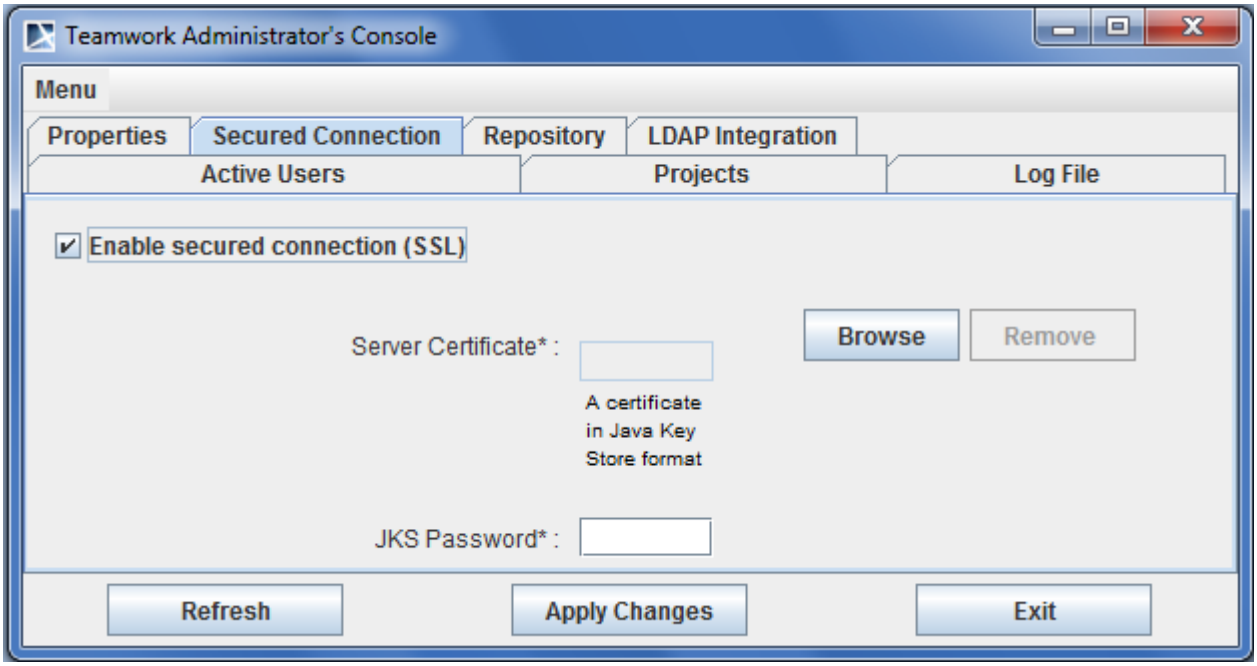



*Figure 9 -- Teamwork Administrator's Console. Secured Connection tab*

To generate certificates

TIP! We recommend you to use 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:
   - 2.1. Select **Create** > **Keystore**.
   - 2.2. Create an empty keystore file for the server. Do the following:
     - 2.2.1 In the **Keystore file** dialog, set the location of the file and type a file name.
     - 2.2.2 In the **Keystore password** dialog, type the password for the server keystore file and click **OK**.
   - 2.3. Create an empty keystore file for the client. Do the following:
     - 2.3.1 In the **Keystore file** dialog, set the location of the file and type a file name.

       TIP! For easier certificate transfer in next steps create a new folder “certs” and save the file named *cert.jks* in it.

     - 2.3.2 In the **Keystore password** dialog, type the password for the client keystore file and click **OK**.
3. Create a RSA keypair for the server:
   - 3.1. Select **Create** > **Keystore’s entry** > **Private key, with vers. #3** > **RSA**.
   - 3.2. In the **Keystore file** dialog, the **Source** section, open the created server keystore file and type a password.
   - 3.3. Provide the required information in the **Target** section and click **OK**.
   - 3.4. 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**.
   - 3.5. You will see the created alias. Close the dialog.
4. Exclude a public key from the keypair to provide it to the client:
   - 4.1. Select **Export** > **Private’s key first certificate in chain** > **As simple certificate file**.

4.2. In the **Keystore file** dialog, the **Source** section, open the server keystore file and type its password.

4.3. Create a file, whereto 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**.

4.4. The dialog for selecting an alias will open. Select from the list the alias that has been created in step 3.4 and type its password. You will be able to see the created certificate.

5. Import a public key into the client certificate:

5.1. Select **Import** > **Keystore's entry** > **Trusted certificate** > **Regular certificate**.

5.2. In the **Source** section set the certificate file, which has been created in step 4, as a regular certificate file.

5.3. In the **Target** section set the client keystore file *client.jks* as a keystore file and click **OK**.

5.4. The dialog will open asking to enter a new alias name.Enter the alias name created in step 3.4 and click **OK**.

5.5. Some pop-up windows will open informing about the generation process. Close all of them after reviewing.

Generated certificates are ready to use now. Paste them into the right location. For the information on how to configure SSL in a server side please refer to procedure "To enable the secured connection (SSL)" on page 36.

## Repository tab

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


For more information about repositories, see "Teamwork System Design" on page 7.


Restart the Server to apply changes.

There are 3 different types of repository to choose in the Repository type combo box. General information about the types is described in the Section "Teamwork System Design" on page 7. This section, presents the configuration fields in detail.

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, that is the **Repository Location**.

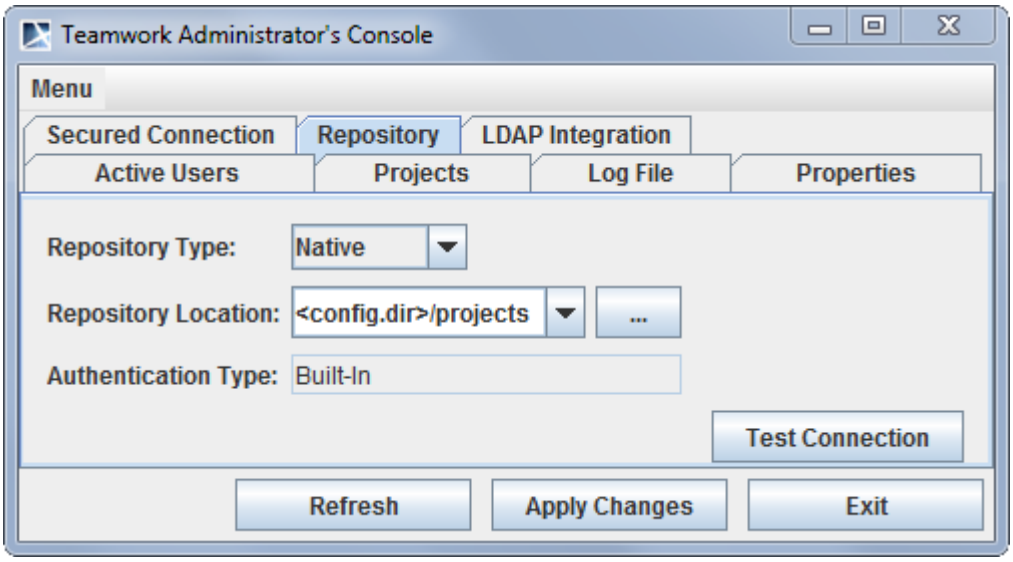



*Figure 10 -- Teamwork Administrator's Console. Repository tab for Native repository type*

The GUI elements of the **Repository** tab for the Native repository type are described in the following table.

| Element name | Description |
|---|---|
| **Fields** | |
| **Repository Location** | Designate the directory where the Teamwork Server stores information about: project list, users list, project versions, etc.<br>The designated directory depends on Operating System (OS) and is the following:<br>• On Windows 7/8 - C:\ProgramData\.magicdrawserver\<version number>\projects<br>• On Windows 2000/XP - C:\Documents and Settings\All Users\Application Data\.magicdrawserver\<version number>\projects<br>• On Windows NT4 - C:\WINNT\Profiles\All Users\Application Data\.magicdrawserver\<version number>\projects<br>• On other OS - <install.root>/projects<br>For more information on how to change the default configuration files location, see the “MagicDraw configuration files location” section in MagicDraw UserManual.pdf. |
| **Authentication Type** | The Native repository type can only use built-in authentication type. This field is not editable and for information purposes only. |
| **Buttons** | |

| Element name | Description |
|---|---|
| **Test Connection** | Test new parameters before applying them to the Teamwork Server. If the server cannot start with these new parameters, you will not be able to connect it with the Administrators console and change the parameters back to the old values. You will need to manually edit the muserver.properties file on the computer where Teamwork Server is installed.<br>The Native repository type is simple because it requires at least a repository location directory that can be created and edited.<br>**NOTE**: The *muserver.properties* file location depends on OS and is the following:<br>• On Windows 7/8 - C:\ProgramData\.magicdrawserver\<version number>\data<br>• On Windows 2000/XP - C:\Documents and Settings\All Users\Application Data\.magicdrawserver\<version number>\data<br>• On Windows NT4 - C:\WINNT\Profiles\All Users\Application Data\.magicdrawserver\<version number>\data<br>• On other OS - <install.root>/data |

