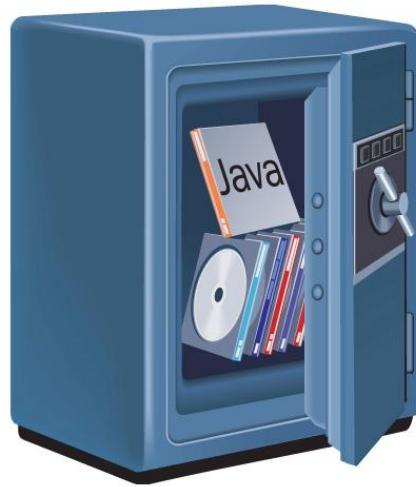jProductivity LLC

Protect your investments with Protection!

Protection!™

Licensing Server

v5

Administrator Guide

Notice of Copyright
Published by jProductivity, LLC
Copyright ©2003-2021
All rights reserved.

Registered Trademarks and Proprietary Names
Product names mentioned in this document may be trademarks or registered trademarks of jProductivity, LLC or other hardware, software, or service providers and are used herein for identification purposes only.

Applicability
This document applies to Protection! Licensing Server v5 software.

Contents

Chapter 1

Protection! Licensing Server Overview

Protection! Licensing Server consists of two major components to support the Licensing Server functionality – the Licensing Server itself and the Management Console.

1.1 Protection! Licensing Server

Licensing Server is responsible for:
- Distribution of licenses so a client application may ask Licensing Server to obtain (download) desired license for a certain user. To achieve it - licenses to distribute must be added to the Licensing Server and allocated to the particular users or groups. See Allocation of Licenses for Specific Users and/or User Groups for more information.
- Tracking of the concurrent use of licenses to support Floating and Named licensing models. Those models assume that only a certain number of users can work concurrently with the license according to the license number of copies value. Actual usage check is done by contacting the Licensing Server and acquiring the License Lock. After the application quits - it must release License Lock to free that license and make it available for other users.

Note	There are almost completely independent sub-systems for licenses distribution and for concurrent models support. Therefore: a) Licenses of any type and any user licensing model can be distributed by the Licensing Server. b) Licensing Server allows tracking of concurrent use of the licenses regardless how those licenses were obtained.

The Licensing Server application is a turnkey solution that is deployed inside the customer's network. It can be run as a Windows service or as a UNIX/Linux daemon. The main features of the Licensing Server are:
- Support for Floating and Named User models by tracking licenses usage using the concept of license lock (acquire) and unlock (release).
- Ability to revoke license locks sessions to force client applications to unlock (release) licenses.
- License lock session's expiration to allow correctly releasing of the license lock when the client application is not responding or the network (connection to the Licensing Server) is down.
- Ability to host and to distribute an unlimited number of licenses for an unlimited number of products.
- Ability to allocate licenses for specified users and/or user groups.
- Ability to control access to the Licensing Server by using access control lists (whitelist and blacklist) for any types of operations.
- Ability to remotely monitor and manage the Licensing Server via supplied Management Console application.

1.2 Protection! Licensing Server Management Console

The Management Console is a GUI application intended for remote monitoring and management the Licensing Server(s). The main features of the Management Console are:

- Authentication and authorization per Licensing Server with the ability to maintain the Licensing Server users.
- Support for multiple Licensing Server connections.
- Ability to discover running instances of the Licensing Servers via network broadcast.
- Ability to maintain access control lists.
- Ability to view licensing statistics and server statistics.
- Ability to disable/enable any licensing activities.
- Ability to shut down the Licensing Server.
- Ability to visually monitor current license sessions.
- Ability to revoke specific license sessions.
- Ability to maintain licenses storage.
- Ability to maintain license users/groups
- Ability to allocate licenses for specified users and/or user groups.
- Ability to allocate licenses to the users fetched from LDAP directory
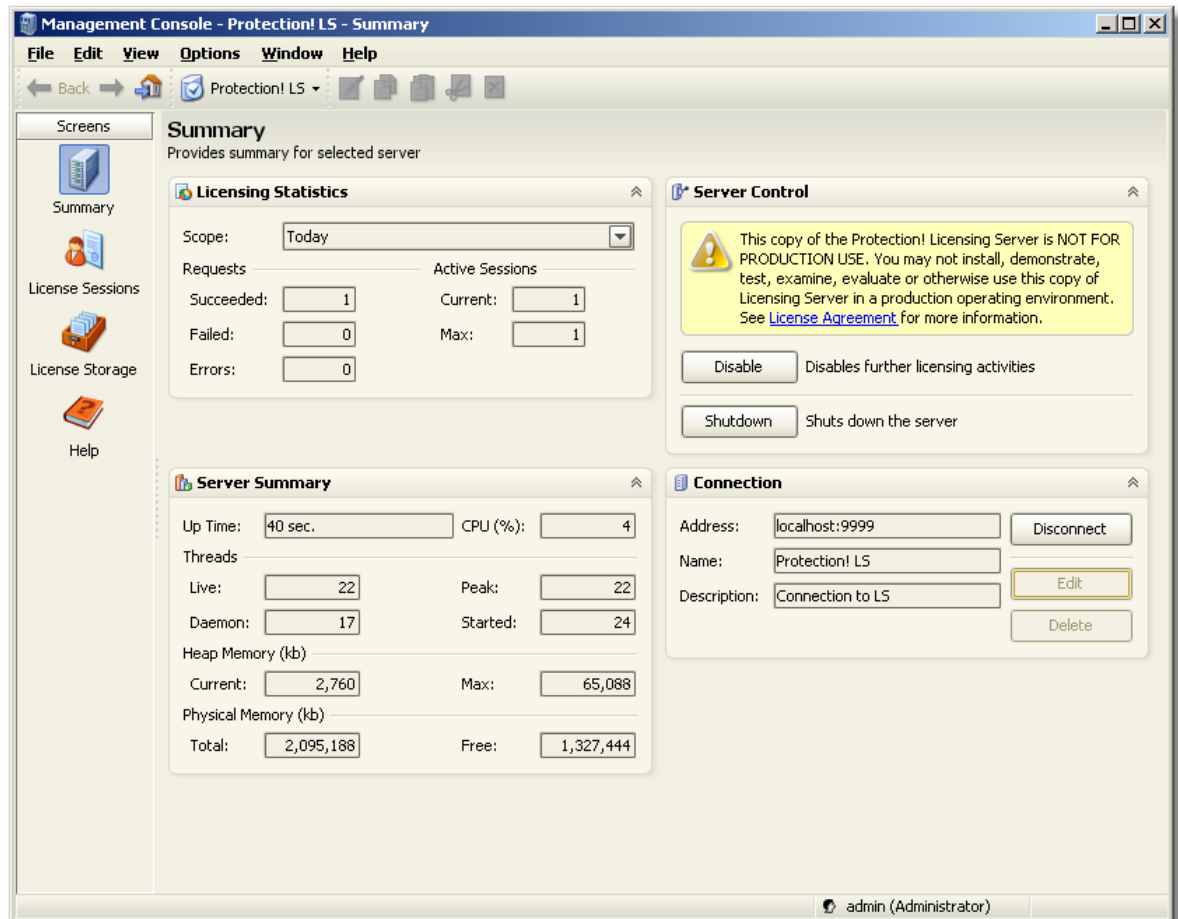
Figure 1: Licensing Server Management Console

Chapter 2

Working with the Protection! Licensing Server

This topic outlines how to work with the Licensing Server application running on the different platforms.

2.1 Running the Licensing Server

The Licensing Server application is a turnkey solution that is deployed inside the customer's network. It can be run as a Windows service or as a UNIX/Linux daemon. Licensing Server can also be started as an application via the command line:

```
java –jar ProtectionLS.jar
```

The following screenshot illustrates sample output when Licensing Server has been started via command line:

Figure 2: Starting the Licensing Server via command line

2.1.1 Microsoft Windows

The Licensing Server Windows service is registered by the Licensing Server installer. It is also possible to install services from the command line by passing `/install` to the `ProtectionLS.exe` service executable. If the Licensing Server service is installed via command line then the default start mode of the service can be determined as follows:

- Start on demand - In start on demand mode, the Licensing Server service must be manually started by the user in the Windows service manager. This start-up mode can be forced on the command line if the user passes `/install-demand` to the Licensing Server executable instead of `/install`.

- Auto start - In auto start mode, the Licensing Server service is always started when Windows is booted. This installation mode can be forced on the command line if the user passes `/install-auto` to the Licensing Server executable instead of `/install`.
- Windows services are always uninstalled by passing `/uninstall` to the Licensing Server service executable. All command line switches also work with a prefixed dash instead of a slash (like `-uninstall`) or two prefixed dashes (like `--uninstall`).

To start or stop the Licensing Server service, the `/start` and `/stop` options are available respectively. In addition, a `/status` argument shows if the service is already running. The exit code of the status command is `0` when the Licensing Server service is running, `3` when it is not running and `1` when the state cannot be determined (for example when it is not installed on Windows).

As a second parameter after the `/install` parameter, you can optionally pass a service name. In that way you can install the Licensing Server service with a different service name than the default name. Protection! Licensing Server will allow only a single running service instance regardless if multiple services with different names were installed. You will not be able to use the same Licensing Server service executable to start multiple services with different names.

Note	If you install Licensing Server service with a different name then its default name then you also have to pass the same service name as the second parameter if you use the `/uninstall`, `/start` and `/stop` parameters.

2.1.2 Unix

For Unix service executables, the `start`, `stop` and `status` arguments are available for the Licensing Server start script (by default script is called `ProtectionLS` and it is located in the `/bin` directory). The `stop` command waits for the Licensing Server service to shut down. The exit code of the status command is `0` when the Licensing Server service is running and `3` when it is not running.

On Unix-like platforms, the Licensing Server start/stop script has to be integrated into the boot sequence by the administrator.

Tip	If you experience any issues during functioning of the Licensing Server - turn on DEBUG logging to see some more detailed info.

2.2 Shutting Down the Licensing Server

To shutdown the Licensing Server:
- If the Licensing Server is running as a Windows service or as a UNIX/Linux daemon - use an appropriate OS command to stop the service/daemon or by passing /stop to the Licensing Server service executable.
- If the Licensing Server started as an application via the command line then to shut down the server send an appropriate interrupt command (e.g., *Ctrl+C*).
- You can also stop the Licensing Server service via the Licensing Server Management Console. In the Management Console click on Summary shortcut on the Shortcuts Side Bar and in the Server Control panel click the "Shutdown" button (see the topic "*Shutting Down the Licensing Server via Management Console*" later in this Guide).

2.3 Analyzing the Licensing Server Logs

All of the Licensing Server activities are written into the log files located in the *<Licensing_Server_Install_Folder>/logs* folder. The logs are rolled over daily by default where the older (rolled over) logs are automatically renamed using the following naming convention `<log_name.log.yyyy-mm-dd>`.

There are 2 (two) server logs – *access.log* and *error.log*. Access log contain all of the Licensing Server start/stop events, all licensing activities and errors (e.g., when a port is occupied by a different service). Error log just contains all of the Licensing Server errors.

It is possible to configure and fine tune the Licensing Server logging sub-system by modifying the *<Licensing_Server_Install_Folder>/conf/log4j.properties* file. Please consult the log4j documentation (http://logging.apache.org/log4j/docs/) for more information.

2.4 Configuring the Licensing Server

The Licensing Server configuration can be done by modifying the `bin/server.xml` file. The Licensing Server must be restarted to apply changes in the configuration.

2.4.1 Configuring the RMI Licensing Server Interface

To configure RMI Licensing Server interface - alter the `server.xml` file and change attributes of the `rmiInterfaceModule` element:

- `hostName` - represents the fully qualified host name or public IP address of the physical server the Licensing Server is running on. The `hostName` attribute must be explicitly specified when the server is running behind the firewall. Default value is "" which means that local IP of the server will be used.
- `port` - the port that is used by the Licensing Server to accept requests from the clients. Default value for the `port` attribute is *1099*. If a different value is specified for the *port* attribute then it must be explicitly specified for all of the Licensing Server connections.
- `exportPort` – represents the port number that is used by the Licensing Sever to receive calls from the clients. Default value for this attribute is `0` which means that the server's listening port is chosen arbitrarily at runtime. The `exportPort` attribute must be explicitly specified when the server is running behind the firewall.
- `secure` – specifies that RMI over SSL is turned ON. *Note: it is OFF by default to maintain compatibility with existing apps.*

To enable *"RMI over SSL"* for the customer apps:
a. Edit your product in Developer Control Center.
b. Open the *"Integration | Licensing Server"* page
c. Change the *Connection* property to be *"RMI over SSL"*
d. Rebuild launcher.
e. Check that the server is properly configured to allow RMI over SSL (see how to configure RMI Licensing Server interface above).

The `hostName`, `port`, `exportPort` attributes must be explicitly specified when the server is running behind the firewall and corresponding ports must be opened in the firewall configuration.

2.4.2 Configuring the HTTP Licensing Server Interface

There is a new implementation of Licensing Server interface that utilizes HTTPS protocol to support secure communication between customer apps and the server. By default, the server utilizes port 443 for HTTPS communication. You can change that port to any other value by altering the `server.xml` file and changing the `securePort` attribute of the `httpInterfaceModule` element to a desired value.

To enable using HTTPS for the customer apps:
1. Edit your product in Developer Control Center.
2. Open the *"Integration | Licensing Server"* page
3. Change the *Connection* property to be *HTTPS*
4. Rebuild launcher.

It is recommended to use HTTPS in most of cases because:
1. It can be more efficient than RMI when working with big number of connections
2. It is easier to configure when the server is running behind firewall; there is just one port to open and that port can be already opened if the standard HTTPS port is utilized.

2.4.3 Configuring the Management Interface

To configure Management interface - alter the `server.xml` file and change attributes of the `managementSupport` element:

- `port` - the port that is used by the Licensing Server to accept requests from the Management Console to allow remote management of the Licensing Server. Default value for the *port* attribute is *9999*. If a different value is specified for the *port* attribute then it must be explicitly specified for all of the Management Console connections.
- `exportPort` – represents the port number that is used by the Licensing Sever to receive calls from the Management Console. Default value for this attribute is *0* which means that the server's listening port is chosen arbitrarily at runtime. The `exportPort` attribute must be explicitly specified when the server is running behind the firewall.

The `port`, `exportPort` attributes must be explicitly specified when the server is running behind the firewall and corresponding ports must be opened in the firewall configuration.

2.4.4 Specifying the System Properties

There is an ability to specify the System Properties, which may be useful to fine tune some functionalities of the server. It can be done by altering the `server.xml` file, specifying `<systemProperty/>` sub-elements of the `<systemProperties/>` element, and restarting the server if it is running:

```
<server>
     <systemProperties>
          <systemProperty
           name="a" value="value1"/>
          <systemProperty name="b" value="value2"/>
     </systemProperties>
…
<server/>
```

If there is a need to examine the System Properties of the running server – use Management Console and show the system information via the "*About | System Information*" menu item.

2.4.5 Configuring the Server Users Authenticator

To configure the Server Users Authenticator - alter the `server.xml` file and change attributes of the `authenticator` element:
1. `passwordValidityPeriod` – password validity period in days (90 days by default). To allow non-expiring passwords – state 0 for this attribute or remove it completely.

2.4.6 Supplying Valid SSL Certificates

Licensing Server uses a self-signed certificate to allow it running out of the box with no additional configuration. If there is a need to let it use a valid SSL certificate - the following should be done:
1. Obtain a valid SSL certificate for a host the Licensing Server will be up and running. Note: you must have either a certificate and a matching private key or a valid key-pair to proceed.
2. Stop the server if it is running.
3. Use Java's `keytool` utility regenerate the `/bin/.keystore` key store so it will contain a valid key pair. Note: the key store is password protected. Please contact a vendor of protected app or directly jProductivity LLC for the password.
4. Start the server.

Tip	If you need more information about certificates and converting them to keystore – please check the following links for more information: • Install a CA-signed SSL certificate with the Java keytool • Install a CA-signed SSL/TLS certificate with KeyStore Explorer • Install a CA-signed SSL certificate with OpenSSL

2.4.7 Disabling Older TLS Implementations

Support of older TLS versions such as TLS 1.0 and TLS 1.1 is widely recognized as a possible security risk. Though those versions are still enabled by default in a majority of JRE/JDK distributions - thus it is better to disable them. To disable older TLS implementations:

1. Stop the server if it is running.
2. Identify and locate a JRE the server is running on.
3. Edit `<JRE_HOME>/conf/security/java.security` (`lib/security/java.security` in JDK 8 and earlier) and add TLS 1.0 and TLS 1.1 to the `jdk.tls.disabledAlgorithms` property by appending the following values:

```
TLSv1, TLSv1.1
```

For example, if the current value is:

```
jdk.tls.disabledAlgorithms=SSLv3, RC4, DES, MD5withRSA, DH keySize <
1024, \
    EC keySize < 224, 3DES_EDE_CBC, anon, NULL
```

then the new value would be:

```
jdk.tls.disabledAlgorithms=SSLv3, RC4, DES, MD5withRSA, DH keySize <
1024, \
    EC keySize < 224, 3DES_EDE_CBC, anon, NULL, TLSv1, TLSv1.1
```

4. Start the server.

Please see more information at: https://java.com/en/configure_crypto.html

Tip If there is a need to identify and locate a JRE the server is running on - check the System Properties of the running server by using Management Console and showing the system information via the *"About | System Information"* menu item.

2.4.8 Supporting Secure LDAP

Licensing Server should be specially configured to allow it access LDAP servers using secure LDAP (LDAPS) protocol. To configure it:

1. Obtain a SSL Certificate from your LDAP server.
2. Register your server SSL certificate into Java keystore. Java offers the certificate management utility `keytool` to handle certificates into your keystore. To import a certificate, you need to specify three arguments:
   - `keystore`: Absolute path to your keystore. (for example: `C:\Program Files\Java\jdk1.7.0_79\jre\lib\security`)
   - `alias`: Give a name to your certificate The given name should not already exist in the keystore
   - `file`: Absolute path to the certificate you want to import.

     Use the following code to import your certificate into the default java keystore:

     ```
     keytool -keystore <PATH_TO_JRE>/lib/security/cacerts
             -import
             -alias certificate
             -file <PATH_TO_CERTIFICATE>/certificate.cer
     ```

3. Configure LDAP connection using the `ldaps` protocol (Management Console -> *"License Storage Screen"* -> *"Edit | LDAP Connection"*).

2.5 Automatic Reloading of Some Configuration Files

To allow bulk modification of configuration files the Licensing Server provides automatic reloading of them. This can be useful, for example, to import all the users from the corporate directory.

To prevent concurrent modification and possible data corruption you must first create an empty lock file. Such lock file should be named as a file to alter, adding *.lck* extension. Certainly, any lock files must be removed after complete of update.

For example, to update license users file (*licenseUsers.xml*): a *licenseUsers.xml.lck* file must be created; then license users file should be updated and finally the lock file should be removed.

Note	File locks may be automatically cleared (removed) after 1 min of timeout. This is done to avoid orphaned file locks that would prevent any further configuration updates done even using the Management Console. So any updates should be done within 1 min time range.

The following files in the *<Licensing_Server_Install_Folder>/conf/* can be updated on the fly:

- *acl_named.xml* - ACL | Lock License-Named
- *acl_floating.xml* - ACL | Lock License-Floating
- *acl_get.xm*l - ACL | Obtain License
- *acl_global.xml* - ACL | Global
- *licenseUsers.xml* - License Users. Note: when defying LDAP Connection attributes the password must be specified either as an unencrypted value or an encrypted value copied from *licenseUsers.xml* previously generated by Licensing Server itself.

2.6 License Inbox

Licensing Server provides the License Inbox functionality that allows easily importing and allocation of licenses with no need to launch Management Console. To use License Inbox – just copy licenses to `<LS_INSTALL_FOLDER>/inbox/` folder. Copied licenses will be promptly imported and allocated to the default License User Group (if defined). To indicate the status of the import process – licenses will be renamed after the import attempts. So successfully imported licenses will get the `.imported` sub-extension, failed licenses will get the `.failed` sub-extension.

If you need to allocate imported licenses to different License User Groups or to License Users – create sub-folders inside the inbox folder with names matching the names of desired License User Groups or License Users and copy licenses inside those sub-folders.