**SVN repository**

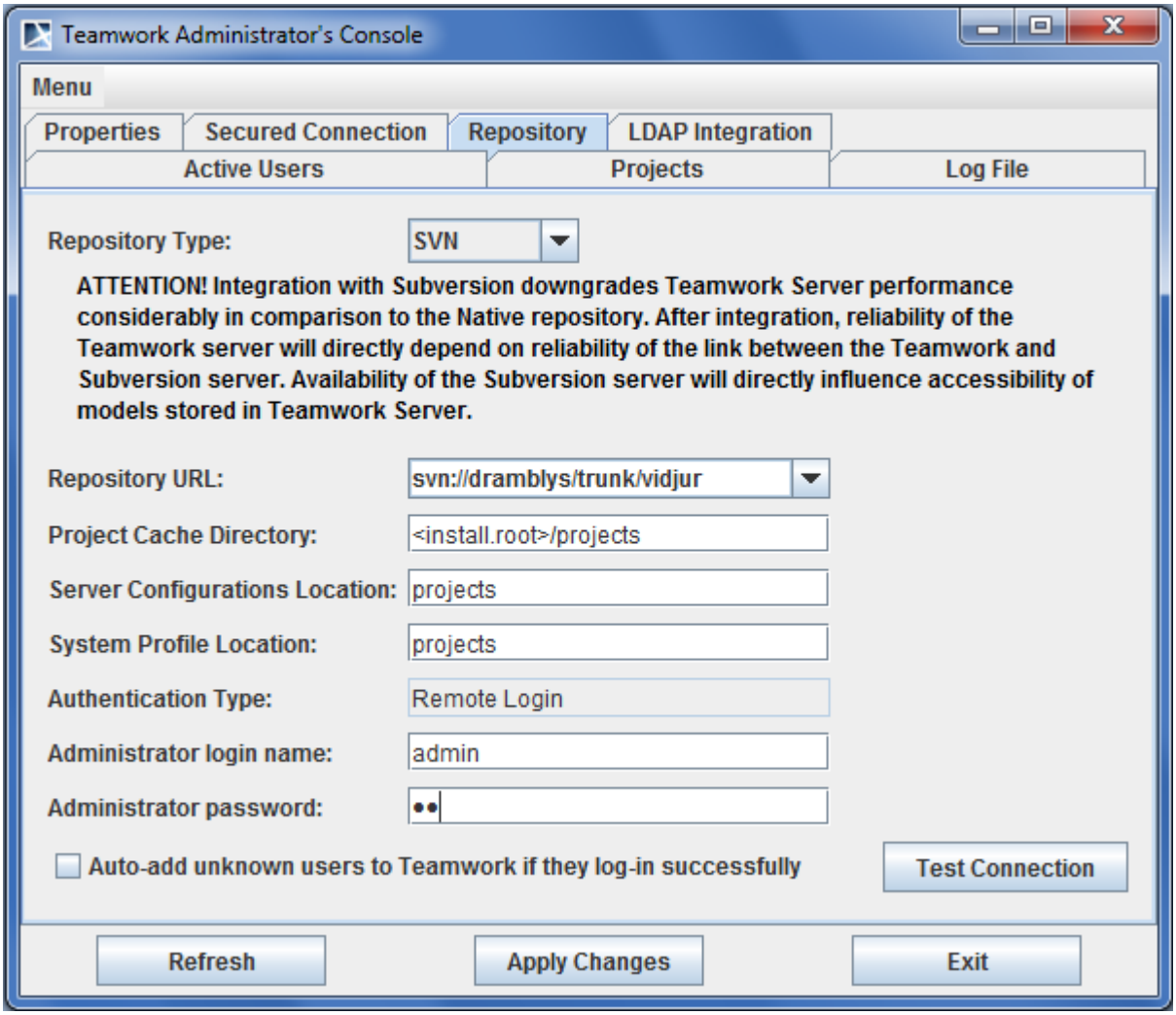



*Figure 11 -- Teamwork Administrator's Console. Repository tab for SVN repository type*

When configuring Teamwork Server to use the SVN repository, the SVN client executable must be installed on the computer where Teamwork Server runs. 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 different computer than the Teamwork Server runs.

| Element name | Function |
|---|---|
| **Fields** | |
| **Repository URL** | An URL field, which points to the SVN repository that will be used. The path can direct to a repository root or any other inner repository folders. All other paths (configurations location, profile location, also project paths, when adding project to the repository) will relate to this location.<br>All repository access types, which are supported by standard SVN installation, are supported here. In particular, file:// (local repositories), svn:// and svn+ssh:// (remote repositories accessible through remote login and svnserve), http:// and https:// (remote repositories, accessible through HTTP and WebDAV) URLs can be used here. |
| **Project Cache Directory** | The directory where the Teamwork Server checksout files to work with the repository. The Teamwork Server uses this directory to launch all necessary SVN operations to implement project storage and create project version. This is a “scratchpad” for the Teamwork server to work. |
| **Server Configuration Location** | The directory in SVN repository where the Teamwork Server keeps its configuration files (project and user lists). |
| **System Profile Location** | The directory in SVN repository where the MagicDraw's system profiles are stored. |
| **Authentication Type** | Currently not editable. This is an informational-only box indicating what authentication type will be used. The type depends on the repository access method (repository URL). If the repository URL is:<br>• svn://, svn+ssh://. The Authentication Type will be **Remote Login**. This means that when performing SVN actions, the Teamwork Server will log in remotely, as the user, who performs the action. A pass-through authentication of MagicDraw clients will be used.<br>• http://, https://. The Authentication Type will be **Basic/Digest HTTP**. This means that when performing SVN actions, the Teamwork Server will authenticate with the web server using HTTP Basic Auth or HTTP Digest Auth (see RFC2617), as the user who performs the action. A pass-through authentication of MagicDraw clients will be used.<br>• file://. The Authentication Type will be **Built-In**. A pass-through authentication is impossible for this type of repository access, hence the Teamwork Server refers back to the same authentication type as that of Native repository. The Server will keep its own users list with their passwords, and authenticate users itself.<br>The pass through authentication means that the Teamwork Server will only keep users list, but not their passwords. When a MagicDraw client logs into the Teamwork Server, the server will not verify the password, but will log into SVN instead. It stores the user's password in memory and use it for all subsequent communications with the SVN prompted by the user's actions. This feature decreases the burden on both the administrator and users because all passwords are kept in one place.<br>Certificate based authentication method for svn+ssh:// and https:// is not yet supported. |
| For pass-through authentication methods (all types of repository URLs except file://) the following fields are available: | |
| **Administrator login name** | Specify the administrator of the Teamwork Server. This user will have full permissions to the projects and various administrative tasks, such as updating user list file, project list file, etc. in the SVN repository. |

| Element name | Function |
|---|---|
| **Administrator password** | The Administrator's password. The Teamwork Server needs this password to perform various administrative tasks in the SVN repository.<br>**NOTE:** this password will be stored and encrypted in a form that is difficult decrypt. This encrypted password is stored in the muserver.properties file. You can set restricted permissions on this file for example, accessible only to user in the Teamwork Server.<br>The *muserver.properties* file location depends on OS and is the following:<br>• On Windows 7/8 - C:\ProgramData\.magicdrawserver\<version number>\data<br>• On Windows 2000/XP - C:\Documents and Settings\All Users\Application Data\.magicdrawserver\<version number>\data<br>• On Windows NT4 - C:\WINNT\Profiles\All Users\Application Data\.magicdrawserver\<version number>\data<br>• On other OS - <install.root>/data |
| **Auto-add unknown users to Teamwork if they log in successfully** | Decreases the administrator's burden to manage users list. If this check box is cleared, all MagicDraw users who try to log in, but are not in the users list of the Teamwork Server will be refused an access if they supply correct usernames/ passwords to login the SVN repository. If the check box is checked, MagicDraw users who are not in the users list of the Teamwork Server, but supply correct usernames/passwords to login to the SVN repository will be automatically added to the list and access the server. |
| The default Built-In authentication method (file:// type URLs) login name is "Administrator" without a password field (not necessary). The "Auto-add unknown users to the Teamwork if they log in successfully" check box is also unavailable (meaningless) in this case. | |
| **Buttons** | |
| **Test Connection** | Test new parameters before applying them to the Teamwork Server. If the server cannot start with these new parameters, you will not be able to connect it with the **Administrator's Console** and change the parameters back to the old values. You will need to manually edit the muserver.properties file on the computer where the Teamwork Server is installed.<br><br>This test ensures that the Teamwork Server is able to access and write the cache directory, access (and log in with the specified administrator user and password) the SVN repository in the specified location. It also checks the configuration and system profile locations on the SVN repository (the administrator must be able to write there). |

### ClearCase repository

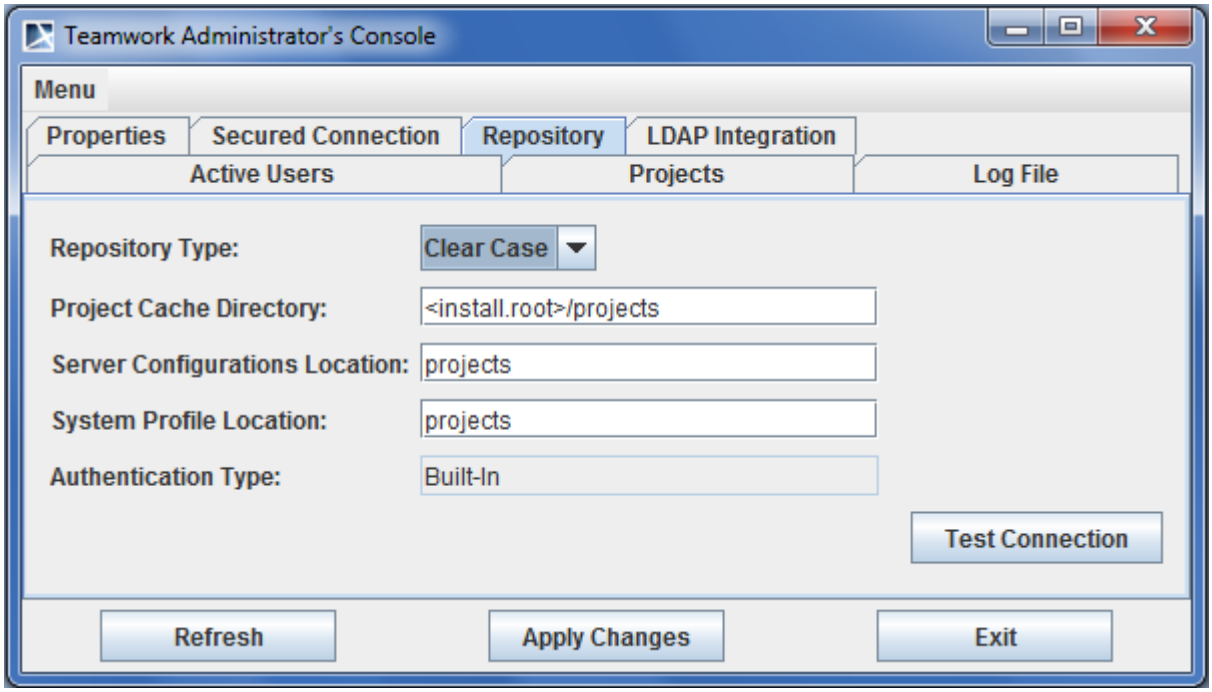



*Figure 12 -- Teamwork Administrator's Console. Repository tab for Clear Case repository type*

The ClearCase type of repository has a few parameters. When configuring this repository, the ClearCase client part must be installed on the computer where the Teamwork Server runs. The Teamwork Server will use cleartool executable to access and manage files in the ClearCase repository, so cleartool must be accessible - the PATH and have appropriate permissions to run. Also, the ClearCase repository must have the right permissions for the user, the Teamwork Server.