By default, License Inbox folder is `<LS_INSTALL_FOLDER>/inbox/`. If you want to use a different folder – alter the `server.xml` file and specify a new path as `inboxPath` attribute of the `<module id="licenseStorage"/>` element and restart the server.

By default, files having the `key` and `license` extensions are considered to be licenses so Licensing Server will try to import only such files. If you issue licenses having different extensions – alter the `server.xml` file and specify a comma-separated list of license extensions as the `licenseFileExtensions` attribute of the `<module id="licenseStorage"/>` element and restart the server.

Chapter 3

Working with the Licensing Server Management Console

This topic outlines how to use the Licensing Server Management Console to manage Licensing Servers.

3.1 Connecting to the Licensing Server(s)

The Licensing Server Management Console allows the administrator to connect to multiple running instances of the Licensing Server(s). If there is no connections setup then "New Connection" dialog would be launched automatically on the first run of the Licensing Server Management Console.

To set up a new connection the following steps needs to be performed:
1. Click the "New Connection" button on the Management Console toolbar or by selecting the "*File | New Connection*" menu item.
2. In the *"New/Edit Connection"* dialog shown on the screen enter the following attributes:
   a. *Name* – Unique name for connection. This name will be shown in the connections drop-down on the Management Console toolbar;
   b. *Description* – detailed description of the connection;
   c. *Host* – specifies hostname or IP address of the Licensing Server;
   d. *Port* – specifies port of the Licensing Server;
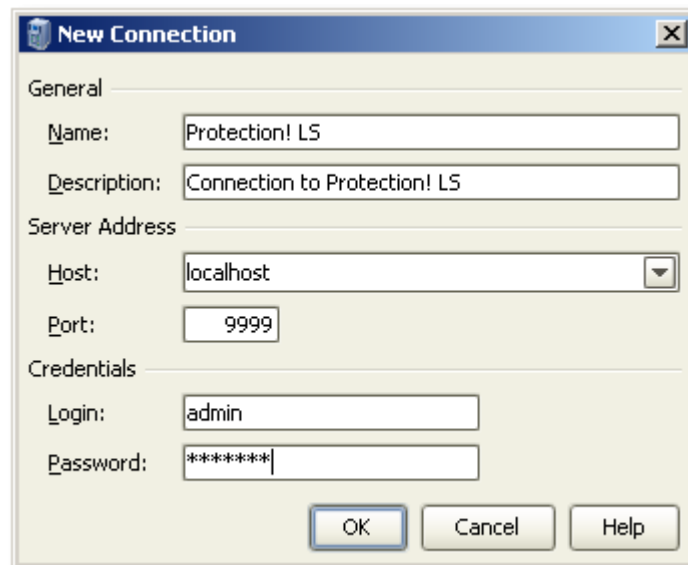   e. *Login* and *Password* required for authentication.

Figure 3: New Connection dialog

3. Press the OK button to close the dialog and to create a new connection.

When a new connection is created it will be shown in the Summary Screen | Connection dashboard outlining connection parameters.

Appropriate buttons on the application toolbar or in the Summary Screen | Connection dashboard allows the following actions:
1. Connect – connects to the specified Licensing Server;
2. Edit – Launches "Edit Connection" dialog for connection parameters modification;
3. Delete – Deletes selected connection.
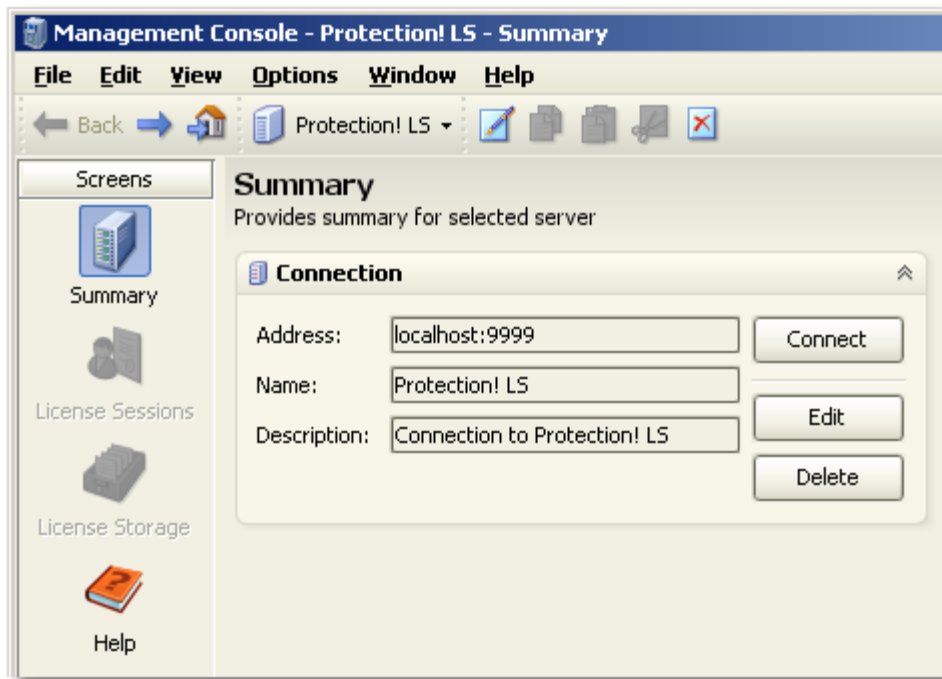
Figure 4: Connection Summary

It is possible to change the current connection using the connections drop-down on the application toolbar or via the *View | Connection* menu.

3.2 Managing Server Users

The Licensing Server Management Console provides the ability to manage Licensing Server system users. Only users with administrative privileges are able to maintain Licensing Server users. There are 3 (three) type of server users:

• *Administrator* – Has full Licensing Server and Management Console permissions

- *Manager* – Same permission as *Administrator* user but can not manage Licensing Server Users
- *User* – Can only view (read-only) Server activity and License Storage contents. Users with the *User* privileges can not disable Licensing Server licensing activities nor do they can shutdown the Licensing Server.

To maintain server users:
1. Choose the *Edit | Server Users* menu item
2. Choose appropriate option from the License Users Dialog:
    a. *New* – to create a new user;
    b. *Edit* – to modify parameters for the selected user;
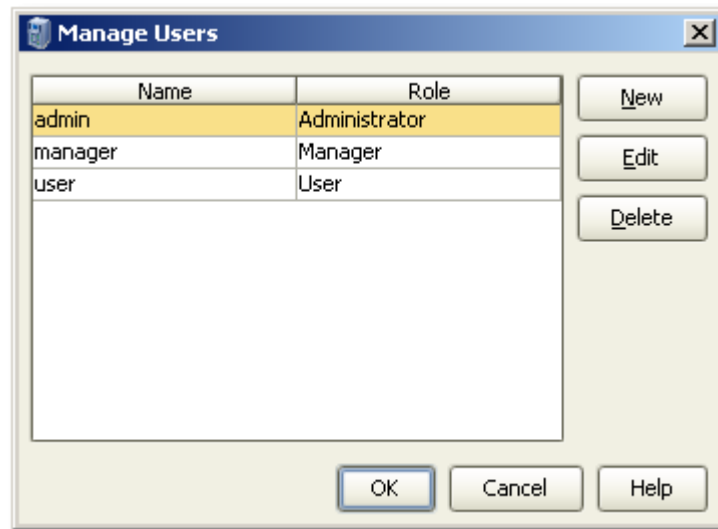   c. *Delete* – to delete the selected user.

Figure 5: Manage Server Users Dialog

There are several attributes that need to be specified for the Server user:
1. *Name* – specifies user login name;
2. *Password* – specifies user's password;
3. *Confirm Password* – confirms specified user's password;
4. *Role* – specifies user's role.

Figure 6: Manage Server Users Dialog

Passwords for Server Users expire after certain period of days (90 days by default). To change the value for that period - alter the `server.xml` file and change attributes of the authenticator elements:

- `passwordValidityPeriod` – password validity period in days (90 days by default). To allow non-expiring passwords – state 0 for this attribute or remove it completely.

There are strict rules for the passwords composing so a password must contain:
- At least one upper case letter
- At least one lower case letter
- At least one digit
- At least one special character: `# ? ! @ $ % ^ & * –`
- At least 8 characters

3.3 Summary Screen

This screen provides the ability to view Connection information, Licensing Statistics and the Licensing Server summary and to perform certain actions via the set of dashboards.

To activate the Summary screen select the Summary   icon on Shortcuts Side Bar or use the *View | Summary* menu item.

3.3.1 Connection Dashboard

This dashboard provides the summary for selection connection and provides the following actions via corresponding buttons:
1. Connect – connects to the specified Licensing Server;
2. Edit – Launches "Edit Connection" dialog for connection parameters modification;
3. Delete – Deletes the selected connection.

3.3.2 Licensing Statistics Dashboard

This dashboard provides the ability to view licensing statistics for the selected scope (e.g. for today, overall or several days ago).

The *Scope* combo box allows choosing the desired scope to view.

The requests statistics shows:
a. *Succeeded* – number of succeeded requests;
b. *Failed* – number of failed requests;
c. *Errors* – number of errors that occurred during requests processing.

The active session's statistics shows:
a. *Current* – number of active license sessions;
b. Max – maximum number of active license sessions.

It is possible to download detailed statistics reports for a selected scope using the *Download Reports* button. The following reports are available:
1. *Usage per User* –license utilization per user over time.
2. *Total Users (Succeeded)* - total number of users utilizing licenses within a timeframe.
3. *Total Users (Failed)* - total number of users failed to utilize licenses within a timeframe.
4. *Statistics* – report for all licensing activities.

Most of columns in those reports are self-explained. Some columns may need more explanation:

1. Operation – shows operation performed by Licensing Server. Possible values are:
   a. *LOCK* – license lock was obtained (aka. checked-out).
   b. *UNLOCK* – license lock was released (aka. checked-in).
   c. *OBTAIN* – license was obtained (aka. downloaded).
2. *Result* – result of the operation. Possible values and corresponding codes are:
   a. *OK = 1* – operation was successful.
   b. *Error = 2* – operation was failed
   c. System Error = 3 – operation was failed with unexpected system error
   d. Expired = 5 – license session was expired because a client application was terminated or a computer just went to sleep.
   e. *Access Denied = -1* – operation was failed because access was denied by administrator rule.

3.3.3 Server Control Dashboard

This dashboard optionally shows information about the status of the Licensing Server in the yellow message box on the top and provides the ability to shut down the Licensing Server and to disable/enable licensing activities.

Shutting Down the Licensing Server

The Shutdown button allows to remotely shutdown the Licensing Server. After clicking the Shutdown button the current connection will be closed and the Licensing Server shutdown will be initiated.
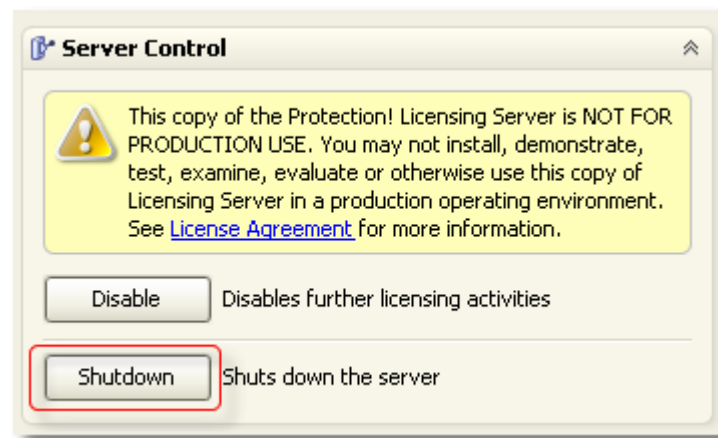

Figure 7: Shutting Down the Licensing Server Instance

Note

Current implementation has no ability to start/re-start the Licensing Server after the shutdown process has been initiated.

Disabling/Enabling Licensing Activities

In some cases the Administrator needs to temporarily disable all licensing activities without shutting down the Licensing Server. This is especially useful if the Licensing Server maintenance needs to be performed such as user and user group's management, access control list management, etc. The Disable button allows disabling all licensing activities.
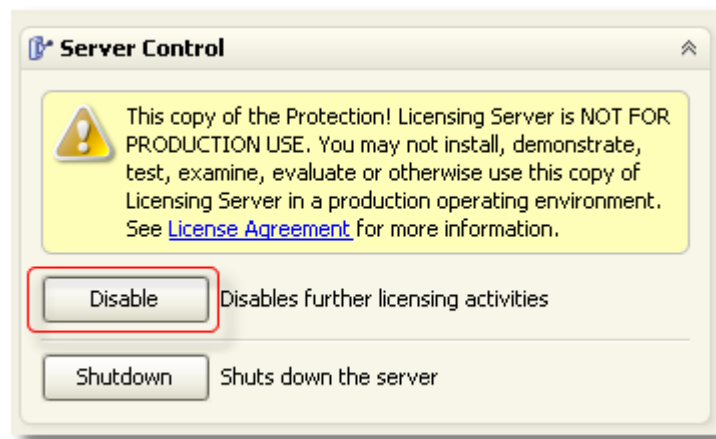
Figure 8: Disable all licensing activities

After pressing the Disable button the confirmation dialog will be shown on the screen providing optional reason why all licensing activities will be disabled.
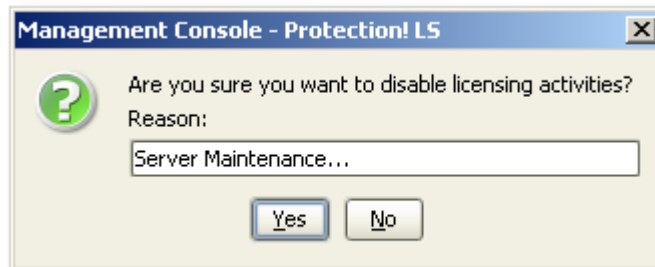

Figure 9: Confirmation dialog with optional reason

Any new license lock requests from the clients would not be fulfilled while the Licensing Server in the disabled state. Any client application trying to acquire license lock from the licensing server will see the following dialogs:
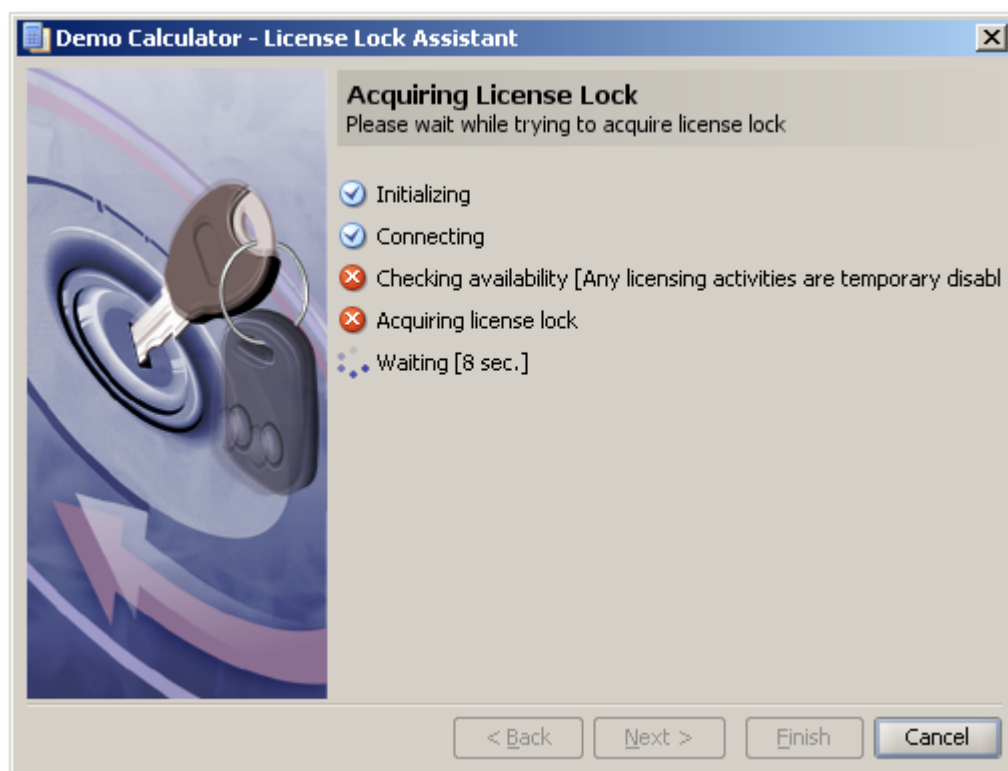
Figure 10: License Lock assistant – unable to acquire lock
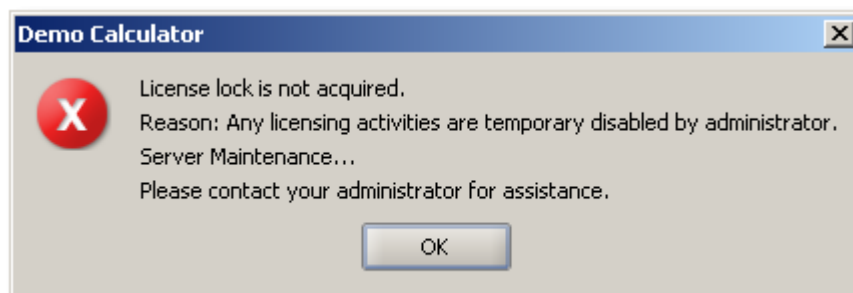
Figure 11: Client application message

Note	The disabled state of the Licensing Server is not persisted and will be cleared during the Licensing Server re-start.

When Licensing Server is in the disabled state the Enable button allows restoring licensing activities. After clicking the Enable button all licensing activities will be immediately enabled.

Figure 12: enable all licensing activities

3.2.4 Server Summary Dashboard
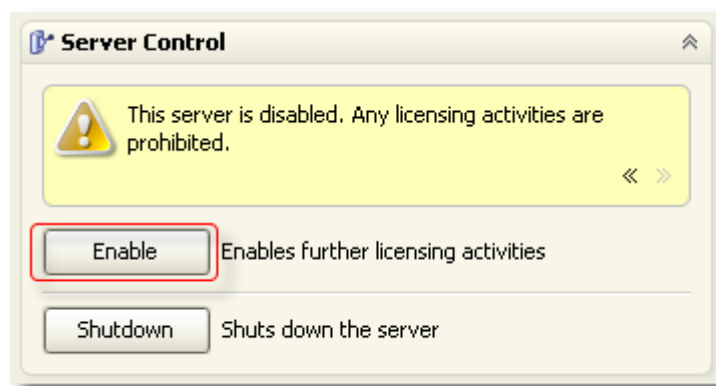
This dashboard shows information about resources used by the Licensing Server. This information includes:
1. Up time;
2. CPU usage;
3. Threads statistics;
4. Memory statistics;

3.4 License Storage Screen

The License Storage screen provides the ability to maintain licenses available for distribution and to allocate them to the users. To activate the screen select the License Storage icon on Shortcuts Side Bar or use the *View | License Storage* menu item.

3.4.1 Maintaining License Storage

The Licensing Server Management Console provides the ability to maintain the License Storage. License Storage is designed to store products and their licenses in one uniform location (container) and optionally allocate license users and/or license user groups with appropriate licenses. The License Storage allows holding and storing an unlimited number of products and their licenses.

Note	Products must first be imported into the License Storage before the licenses for these products could be imported.

To import products:
1. Choose the *File | Import Product* menu item;
2. Navigate to the supplied by vendor product file (e.g. `DemoCalc.product`) and select it;
3. Press OK to import selected product.

Tip	Product (e.g. `DemoCalc.product`) is actually a Launcher for corresponding product. To get it - build a Launcher file (using Protection! Developer Control Center or Protection! Builder Utility) for a Launcher Builder Configuration with Encryption turned OFF.

To view imported products and their versions use the Products dialog that can be shown using the *View | Products* menu item.

To import licenses:
1. Choose the *File | Import Licenses* menu item;
2. Navigate to supplied by vendor license(s) and select it ('em);
3. Press OK to import selected license(s).