| Property name | Function |
|---|---|
| **Fields** | |
| **Project Cache Directory** | The directory where the Teamwork Server checksout files to work with the repository. The Teamwork Server will create a snapshot view (in ClearCase parlance) in this directory and place any necessary VOBs there project and perform all operations necessary to implement project storage and project versions. This is a “scratchpad” for the Teamwork Server to work. |
| **Server Configuration Location** | The directory in the ClearCase repository, where the Teamwork server keeps its configuration files (project and user lists).<br>**NOTE:** Must begin with some VOB name and cannot be left empty. |
| **System Profile Location** | A directory in the ClearCase repository where the MagicDraw's system profiles are stored.<br>**NOTE:** Must begin with some VOB name and it cannot be left empty. |

| Property name | Function |
|---|---|
| **Authentication Type** | The ClearCase repository type can only use built-in authentication type. This field is not editable and for information purposes only.<br>The Built-In authentication means that the Teamwork Server manages its own users list and their passwords. When MagicDraw clients connect to the server, it will authenticate them using this list.<br>When the Teamwork Server works with the ClearCase repository, it uses the standard system authority to identify itself to the repository. This means that the Server run as authorized by the user. All actions in the ClearCase repository will be attributed to that user (irrelevant to users in the connecting MagicDraw client). If a user starts the server, it will run as authorized by that user. If the server is started as NT service the service will run under a local system account. This can be a source of cryptic errors, for example if the local system account has no rights to access the ClearCase repository, the Teamwork Server will fail when started as a service, while everything works all right when started from the command line. |
| **Administrator Login Name** | When the Built-In authentication type is used, the administrators name is “Administrator”. This field is not editable and for informational purpose only. The default Administrator’s password is (for a new set-up repository) “Administrator”. For security reason it is highly recommended to change the default password after the server starts. |
| **Buttons** | |
| **Test** | Test new parameters before applying them to the Server. If the server cannot start with these new parameters, you will not be able to connect it with the Administrators console and change the parameters back to the old values. You will need to manually edit the muserver.properties file on the computer where Teamwork Server is installed.<br>This test ensures that Teamwork Server is able to create the ClearCase view in the specified directory and to access the two configured VOBs (for configuration storage and profile storage).<br>**NOTE:** The *muserver.properties* file location depends on OS and is the following:<br>• On Windows 7/8 - C:\ProgramData\.magicdrawserver\data<br>• On Windows 2000/XP - C:\Documents and Settings\All Users\Application Data\.magicdrawserver\data<br>• On Windows NT4 - C:\WINNT\Profiles\All Users\Application Data\.magicdrawserver\data<br>• On other OS - <install.root>/data |

## LDAP Integration tab


For more information about LDAP Integration, see "LDAP Support" on page 48.

# NEW! Administrating Server via Command Line Utility

Administrative tasks, such as users, projects, and permissions management, can also be performed by using a command line utility. The utility is stored in *<Teamwork Server installation directory>\bin* and 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 command 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, thus enabling conditional script execution.

To start performing the 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 start performing 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 the users or tests if the given user exists. |
| **mkproj** | Creates an empty project. |
| **rmproj** | Deletes the given project. |
| **lsproj** | Lists all the projects or tests if the given project exists. |
| **stperm** | Sets the specified permission of the specified user on a given project. |
| **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, which is 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.


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


If your Teamwork Server runs on the Linux or other Unix-based operating system, you can escape typing a long command every time by using once the following command:

**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*".

### Giving commands on the input stream

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

**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.

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
```

## Data Migration between Different Repositories

3 different types of repositories are supported in MagicDraw v12.5 and above. By default (when first installed) the server uses the Native repository type. When switching to a different repository type, the task of project migration is inevitably increased.

There are many different but similar use cases, such as migrating the data from the previous Native type repository into a new SVN/ClearCase based repository.

It is possible for the Teamwork Server to import and export projects. This function is triggered from the Administrators console (see section "Administrator's Console Dialog" on page 28). *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 words "import" and "export" here are used relatively to the currently running server (the one to which administrator's console is attached):

- "import" means importing data into the current server from the designated directory;
- "export" means exporting data from the current server into the designated directory.

Migrating the server from the Native repository to the SVN/ClearCase repository

1. The Server is started on the Native repository (projects/versions stored in a directory).

2. Open the Administrator's Console. Reconfigure the server to work with the SVN/ClearCase repository.
3. In the **Administrator's Console**, reconfigure the server to work with the ClearCase/SVN repository.
4. Restart server to use the new repository. The Server starts. There is nothing in the repository but the profiles needed to work with MagicDraw.
5. Login again to the **Administrator's Console** and trigger project import. Select the directory to import from, the same directory from where the export was performed.
6. Projects are now in a new SVN/ClearCase repository.

Migrating from SVN repository to ClearCase repository

1. Server is started on SVN/ClearCase repository.
2. Open Administrator's Console, trigger project export. Choose some temporary directory to dump the data to.
3. In **Administrator's Console**, reconfigure server for work with some ClearCase/SVN repository.
4. Restart the server, to use new repository. Server will be started, repository will be empty - no projects - only the profiles, needed for work with MagicDraw will be added to repository.
5. Login again in the **Administrator's Console**, and trigger project import. Choose the directory to import from - the same directory where you previously did export.
6. Projects are now in a new, ClearCase/SVN repository.

## Changing Teamwork Server Debugging Mode

You can change Teamwork Server debugging mode without restart.

To enable Teamwork Server debugging:

1. Start Teamwork Server Administrator's Console.
2. Select **Enable Debugging** from the Menu (Figure 13 on page 48).


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

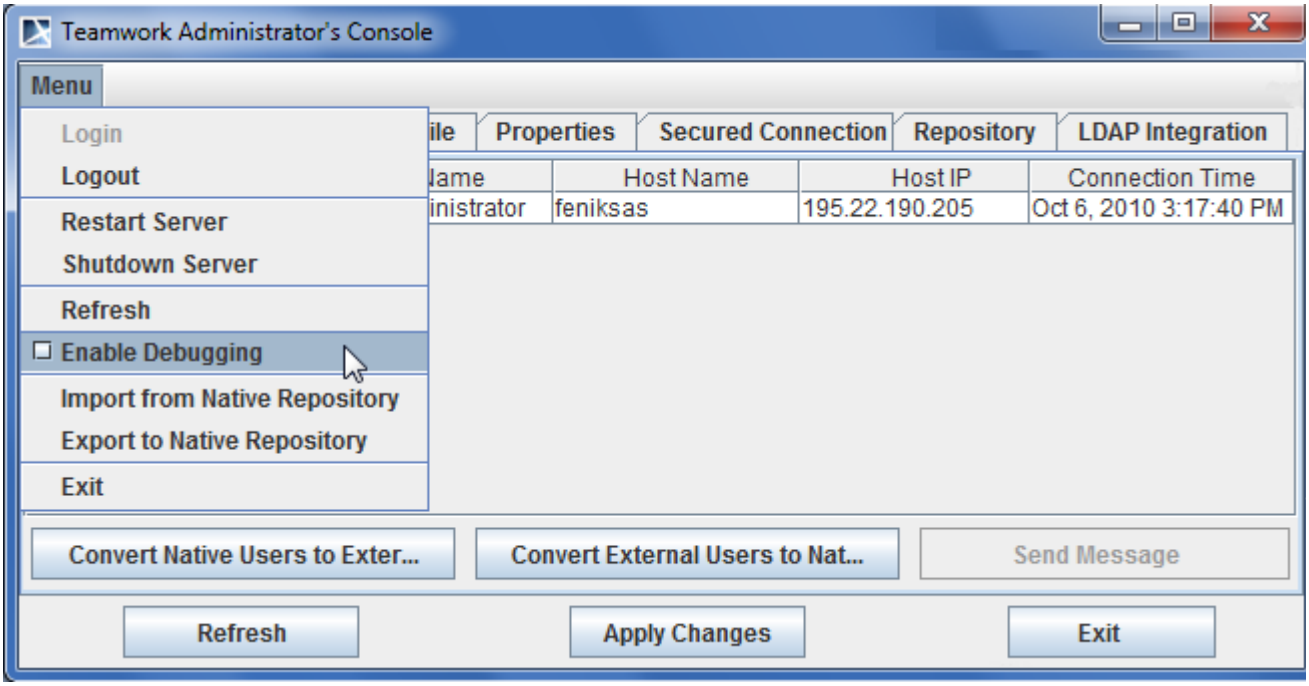



*Figure 13 -- Teamwork Administrator's Console - Enable Debugging*

# 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.

## Enabling LDAP Integration

To enable LDAP Integration

1. Start Teamwork Administrator's Console (see "Starting the Administrator's Console" on page 28).
2. Click the **LDAP Integration** tab.
3. Click **Enable LDAP Integration**. LDAP integration settings become active (see the following figure).
4. After enabling the integration, specify all mandatory setting values. Mandatory settings are marked with the star sign at the end of name. There are four groups of settings in the **LDAP Integration** tab:
   - Connection Settings (see on page 49)
   - Authentication Settings (see on page 50)
   - User Data Retrieval Settings (see on page 55)

5. Click **Apply Changes** when you are done. You will be required to test the connection against the LDAP server.