Details for selected license could be seen on the License Tab at the right of the screen.
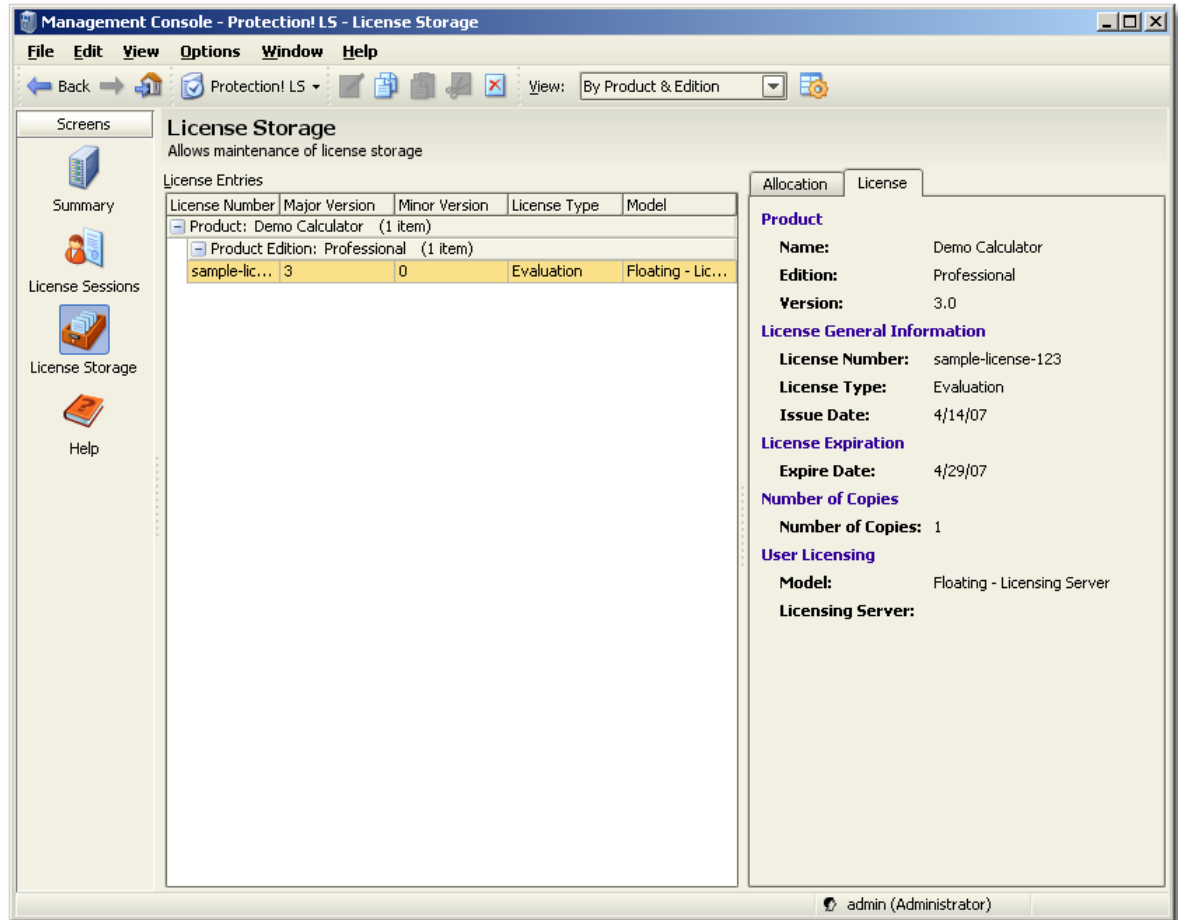

Figure 13: License Storage Screen – License Information

3.4.2 Maintaining LDAP Connection

To allow allocation of licenses to the users fetched from a LDAP directory – the LDAP Connection should be specified via the *"Edit | LDAP Connection"*.
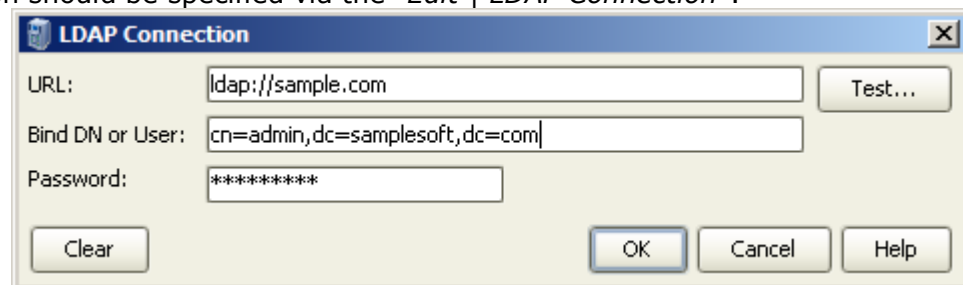

Figure 14: LDAP Connection Dialog

There the following connection attributes can be specified:

1. URL – an URL to a directory server that supports LDAP protocol. The URL must have `ldap` scheme e.g. `ldap://sample.com:389`. If port is not explicitly specified – port 389 will be used by default.
2. Bind DN or User – bind DN or user name to authenticate the connection. Note: only BASIC authentication is currently supported.
3. Password – password.

3.4.3 Maintaining License Users and License User Groups

The Licensing Server Management Console provides the ability to maintain list of license users and license user groups that could be associated with the licenses in the License Storage.

Maintaining License Users

1. Click the *Edit | License Users* menu item;
2. The License Users dialog will be shown on the screen displaying the list of available license users;
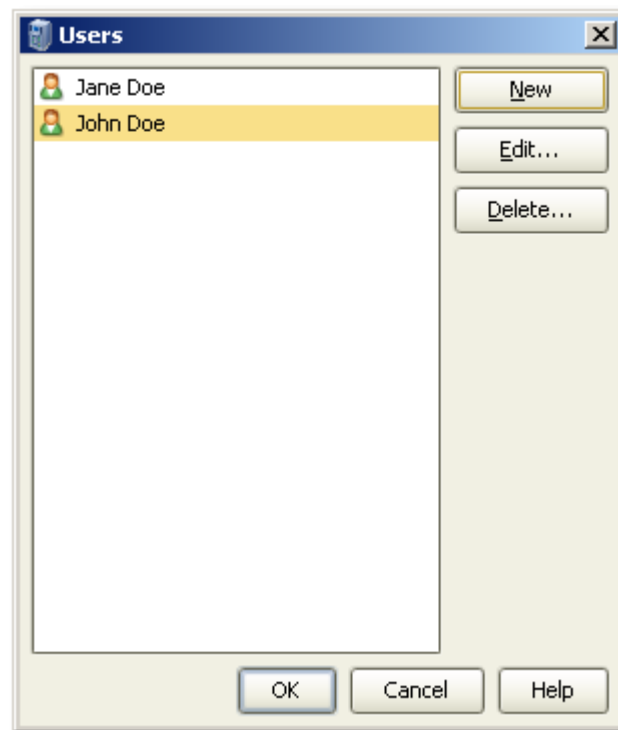
Figure 15: License Users Dialog

3. Choose the appropriate option from the License Users Dialog:
   - a. *New* – to create new user;
   - b. *Edit* – to modify parameters for selected user;
   - c. *Delete* – to delete selected user.

There are several attributes that could be specified for the license user:

1. *Name* – specifies user descriptive name;
2. *User Name* – specifies user's system name. In general this would be the same name (sometimes referred to as "User Id") that is used by the user as identification and is required when logging on to a computer system;
3. *Host* – specifies user's computer name;

4. *IP Address* – specifies user's private IP address. IP addresses are written in "dotted decimal" notation, which is four sets of numbers separated by periods; for example, `192.168.1.1`. Currently only the traditional IP address (IPv4) is supported;
5. *Member Of* tab – provides ability to associate the user with the License User Group.

Figure 16: Add User Dialog

Maintaining License User Groups

To maintain License User Groups - Choose *Edit | License User Groups* menu item. The License User Groups dialog will be shown on the screen displaying the list of available license user groups.
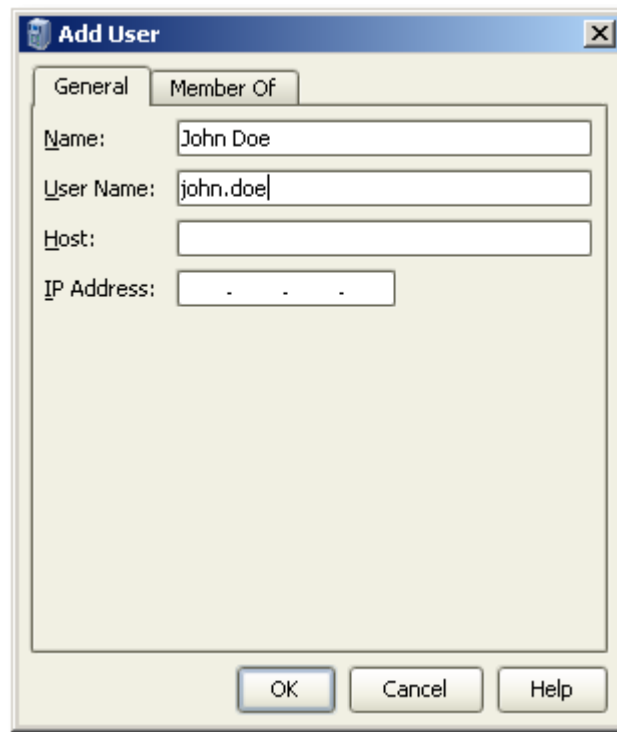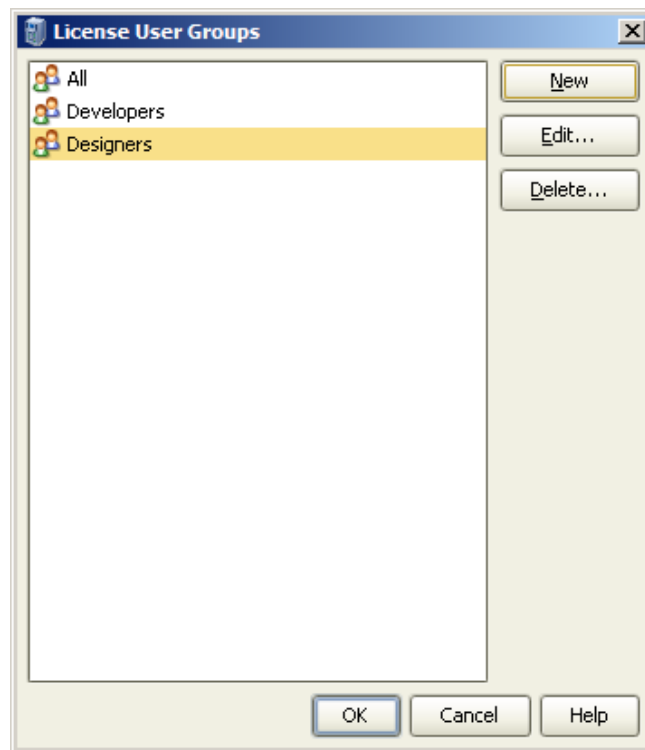
Figure 17: License User Groups Dialog

There it is possible to make the following actions:
a. *New* – to create new group
b. *Edit* – to modify parameters for selected group
c. *Delete* – to delete selected group

There are several pages of attributes that could be specified for the license user group:

General
The following attributes can be specified on the General page:
1. *Name* – specifies user group descriptive name;
2. *Host* – specifies range of user's computer names virtually belonging to the group; Usual * and ? wildcards can be used there. For example, specifying the * as host name allows defining the user group that virtually contains all users from any hosts;
3. *IP Address* – specifies range of private IP address of users virtually belonging to the group. IP addresses are written in "dotted decimal" notation, which is four sets of numbers separated by periods; for example, 192.168.1.1. Currently only the traditional IP address (IPv4) is supported. Usual * and ? wildcards can be used there;
4. *Default* – specifies whether the User Group should be default one. If default group is specified – licenses will be automatically allocated to that group during import process.
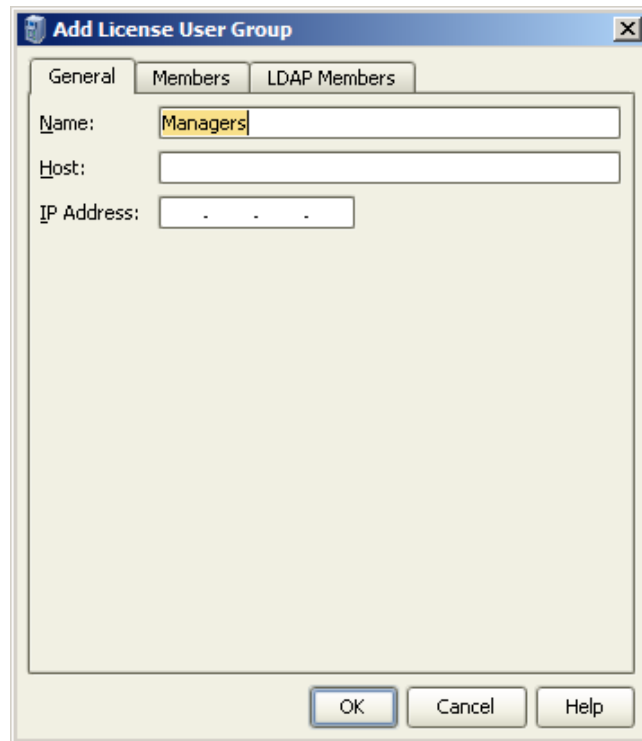
Figure 18: Add License User Group | General

Members
The Members page provides the ability to associate users with the License User Group. There it is possible to add the users to the group or delete them.
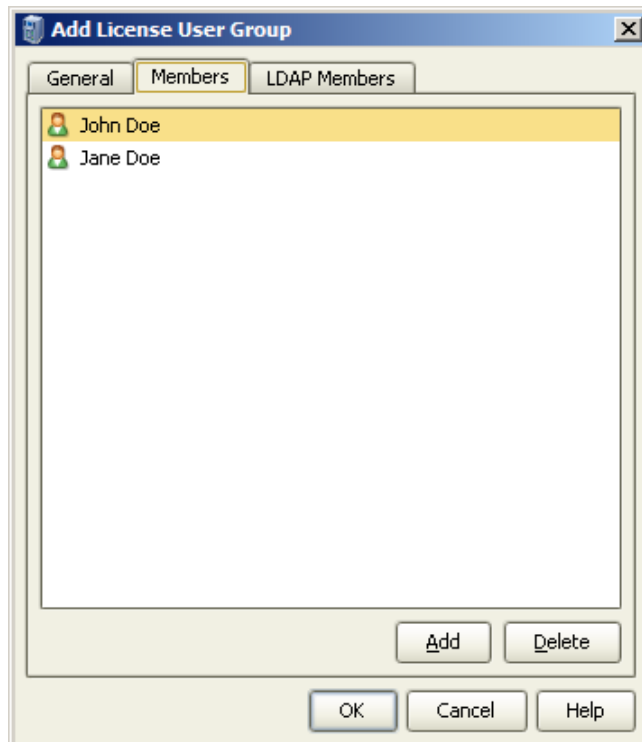

Figure 19: Add License User Group | Members Page

LDAP Members

The LDAP Members page provides the ability to associate the users fetched from a LDAP directory with the License User Group. Note: to make this working – LDAP Connection must be configured first (Edit | LDAP Connection).

The following attributes can be specified in the LDAP Members page:
1. *DN* and *Filter* properties to point to desired set of users.
2. *Search Subtree* defines whether lookup for users should go down sub-tree.
3. *Property Mapping* to define how to map the *"User Name"* (login), *"Host"* and *"IP Address"* to corresponding fields in LDAP objects. At least one mapping must be specified.
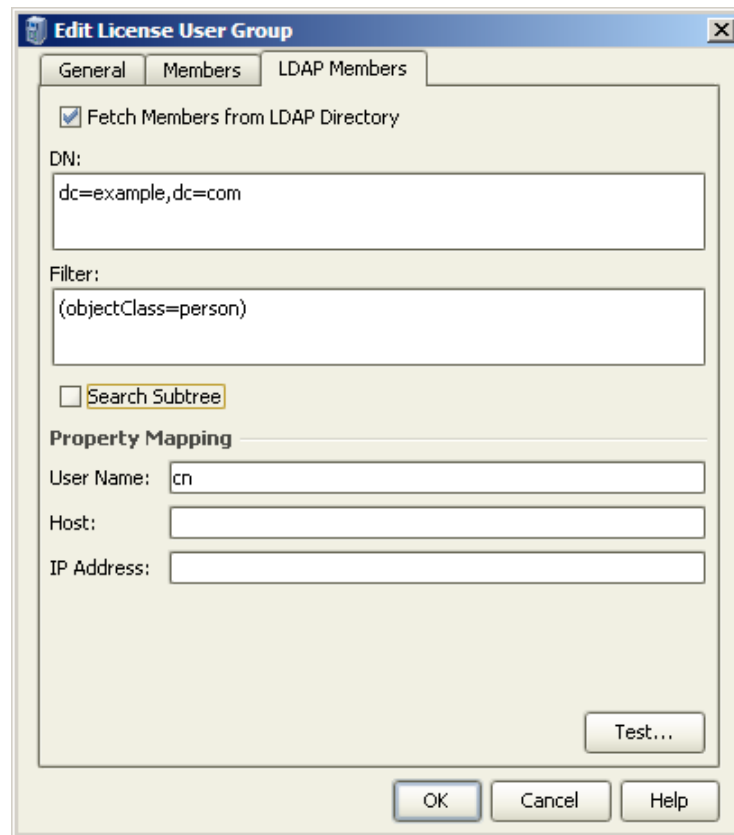
Figure 20: Add License User Group | LDAP Members Page

It is possible to test if LDAP users are correctly fetched by pressing the *Test* button.

Note

It is recommended to keep the *"Search Subtree"* option turned OFF if possible. You should turn this option ON only when it is really required and with a caution.

 If the *"Search Subtree"* option is turned ON and LDAP directory is very complex - it is possible to reach objects (while going down sub-tree) to which the authorized user have no access. In that case it would be expected that such inaccessible objects will be simply skipped from the traversal and everything will continue iterating. Unfortunately, it seems there is an issue in the JRE that breaks everything by throwing a NPE.

3.4.4 Allocation of Licenses for Specific Users and/or User Groups

The Licensing Server Management Console provides the ability to allocate license users and license user groups to the licenses in the License Storage that allows defining which license(s) are available to which users.

To associate a license user and/or license user group with the license:
1. Select a specific license in the License Entries Grid;
2. Click the Add User button ;
3. Select appropriate license user(s) and/or license user group(s) from the Add Users / Groups dialog. Multiple selections could be done by either holding a *Control* or a *Shift* keyboard key.
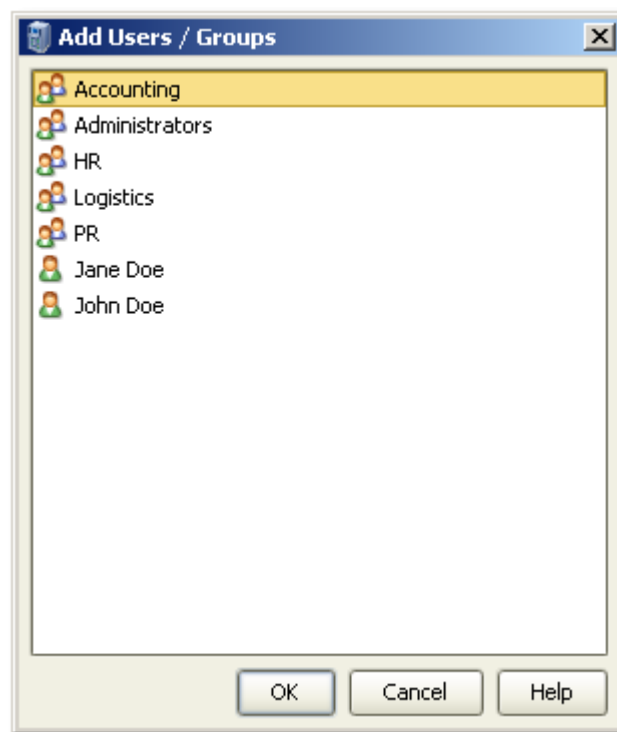
Figure 21: Associate License Users / Groups with the license