For more information about the connection testing, see "Connection Testing" on page 56.

## Connection Settings

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

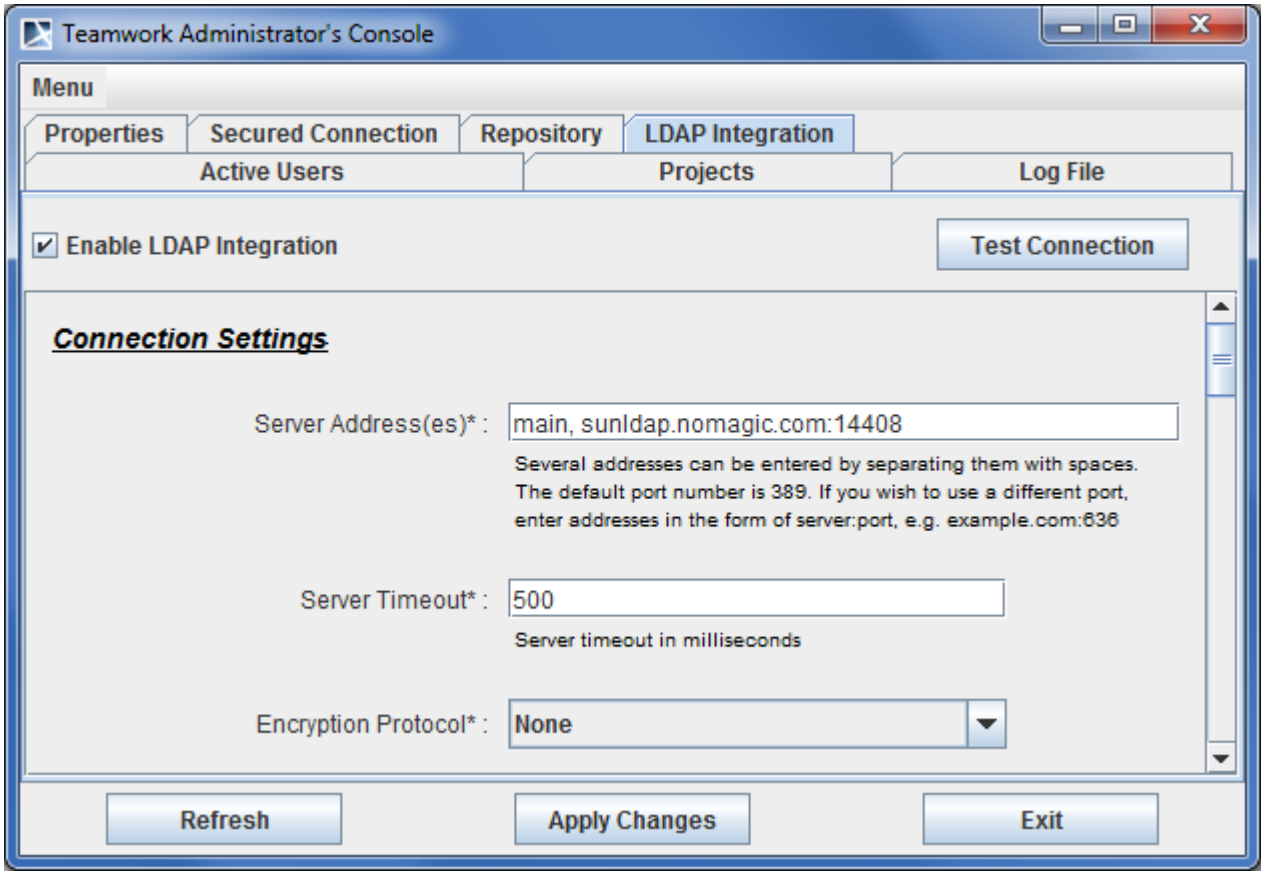



*Figure 14 -- Teamwork Administrator's Console, LDAP Integration tab. Connection Settings*

**Connection Settings** are described in the following table.

| Setting Name | Description |
|---|---|
| **Server Address(es)** | A list of servers separated by spaces where each entry holds 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.<br>Teamwork Server authenticates against servers in the order they are listed in the **Server Adress(es)**. If authentication does not succeed to the first server in the list, the second server is used, and so on until authentication is successful.<br>Authentication failure for the whole integration is considered as an inability to authenticate against any of the specified servers.<br>A single server in the specified list is queried within the period of time specified in the **Server Timeout** setting. |

| Setting Name | Description |
|---|---|
| **Server Timeout** | A time duration that specifies 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. |
| **Encryption protocol** | 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 the servers specified in the **Server Address(es)** list will be encrypted using the SSL protocol.<br>If the encrypted connection is used, Server Root CA certificate and password may be required. Server Root CA certificate should be stored in Java Key Store (JKS) format. For more information about converting certificates to JKS format, see "Converting Certificates to JKS Format" on page 57. |

## Authentication Settings

LDAP Integration supports two most popular LDAP authentication methods. They are as follows:

- Simple User+Password (see "Authentication settings for the Simple User+Password authentication type" on page 51).
- SASL (see "Authentication settings for the SASL authentication type" on page 54)

Teamwork Server transforms user credentials entered in MagicDraw to LDAP authentication credentials by 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 as they have no passwords (in order to complete authentication, authentication to LDAP server(s) is used). You can perform various actions for these users, otherwise. It is possible to setup permissions, remove users, and do other common actions with users.

You can automatically create a proxy user account for Teamwork Server. Select **Auto-add unknown users if they login successfully** checkbox and authenticated user credentials without password 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.

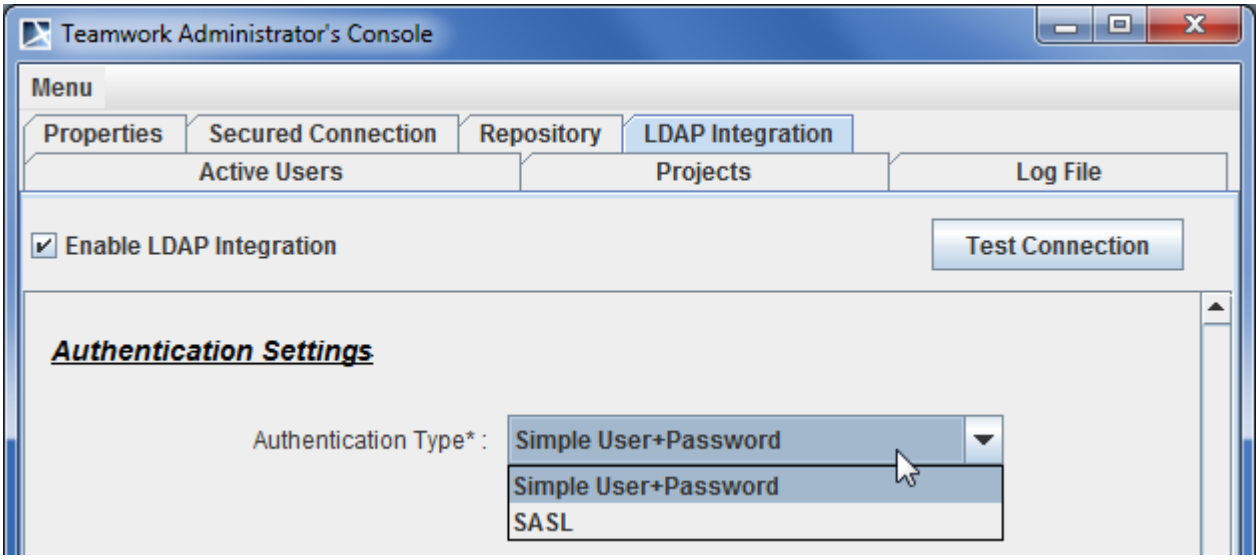



*Figure 15 -- Teamwork Administrator's Console, LDAP Integration tab. Selecting Authentication Type*

## Authentication settings for the Simple User+Password authentication type

Using the **Simple User+Password** authentication type, you can select the following options:

- **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 the login process to Teamwork Server.
- User DN is used to login to LDAP server.

Authentication using retrieved user DN performs 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 User DN by 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 returned User DN and password supplied by the user during the login process.

.

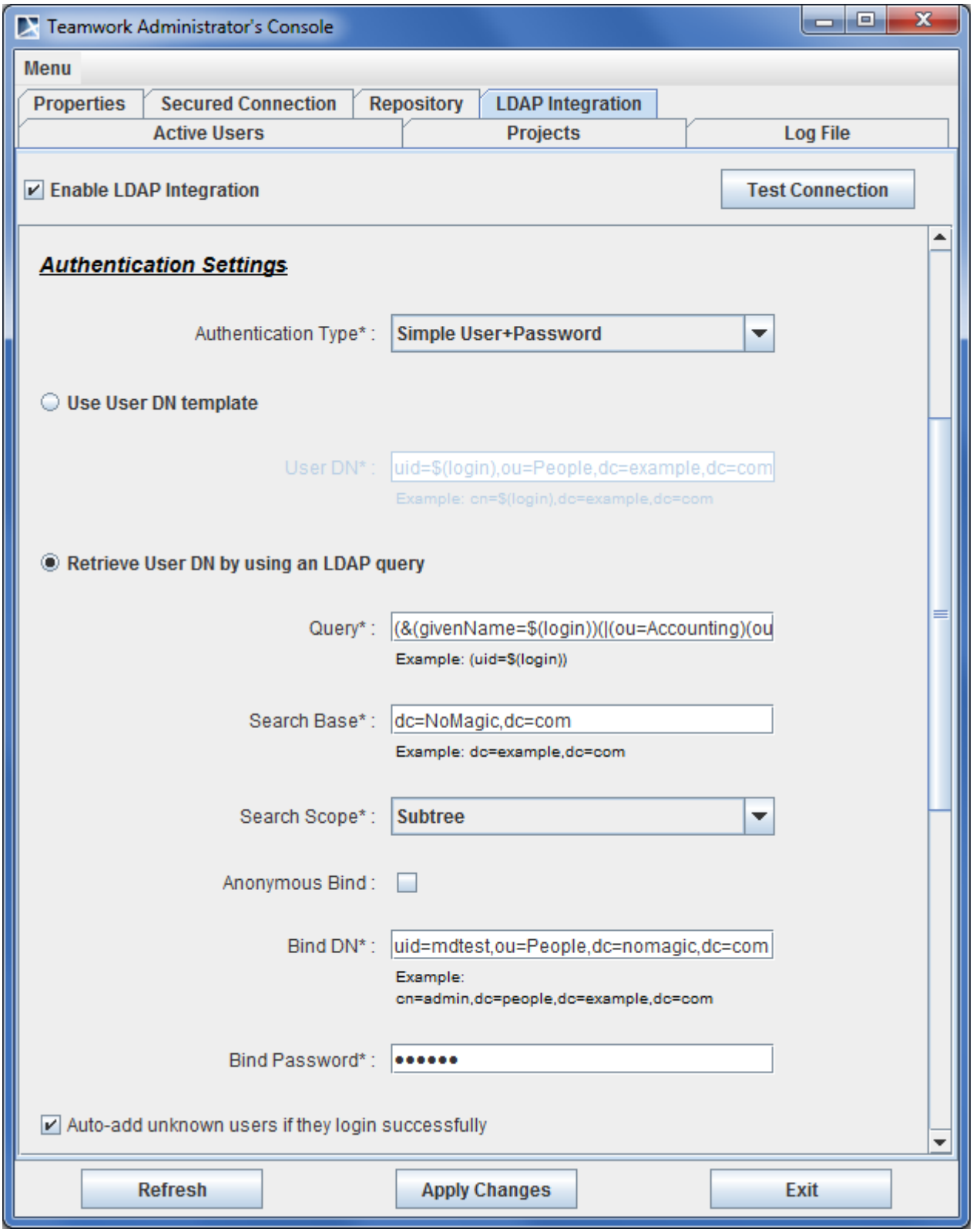



*Figure 16 -- Teamwork Administrator's Console, LDAP Integration tab. Authentication Settings (Simple User+Password)*

**Authentication Settings** are described in the following table.

| Setting Name | Description |
|---|---|
| Settings that are active when the **Use User DN template** is selected | |

| Setting Name | Description |
|---|---|
| **User DN** | User DN stores a template, which is used for mapping the user's authenticating against Teamwork Server to LDAP distinguished names when authenticating. The template recognizes a single keyword $(login). An example of the template:<br>`cn=$(login), dc=example, dc=com` |