To remove the association between the license users/groups and the license select desired users/groups and press the button.
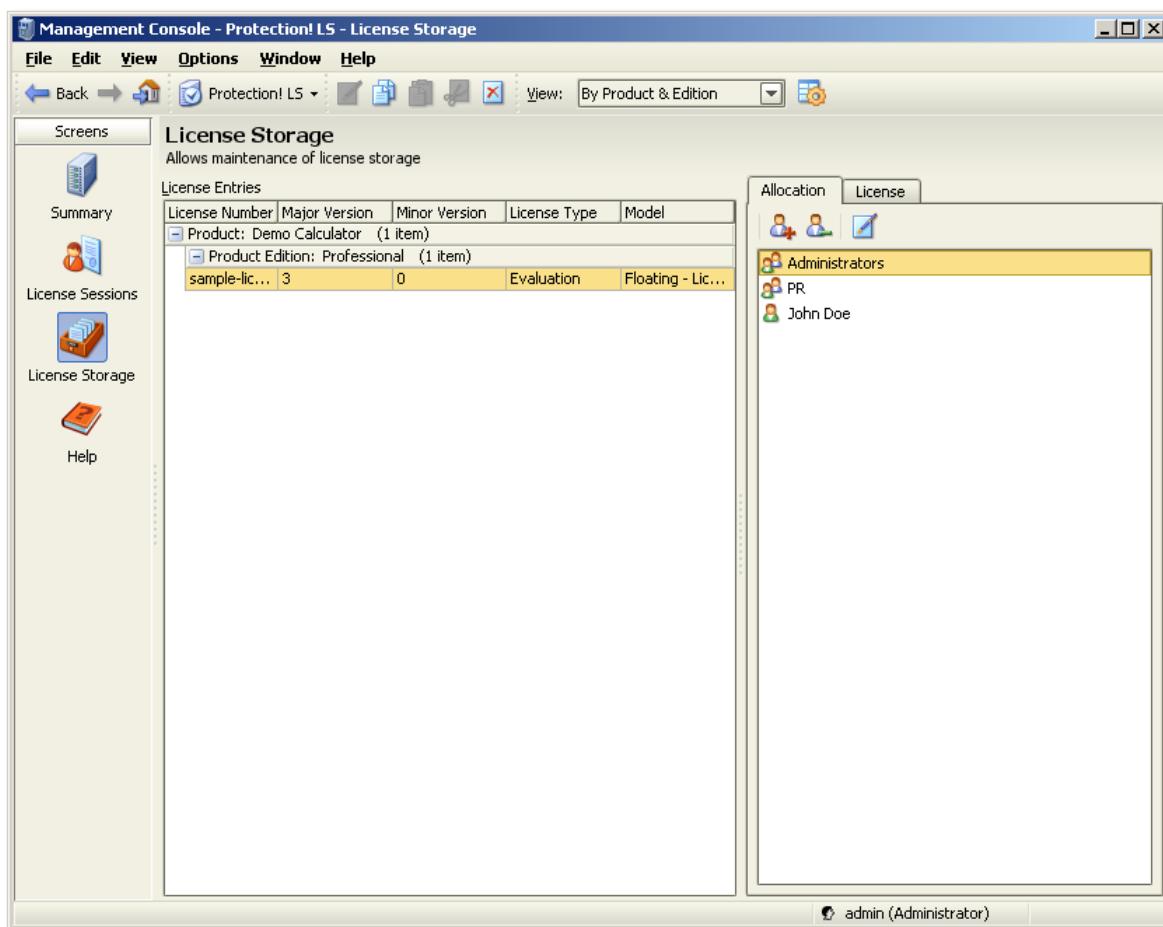
Figure 22: License User and User Group Allocation

3.5 License Sessions Screen

The Licensing Server Management Console provides the ability to visually monitor all currently running license sessions. To activate the screen select the License Sessions

icon on Shortcuts Side Bar or use the *View | License Sessions* menu item.

3.5.1 Monitoring License Sessions

The License Sessions Grid shows all the currently running License Sessions. It is possible to view details for a selected License Session using the Details Pane.
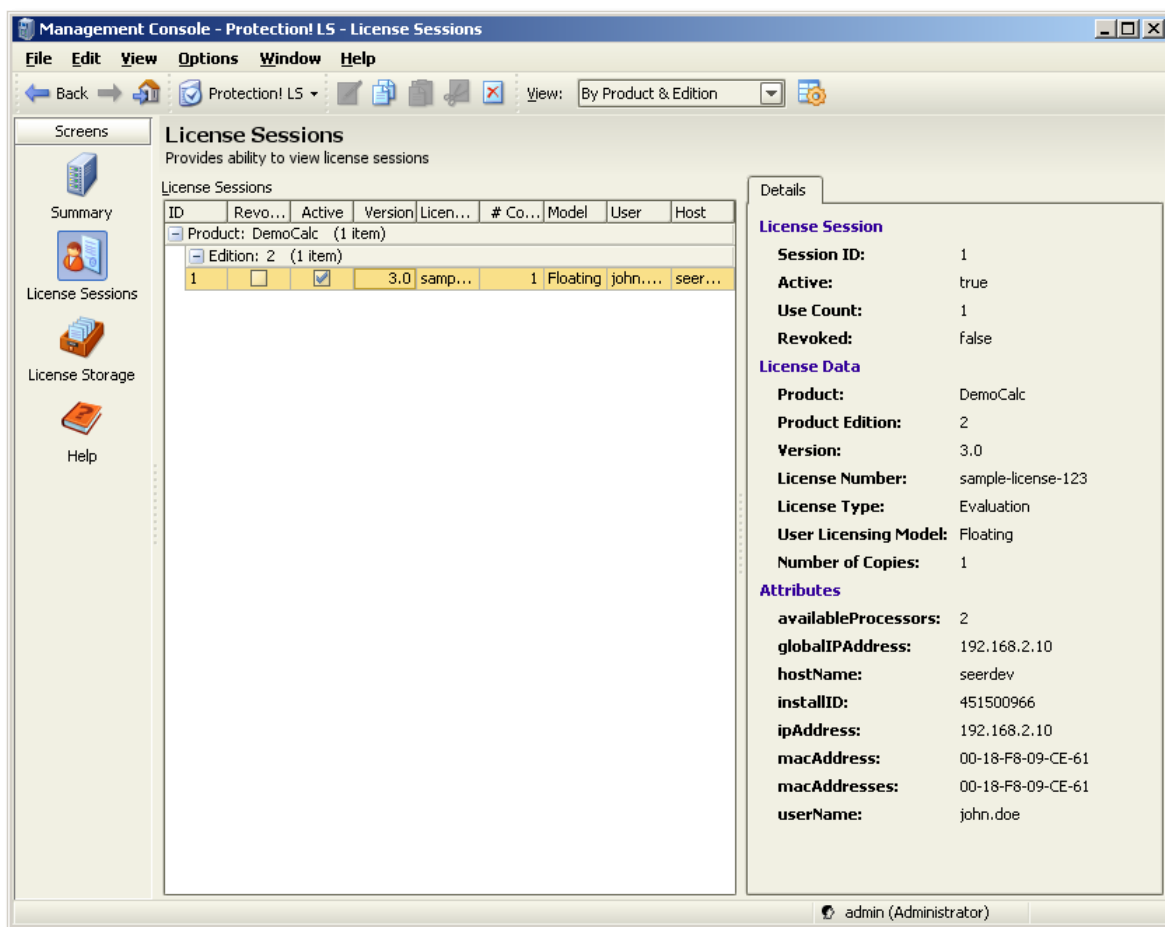
Figure 23: License Sessions Screen

3.5.2 Revoking License Sessions

The Licensing Server Management Console provides the ability to revoke any of the currently running license sessions.

Note	If the License Session to be revoked is in use – the actual revocation will be replaced by a corresponding revocation request. If revocation request was placed then the corresponding License Session will continue to run until request is handled by the client application. It is up to client application implementation how to handle such requests. They could be accepted (leading actual revocation) or simply ignored e.g. based on the users' input.

To revoke license sessions:
1. Navigate to the desired license session in the License Sessions Grid;
2. Click Delete Icon ❌ on the Management Console toolbar
3. In the confirmation dialog provide optional reason why specific license session needs to be revoked.
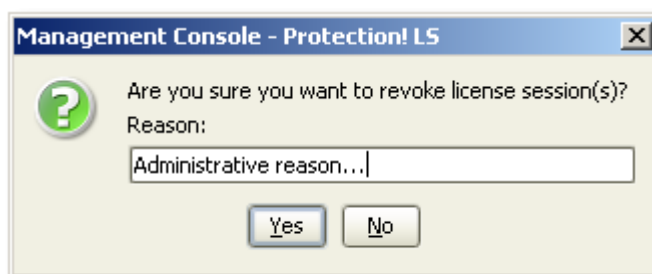
Figure 24: Revoke License Session Confirmation Dialog

The client application for which the license session has been revoked would receive the following notification:

Figure 25: License Sessions Screen

3.6 Maintaining Access Control List

Protection! Licensing Server provides functionality of the Access Control List (ACL) through the common concept of the white and black lists. An access control list is an access matrix, containing the access rights of various users, systems, etc to the Licensing Server processes. Licensing Server's white list is a list of users, systems, etc that the Licensing Server is configured to accept. All licensing activities from the entries in the white list are allowed. White list is a contrast with the blacklist where given users, systems, etc are considered off limits and no licensing activities are allowed to any entry in the blacklist.

Licensing Server Management Consol provides administrator with the ability to maintain several different access control lists:
1. *Global* – an access control list that is applicable to any licensing activity;
2. *Lock License – Named User* – an access control list that is applicable only to licensing activities that are associated with the requests to obtain license lock for the licenses with the Named User Licensing models only;
3. *Lock License – Floating User* – an access control list that is applicable only to licensing activities that are associated with the requests to obtain license lock for the licenses with the Floating User Licensing models only;
4. *Obtain License* - an access control list that is applicable only to licensing activities that are associated with the requests to obtain a license from the Licensing Server.