| Settings that are active when the **Retrieve User DN by using an LDAP query** is selected | |
| **Query** | The LDAP query for retrieving User DN, for example:<br>`uid=$(login)` |
| **Search Base** | DN, from which a search should begin, for example:<br>`dc=example,dc=com` |
| **Search Scope** | Search scope specifies whether the search must be restricted to the directly owned DNs only or it must be performed in the whole subtree.<br>Choose one of the following:<br>• One level<br>• Subtree |
| **Anonymous Bind** | A mode of bind, specifying, whether the user connects to LDAP server with a specific user or anonymously for being able to find the User DN which corresponds to the user who is trying to login to Teamwork.<br>**IMPORTANT!** You must have such a user if you do not have anonymous access. |
| **Bind DN** | Specific User DN for connecting to the LDAP server and perform queries.<br>**NOTE:** This element is active, when **Anonymous Bind** is not selected. |
| **Bind Password** | A specific password to connect to the LDAP server and perform queries (you must to have such a user if you do not have anonymous access!)<br>**NOTE:** This element is active, when the **Anonymous Bind** is not selected. |

## Authentication settings for the SASL authentication type

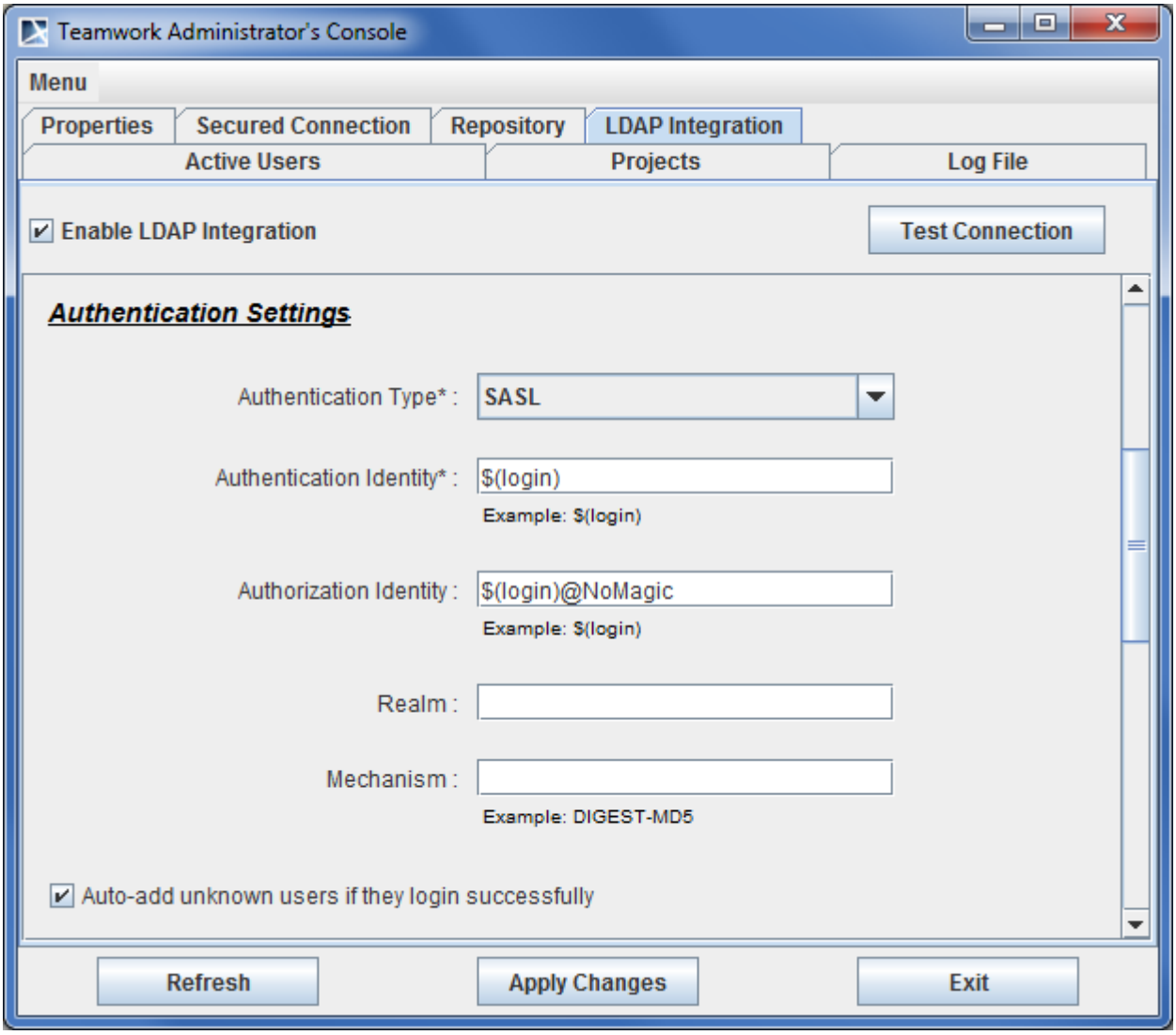



*Figure 17 -- Teamwork Administrator's Console, LDAP Integration tab. Authentication Settings (SASL)*

**Authentication Settings** are described in the following table.

| Setting Name | Description |
|---|---|
| **Authentication Identity** | 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 $(login). An example of a template:<br>`$(login)` |
| **Authorization Identity** | 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 $(login).<br>An example of a template:<br>`$(login)`<br>or<br>`$(login)@example` |
| **Realm** | Specifies the realm of an authentication identity for the SASL bind. |
| **Mechanism** | Specifies the SASL mechanism to be used for authentication. An example:<br>`DIGEST-MD5` |

## User Data Retrieval Settings

Authenticated users usually have an access to User DN attributes in the LDAP database. If the 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.

User DN is retrieved in the same way as it is done 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). When the user logs in to the LDAP server, this connection is further reused for retrieving user information.

If the user information retrieval is disabled or User DN attributes are not accessible to the authenticated user, Teamwork Server creates an external user with the login name that was specified by the user on the authentication.

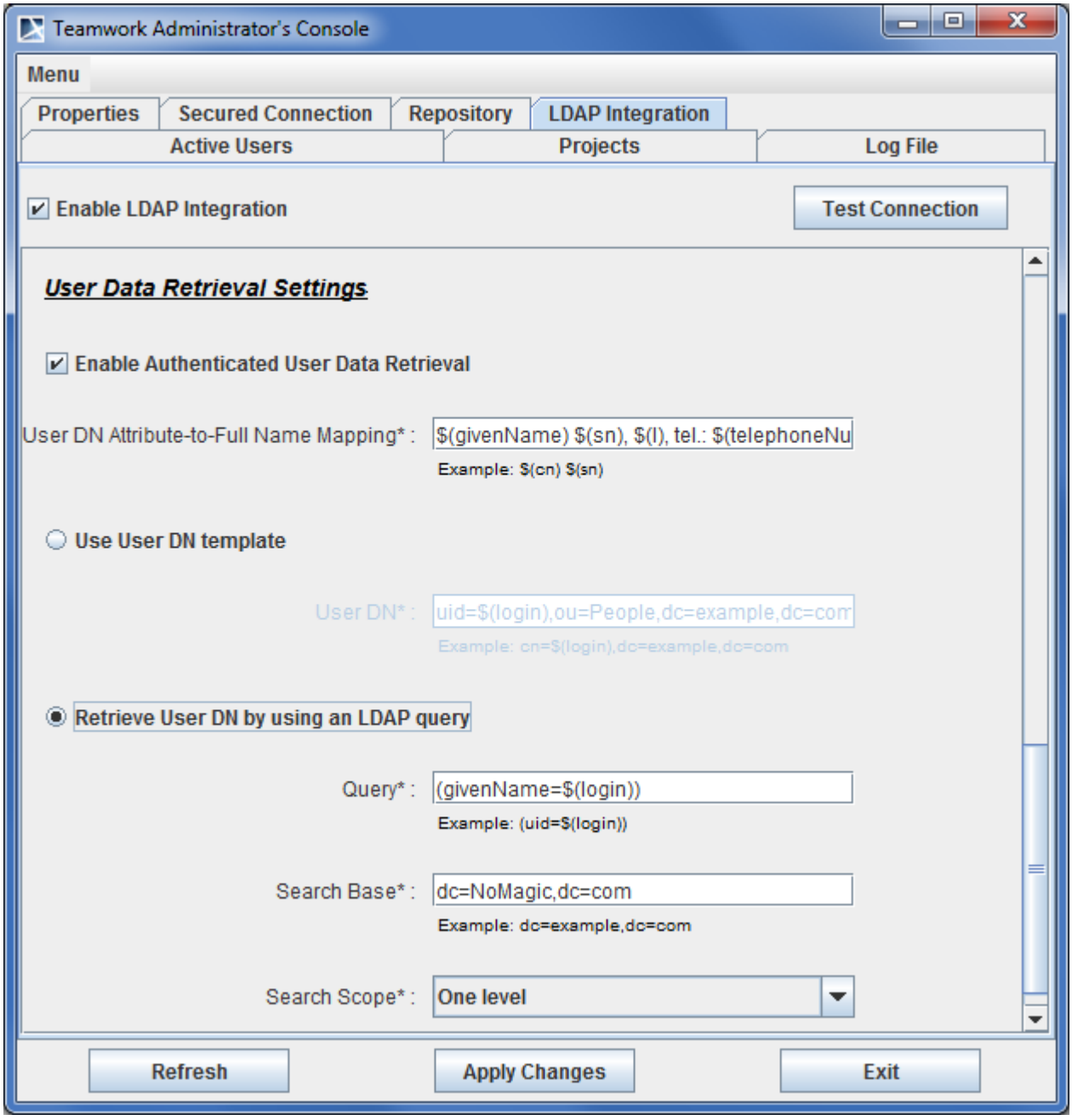



*Figure 18 -- Teamwork Administrator's Console, LDAP Integration tab. User Data Retrieval Settings*

**User Data Retrieval Settings** are described in the following table:

| Setting Name | Description |
|---|---|
| **User DN Attribute-to-Full Name Mapping** | After a specific User DN is found, the name of a local user which is created on the authentication is created by using the Full Name Mapping template for this User DN. The Full Name Mapping template supports placeholders in the form of *$(attribute)*, where *attribute* is an attribute of DN.<br>An example:<br>`$(cn) $(sn)`<br>This will form the Name of the created user out of two LDAP attributes - *cn* and *sn*. |
| Settings that are active when the **Use User DN template**n is selected: | |
| **User DN** | User DN specifies a template used to search for specific DN by the supplied login name. An example:<br>`cn=$(login), dc=example, dc=com` |
| Settings that are active when the **Retrieve User DN by using an LDAP query** is selected: | |
| **Query** | An LDAP query for retrieving User DN, for example:<br>`uid=$(login)` |
| **Search Base** | DN from which the searching should begin, for example:<br>`dc=example,dc=com` |
| **Search Scope** | Specifies, whether the search must be restricted to the directly owned DNs only or it must be performed in the whole subtree.<br>Choose one of the following options:<br>• One level<br>• Subtree |

## Connection Testing

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

To test a LDAP Integration configuration

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

## 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 that are used to authenticate against the LDAP server. Administrator is the user specified in he LDAP Integration configuration.

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

## Converting Certificates to JKS Format

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

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

## Integrating Teamwork Server with SSL-Enabled Active Directory

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

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

- 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 and should 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 Microsoft Management Console, the Certificates Snap-in.

   Do not include private keys while exporting.

2. Import the CA and AD server certificates (.cer files) to Java KeyStore (.jks file), using the KeyTool IUI. Do the following steps:
   2.1. Run the KeyTool IUI.
   2.2. Double-click **Create** on the tree, and then click **Keystore** to create a new keystore file.
   2.3. Choose the JKS format and save a new keystore file.

2.4. Set the password for the keystore.

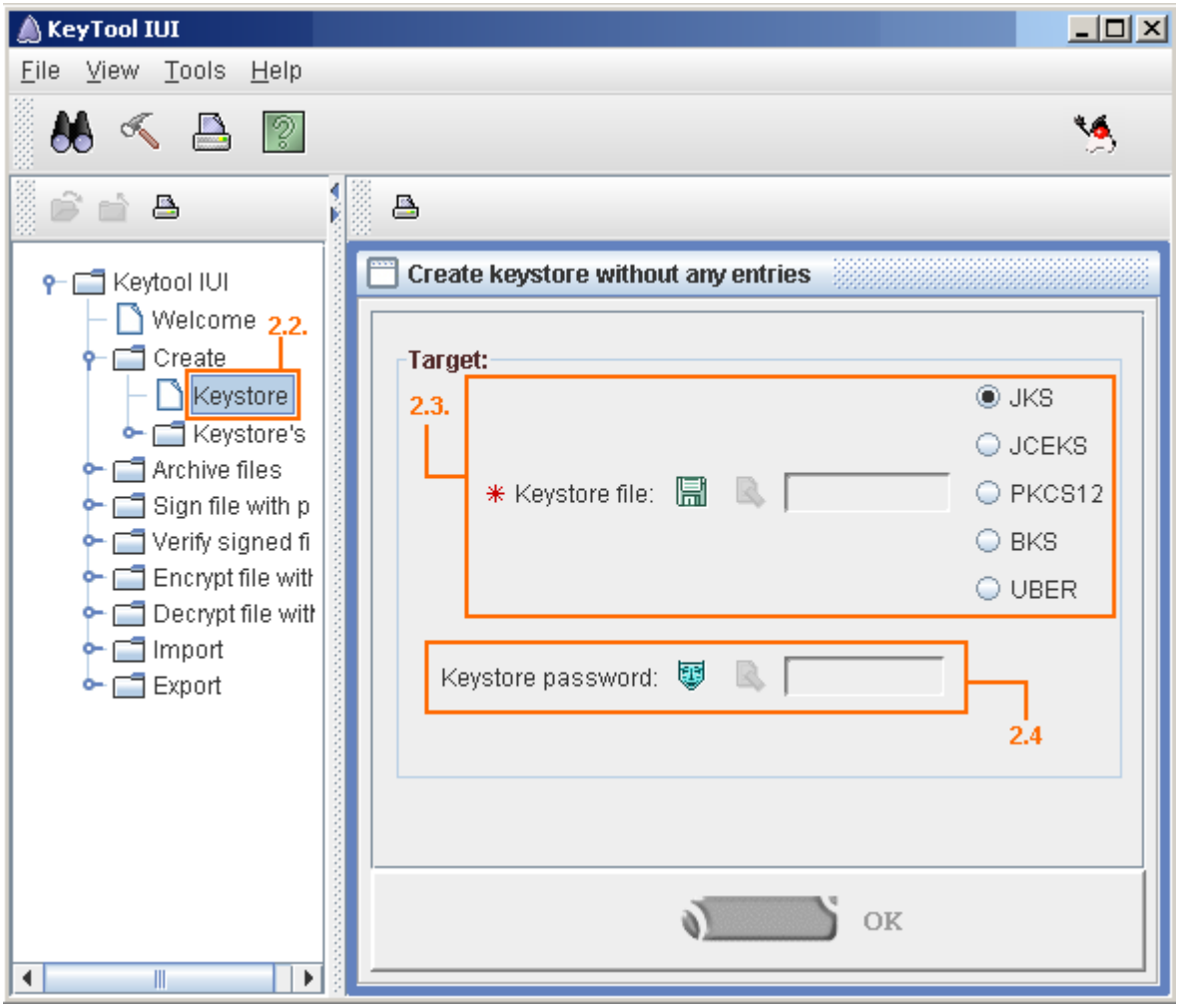



*Figure 19 -- Creating a new keystore file with the KeyTool IUI (steps 2.2, 2.3, and 2.4)*

2.5. 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.

2.6. Select the created keystore file as the target.

2.7. 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**.

2.8. 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**.

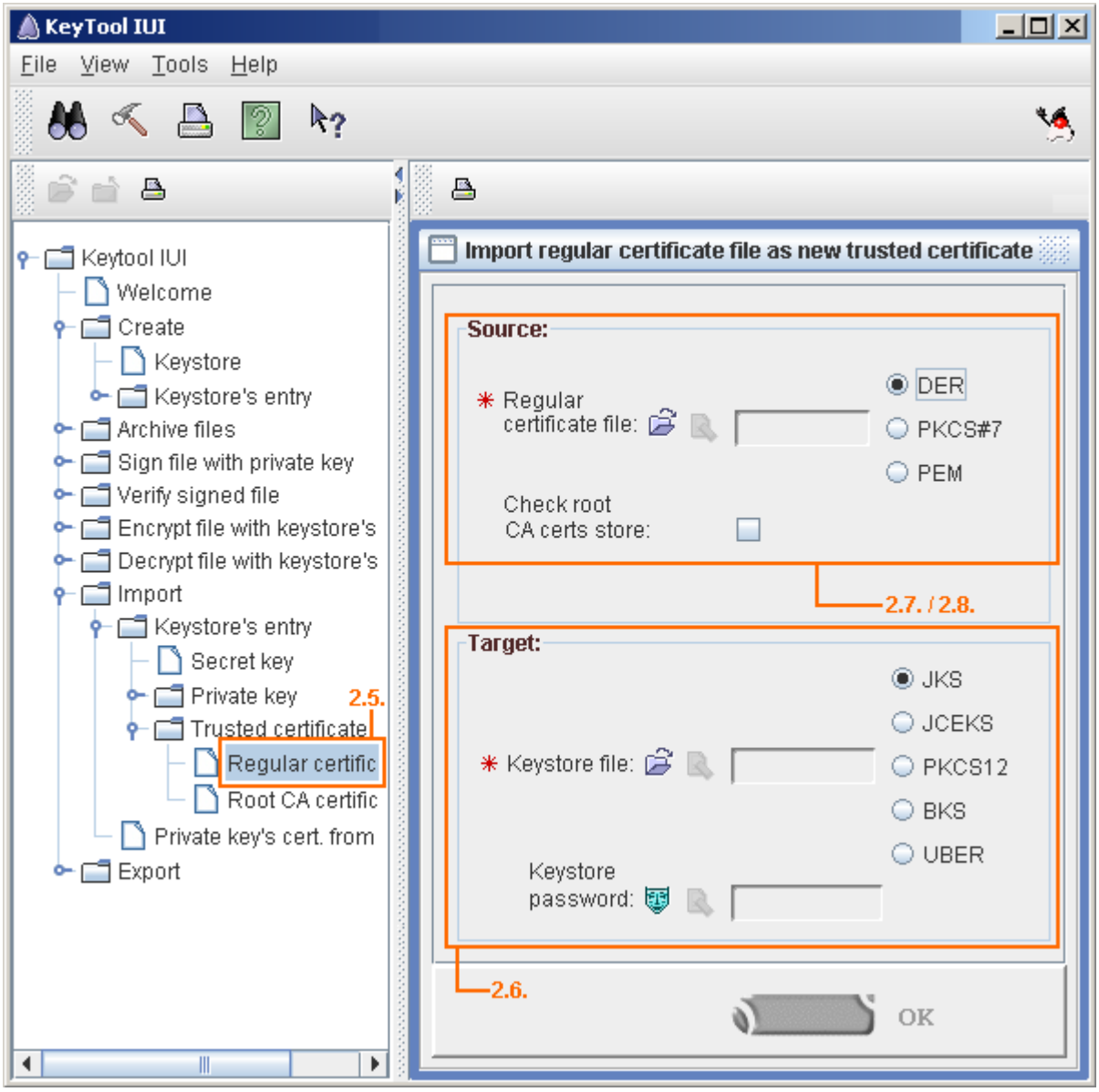



*Figure 20 -- Importing certificate files into the keystore file (steps 2.5, 2.6, 2.7, and 2.8)*

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 section "Enabling LDAP Integration" on page 48.

# Connecting Teamwork server via SSH encrypted tunnel

In this section you will find the description for SSH-encrypted MagicDraw UML and Teamwork Server connectivity.

Requirements on both server and client are as following:

1. Windows 2000/XP/Vista or Windows Server family OS.
2. Local administrative rights to create local user for tunneling and to run SSH service.
3. OpenSSH server and client binary installation files. You may obtain it from http://sourceforge.net/project/showfiles.php?group_id=103886&package_id=111688#.

The OpenSSH software is free of charge.

## Configure the Teamwork Server side

1. Login to the Teamwork Server with administrator privileges.
2. Install OpenSSH with default settings. This will install OpenSSH server and client on your Teamwork Server machine. The warning about editing password and group file appears while installing. Click **OK**.
3. Create a local user for SSH tunnel. To do this right 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 (see Figure 21 on page 60).
   3.1. Enter a new username which you will use to login into SSH service to establish tunneling. For example, *tunnel*.
   3.2. Enter the user password according to your local system policy.
   3.3. Clear the **User must change password at next logon** check box.
   3.4. Click **Create**. The local user will be created.

NOTE

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

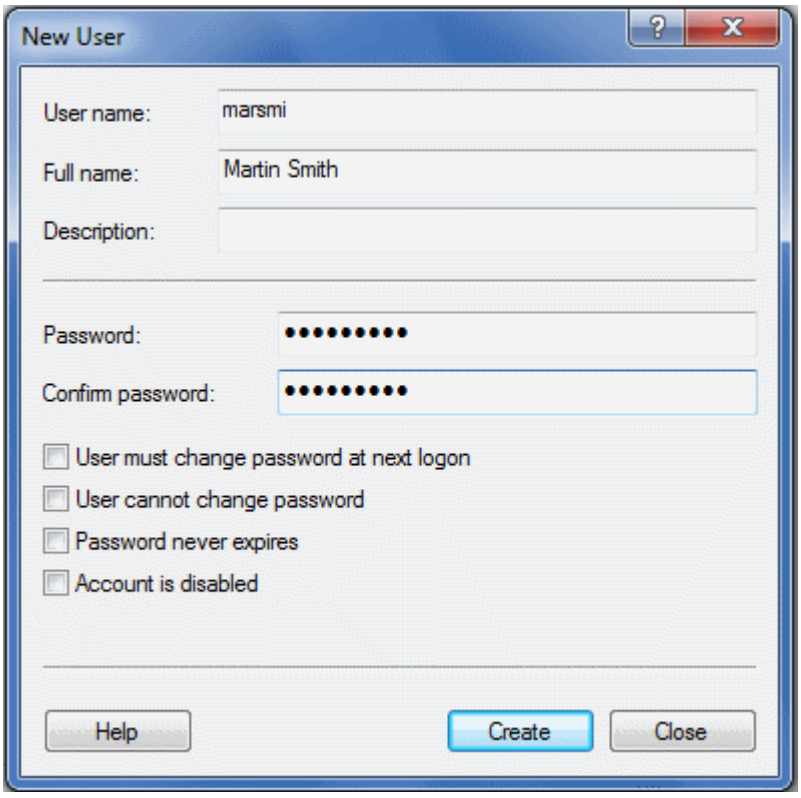



*Figure 21 -- The New User dialog*

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 (see Figure 22 on page 61).

4.1. Enter a new groupname, for example, *SSH*.

4.2. Add the “tunnel” user to the SSH group.

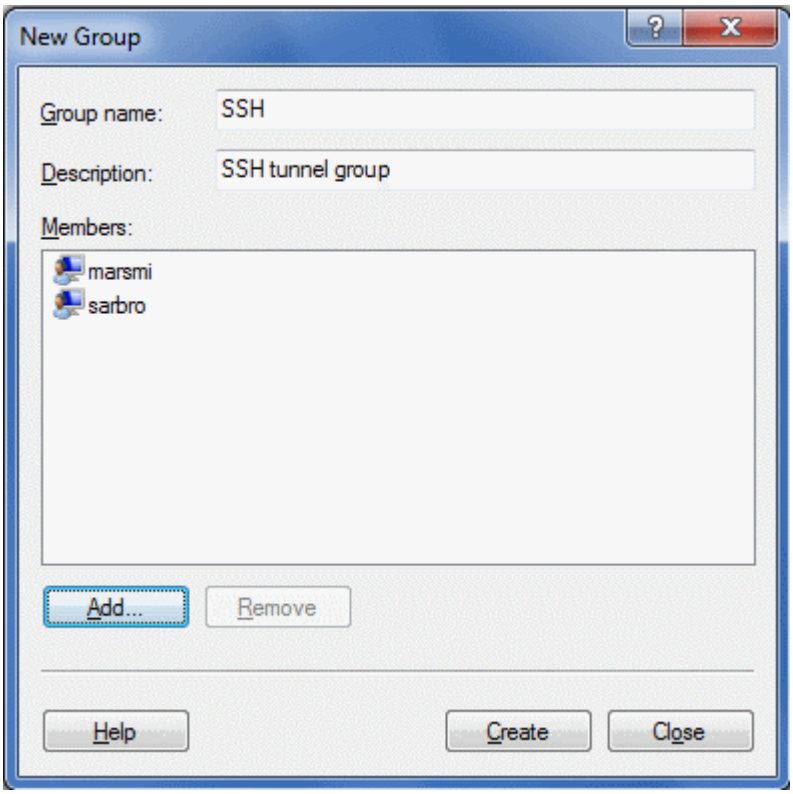



*Figure 22 -- The New Group dialog*

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 SSH-aware local password file run command prompt (click "Start"-> "Run", then type "cmd" and click Enter) and then type the following commands:

```
cd C:\Program Files\OpenSSH\bin
mkgroup -l >> ..\etc\group
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, the **Services** item right-click on the **OpenSSH Server** service and choose **Start**.

7. Test the SSH server.

7.1. 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)?
```

7.2. Type **yes** and press Enter. NOTE: type the full word “yes”, not only “y”.

7.3. Enter the password you created at step 3.

7.4. Warning about nonexisting home directory appears. Please ignore it.

7.5. Now you are logged in into localhost via SSH service and you can see the shell prompt.

7.6. After the SSH server testing exit from the server by typing *exit*.

NOTE

Other 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.

### Client Side Configuration

1. Install OpenSSH client from the same installation file you downloaded. Uncheck "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 Teamwork server SSH service from the command prompt:

ssh - L localport: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 get encrypted and sent to twserver:1100, where actual Teamwork server resides:

```
ssh -L 1100:twserver:1100 tunnel@twserver
```

localport may be any unused port on your workstation.

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 opened, as killing it also kills the SSH tunnel used for the MagicDraw Client.

3. Open MagicDraw Client. Use localhost:localport when connecting to the Teamwork server. Localport is 1100 in this case as we used it when created the tunnel.


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

# Server Synchronization and Multisite Deployment

## Synchronization Overview

Ability to synchronize data between several Teamwork Servers is available as of version 17.0.3. One Teamwork Server can now expose and make accessible the projects of the different servers together with owned projects. Server users can access remote projects (coming from the different locations) in a 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.

### Usage Scenarios, Multisite Deployment

The main usage scenario for the inter-server synchronization capability is 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 inside the site; while connectivity between the sites is not so good - WAN-class speeds (several Mbit/s) are available between sites.

There are cases when developers at one site are responsible for development of some subset of projects, while developers at other sites just need to use (but not modify) the artifacts produced by the first team.

In the earlier versions, company had to choose one site where a single Teamwork Server will be deployed and set up the remote connections for the developers joining from the other sites.The server performance (project open, commit, update, element locking times) for developers of that site were good. But the developers at other sites suffered from degraded performance due to reduced connectivity parameters. This is especially important when managing large projects.

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

Synchronization frequency can be chosen as necessary - hourly, daily, weekly


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

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

For example, team at one site is responsible for developing requirements project; team at another site can take this requirements project and incorporate them into their implementation project(s).

Please note that scenario where developers at different sites edit the same project is not covered. In this case one server will be designated as the home server of the project, and any developer, who needs to edit it, has to log onto the home server of the project to edit it.

There are additional, less frequent scenarios when synchronization between the servers can become handy. For example there is a scenario: there is a project set which has to be readable by a huge number of users, but is developed only by the small number of developers. In this case company may want to set up a small main development server where developers can work unaffected by large numbers of browsing users and a large server for handling all read-only traffic (possibly in a different network security zone etc).

The licensing scheme is very simple. Each Teamwork Server deployment is licensed separately - multiple sites synchronizing between each other 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 synchronization feature.

## Characteristics

The synchronization process characteristics are as follows:

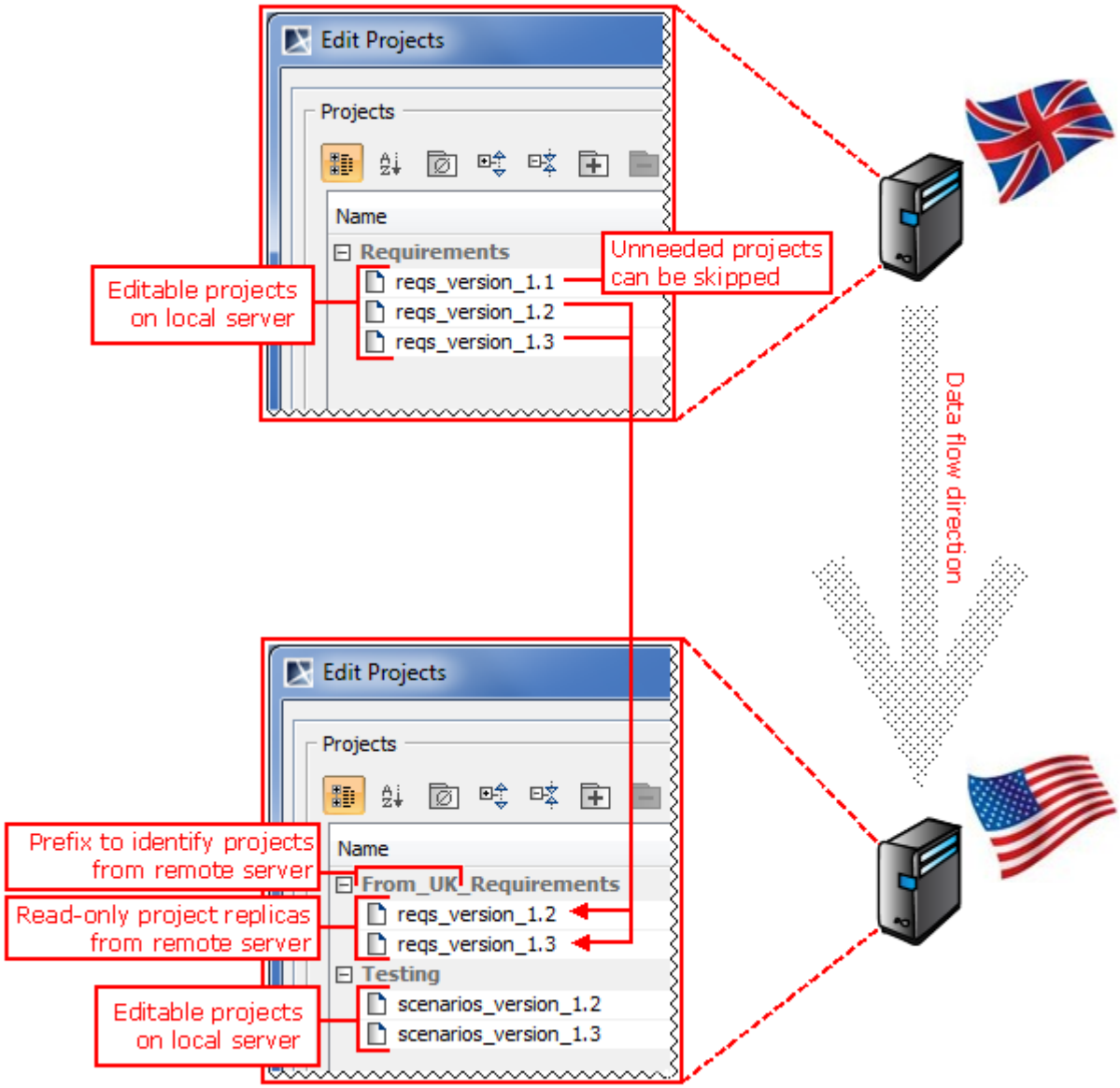
- Synchronized items are Projects and Categories. The entire project history is synchronized with all the commit data (including all the branches) and all the comments. User's data is NOT synchronized. Each server has its own independent user list and permissions are assigned on each server by the administrator of each server.
- Each project has Server-Affinity, with one writer and (potentially) many readers. Each project has a home server. The project can be edited only on home server. All other servers that fetch this project 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 the incompatible modification conflicts between the servers without introducing any additional burden - such as inter-server locks.

- Synchronization is Incremental. 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 the amount of changes since the last synchronization.
- Synchronization is Asynchronous. Servers continue to run asynchronously. Changes on one server are propagated to other servers only during synchronization procedure. Synchronization procedure can be arbitrarily delayed from the time moment when change happens (for example only monthly synchronization).
- Synchronization is Batched. Synchronization is not a continuous process. It has a clearly defined start and stop. However frequency of synchronization can be arbitrary (for example each 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 synchronization is done.
- Synchronization is Separate in each Direction. Synchronization between each source and target server pair is independent from 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 at all.
- 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 own repository.
- Synchronization is Direct only. Currently synchronization procedure can take projects only directly from their home server. Transferring projects through 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 that need to be synchronized.

## Using Synchronization

As described earlier, synchronization process is launched between two given servers - source (providing the information) and target (receiving the information). Target server is the driving server - it actively connects to the source server and pulls the necessary data. Synchronization can be bidirectional, 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 be run in parallel on the same target server, pulling the data from several different source servers.




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

The remote project can be removed from the target server, but 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 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 it's effects (no need to stop or restart servers during or after synchronization) except for the possible slight performance degradation.

The results of the pending synchronization appear on the target server gradually during the course of synchronization. So it is possible to see particular remote project X having 100 versions, then 110 after some time and 117 after synchronization is complete. But this should not cause the problems for the clients on the target - partial version data will never be offered (the situation where there are 116 versions of the remote project X declared, but the 116th version is only half-way downloaded are never exposed).

## Distinguishing between Local and Remote Projects

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

During synchronization, administrator provides (optionally) the prefix string and synchronization process modifies 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 GB site, projects coming from the server of American site can have “USA_” prefix added to their category names; and projects of German site can have “DE_” prefix. So if there are projects under category “Requirements” on the German site, on GB site these projects will be located under “DE_Requirements”.

## Controlling Synchronization Scope

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

The target server specifies user/password of the source server when target server initiates the synchronization process. Synchronization process then takes all the accessible projects from the source server. Source server administrator can control which projects are accessible managing user permissions on his server. If the specified synchronization user has a permission to read all the projects on the source server, then all the projects will be synchronized during the process. If the specified user only has permissions to access particular subset of 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.

## Controlling Synchronization Initiation

Not every user can trigger 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.

### Deleted and Inaccessible Projects

During synchronization there can be situations when project was previously located in the source server but is no longer available. Project may have been deleted in the source server or no longer accessible (by decision of source server administrator).

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

Target server administrator should inspect this category from time to time and if there are any projects, inspect whether they are used in the local projects (perhaps using the Project Usage Map.


For more information see “Project Usage Map” in MagicDraw UserManual.pdf ) and remove unnecessary projects.

### Securing the Connection between Servers

Since connection between the source and target servers is usually goring through the WAN, it is advisable to secure this connection - so that 3rd parties cannot access the project data in transit.

There are two options that can be used.

One option is to set up the SSL connection. In this case the source server needs to have public/private key pair configured in a keystore. Target server needs to have the public key (in certificate form) of each source server it wants to connect to.


For more information, see "Secured Connection tab" on page 35.

Another option is to 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).

## Running Synchronization

Synchronization procedure can be started in two different ways - using a build in scheduler in the Teamwork and using the command line utility. As stated earlier, synchronization procedure has to be triggered separately for each synchronization direction in case of bi-directional setup. The synchronization target server (the server receiving the data) is the active server - the one which drives the synchronization process. When started, synchronization process runs till completion or till the unrecoverable error appears (such as network problems) or until interrupted by the user (when using command line utility). The synchronization progress and outcome can be monitored in the server log.

### Triggering Synchronization with Internal Scheduler

Teamwork Server has a build in mechanism to trigger tasks at various periodicities 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 periodicity. There is an example file - synchronization.properties.example, which can be examined and copied/renamed to produce the synchronization task(s). For example two files - syncDE_GB.properties and syncDE_US.properties can be create to describe two synchronizations - one for synchronization from remote server on GB site, another - for synchronization from remote server on US site.

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

When scheduled tasks are run, they are run on behalf of the system user of the Teamwork Server (by the way - this is the reason why you do not need to specify the user on the target server when running synchronization using scheduler, but you need to specify user on target server when running synchronization from 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 there is only one type of task that can be scheduled - the synchronization task. This task type is specified by com.nomagic.teamwork.synchronization.SynchronizationJob value of the property.
- **com.nomagic.teamwork.schedule.cron**. This parameter specifies the frequency of task running. Parameter syntax corresponds to the Unix cron task definition syntax. This syntax allows very flexible description of task running frequency and pattern. For example, value: "0 30 01 ? * MON-FRI" triggers synchronization task at 01:30AM of each Monday to Friday. For more information about the parameter please see the documentation of Quartz framework at: http://quartz-scheduler.org/files/documentation/Quartz-2.1.x-Documentation.pdf See chapter "CronTrigger Tutorial". Alternatively you can read the cron manual of your Unix system.
- **com.nomagic.teamwork.synchronization.SourceServer**. This parameter specifies synchronization source server and port. It is a mandatory parameter for synchronization task.
- **com.nomagic.teamwork.synchronization.SourceUser**. This parameter specifies the user on the synchronization source server (the server providing the data), which will be 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, which will be used to access and fetch the project data. Since password is stored in plain text, please ensure that the *.properties file has the correct and safe permissions.
- **com.nomagic.teamwork.synchronization.UseSecureConnection**. This parameter specifies that SSL connection shall be used between target (this) 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" on page 35.
- **com.nomagic.teamwork.synchronization.CategoryPrefix**. This parameter specifies the prefix that will be prepended to each remote category name in the target (this) server. For example if 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 schedule event happens before 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.

## Triggering Synchronization Using Command Line Utility

If running the synchronization through the scheduler mechanism is not flexible enough (perhaps you want to use the mechanisms of your OS for scheduling tasks, or perhaps you want to 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 in the Teamwork Server installation directory. The utility is called **synchronize.exe** for Windows deployments and **synchronize** for Unix deployments. 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 utility is run, it connects to the target server and initiates the synchronization procedure on it. There is no additional load on the computer which runs the utility - procedure is run on the server side.

Utility outputs the synchronization progress messages.

Utility exits when the synchronization process is complete. It provides the exit codes - 0 in case synchronization is successful and >0 in case of errors. This exit code can then be used for shell scripting purposes (such as restarting synchronization in case of failure or similar).

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. But the utility does accept the Ctrl+C keyboard signal and stops the synchronization process on the server cleanly.

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 the 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 synchronization procedure. 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. Target server will connect to this server using the supplied credentials and pull the necessary data.

`<prefix>`
: The categories from the target server, which are pulled from the source server, will be prefixed with the specified prefix. This allows better differentiation of projects on the target - it is easier to see which projects are remote and which projects are local.

Options:

`-h, --help`
: This help message.

`-q, --quiet`
: Quiet synchronization. No information is displayed on system output stream during synchronization. Errors are still reported on 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 sysop user, p@Ssw0rd password and initiates the synchronization procedure - pulling the information from the tws.company.ca:1102 server:port, using sync user, ssApcnys password. Categories form .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 the copies of 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 in the same way as you would specify parameters on the command line. After this you get a single executable (with all parameters pre-specified), that can be run by one click.

# NEW! 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 by using any external storage device, such as 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 the contributions to the model they make can be successfully merged as well.

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

Here is the workflow on how to interchange projects between two isolated servers:

| Sharing Server | Contributing Server |
|---|---|
| 1. Export selected projects *A*, *B*, and *C* to the location *X*, a directory on the server.<br>For more information, refer to the procedure "To export projects from the current server" on page 32. | |
| 2. Copy the exported data to some external storage device, for example, CD. | |
| 3. Deliver the external storage device to the contributing server. | |
| | 4. Copy data from the external storage device to the location *Y*, a directory on the server. |
| | 5. Synchronize the server repository with data from the location *Y*. For more information, refer to "NEW! To synchronize projects on the current server with data from selected location" on page 30.<br><br>After the synchronization finishes, the server gets 3 new read-only projects: *A*, *B*, and *C*. |
| If you need to update the projects on the contributing server with new data from the sharing server, repeat steps from 1 to 5.<br>If you need to modify these projects on the contributing server and then transfer the changes to the sharing server, perform the steps from 6 to 12. | |
| | 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 the location *Z* on the server. Projects are exported with all their branches including the branch *Contrib*. |
| | 9. Copy the exported data to some external storage device, for example, CD. |
| | 10. Deliver the external storage device back to the sharing server. |
| 11. Copy the data from the external storage device to the location *W* on the server. | |

| Sharing Server | Contributing Server |
| --- | --- |
| 12. Synchronize the server repository with data from the location *W*. For more information, refer to "NEW! To synchronize projects on the current server with data from selected location" on page 30.<br><br>After the synchronization finishes, the projects *A*, *B*, and *C* gets a new read-only branch *Contrib* on the server.<br><br>To review the transferred changes and incorporate them into the trunk, merge the projects from this new branch with the projects in the trunk. For more information, refer to “Model Merge” in *MagicDraw User Manual.pdf*.<br>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 dialog” in *MagicDraw User Manual.pdf*. | |

If you need to update the projects on the sharing server with new data from the contributing server, repeat steps from 8 to 12.