Each of the above access control lists has its respective white lists and black lists. To maintain either white list or black list:
1. Choose desired ACL using the *Edit | Access Control List* menu;
2. Choose the *White List Conditions* or *Black List Conditions* menu item;

3. In the "Conditions" dialog create either new condition or new complex condition. Complex conditions allow using Boolean AND/OR logic to build hierarchical condition tree.
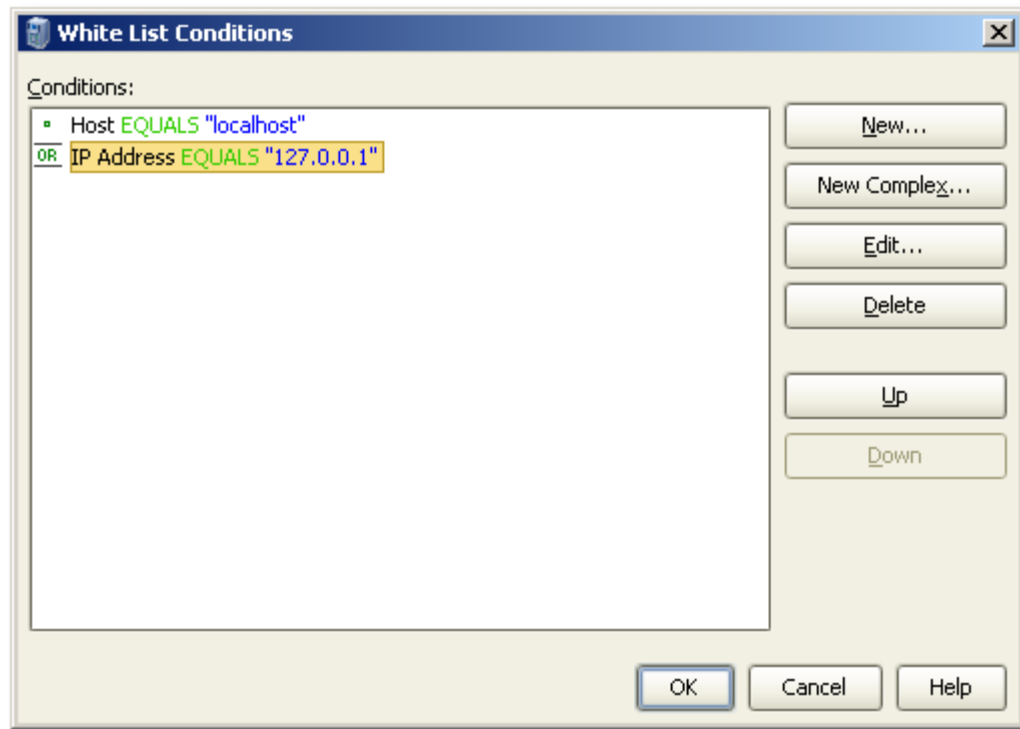
Figure 26: White List Conditions dialog

The New Condition dialog provides the following parameters to create condition:

1. Field – supports the following parameters:
   a. Host
   b. User Name
   c. Private IP Address
   d. Public IP Address
   e. MAC Address
   f. MAC Addresses
   g. Install ID
   h. Available Processors
2. Operation – supports the following parameters:
   a. Equals
   b. Not Equals
   c. Has Value
   d. Has No Value
   e. In List
   f. Not In List
   g. Like
   h. Not Like
   i. Starts With
   j. Not Starts With
3. Value – specifies value or a list of values that will be checked.

Figure 27: New Conditions Dialog

Figure 28: New Complex Conditions Dialog

Using Management Console via Command Line

The Licensing Server Management Console provides ability to manage Licensing Servers via command line interface. Usage as follows:

```
ManagementConsole [options]
```

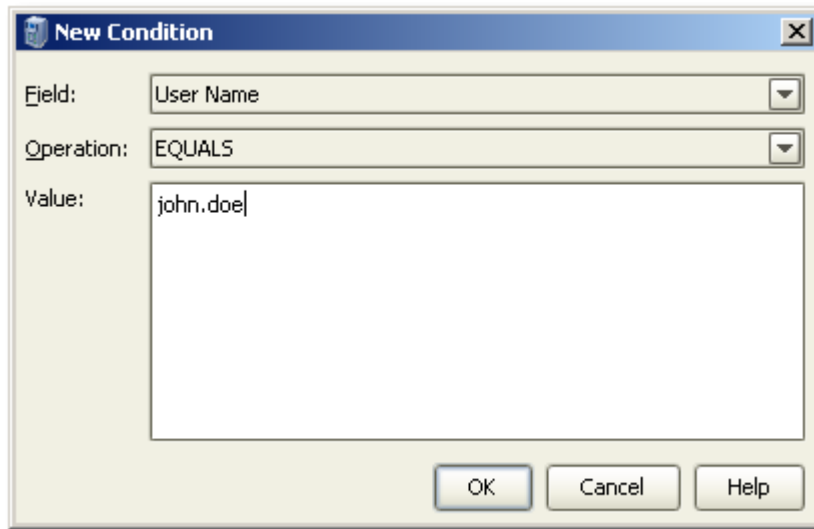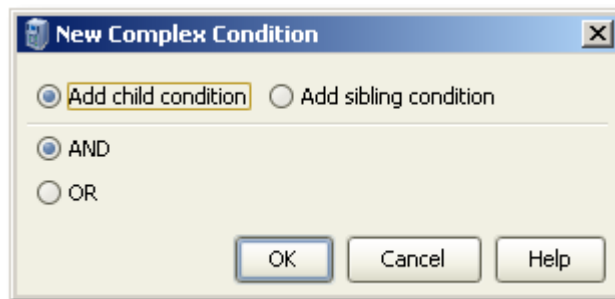|	
`-connection <connection>`	Optional connection to work with. It can be either name of an existing connection or a connection in the form of `'user:password@host:port'`. If not specified - the default connection will be utilized. To configure connections – launch Management Console in the GUI mode.
`-actions <actions>`	Actions to perform in JSON format. `<actions>` can be either: • Path to a file, which contains actions' data. • String with action's data formatted as one line. Note all the quotes must be escaped with a \ character. For example: `[{\"action\":\"enableLicensingActivities"\"}].`
`-help`	print help message

> **Note**
>
> To see console output and use actions supplied as a JSON string the Management Console must be started via direct Java invocation:
>
> `java –jar ManagementConsole.jar [options]`

Sample calls would be:

```
ManagementConsole -connection "localhost" -actions "actions.json"
```

or

```
java –jar ManagementConsole.jar -actions
"[{\"action\":\"enableLicensingActivities\"}]"
```

JSON file or string should contain an array of actions to perform where each action is represented by a map defining action's properties. Action's map should contain the following entries:

Entry	Optional	Description
action	No	A name of action to perform
attributes	Yes	Map of action's attributes with contents depending

		on particular action

For example the following is a JSON that allows performing two actions - downloading a report and importing a license:

```
[
  {
    "action":"downloadReport",
    "arguments":{
      "reportID":"usagePerUser",
      "fileName":"usagePerUser.csv",
      "scope":0
    }
  },
  {
    "action":"importLicense",
    "arguments":{
      "fileName":"DemoCalc.license"
    }
  }
]
```

The following are set of actions can be performed via command-line:
1. Downloading Reports
2. Importing Products
3. Importing Licenses
4. Revoking License Sessions
5. Enabling Licensing Activities
6. Disabling Licensing Activities
7. Shutting Down Licensing Server
8. Checking Licensing Server

4.1 Downloading Reports

The `downloadReport` action allows downloading particular report and saving it to a local file. The following are `downloadReport` action entries:

Entry	Optional	Value
action	No	downloadReport
arguments	No	
reportID	No	Defines ID of a report to download. The following ID's are allowed: ● usagePerUser ● totalNumberUsers ● failedRequests ● licenseStatistics ● usagePerLicenseStatistics ● statistics ● requestsStatistics ● serverSummary Tip: it is possible to download JSON file for all available reports by pressing the "*Download*

Copyright © 2003-2021 jProductivity L.L.C. http://www. jproductivity.com

		Reports" button in the "*Summary Screen | Licensing Statistics*" dashboard and choosing the "*Download JSON File for Command-line*" item.
fileName	Yes	A file name to save a report to. If not specified – default file name for particular report will be utilized.
scope	Yes	Defines period in days to download report for. For example, 0 - today, 30 – last 30 days, etc.

Sample JSON for a `downloadReport` action would be:

```
[
  {
    "action":"downloadReport",
    "arguments":{
      "fileName":"usagePerUser.csv",
      "reportID":"usagePerUser",
      "scope":0
    }
  }
]
```

4.2 Importing Products

The `importProduct` action allows importing a product to Licensing Server database. The following are `importProduct` action entries:

Entry	Optional	Value
action	No	importProduct
arguments	No	
fileName	No	Name of a file containing a product to import

Sample JSON for an `importProduct` action would be:

```
[
  {
    "action":"importProduct",
    "arguments":{
      "fileName":"DemoCalc.product"
    }
  }
]
```

4.3 Importing Licenses

The `importLicense` action allows importing a license to Licensing Server database and optionally to allocate that license to a License User or a License User Group. The following are `importLicense` action entries:

Entry	Optional	Value
action	No	importLicense
arguments	No	

fileName	No	Name of a file containing a license to import
allocateToUserGroup	Yes	Name of a License User or User Group to allocate a license to. If not specified – the license will be allocated to a default License User group (if present).

Sample JSON for an `importLicense` action would be:

```
[
  {
    "action":"importLicense",
    "arguments":{
      "fileName":"DemoCalc.license",
      "allocateToUserGroup":"Testers"
    }
  }
]
```

4.4 Revoking License Sessions

The `revokeLicenseSession` action allows revoking a license session identified by the *License Number* for a user identified by the *User Name*, *Host Name* and *IP Address* attributes. Note: at least one attribute that identifies a user must be specified. If more than one user's attributes are specified – they must identify the same user. The following are `revokeLicenseSession` action entries:

Entry	Optional	Value
action	No	revokeLicenseSession
arguments	No	
licenseNumber	No	License Number of a license to be revoked
userName	Yes*	User name of the user
hostName	Yes*	Host name of the user
ipAddress	Yes*	IP address of the user
reason	Yes	Reason for revocation

Sample JSON for a `revokeLicenseSession` action would be:

```
[
  {
    "action":"revokeLicenseSession",
    "arguments":{
      "licenseNumber":"DemoCalc 577255",
      "userName":"hellen45",
      "reason":"The license is required for another user"
    }
  }
]
```

4.5 Enabling Licensing Activities

The `enableLicensingActivities` action allows enabling licensing activities if they were previously disabled. The following are `enableLicensingActivities` action entries:

Entry	Optional	Value
action	No	enableLicensingActivities
arguments	Yes	

Sample JSON for an enableLicensingActivities action would be:

```
[
  {
    "action":"enableLicensingActivities"
  }
]
```

4.6 Disabling Licensing Activities

The disableLicensingActivities action allows temporary disabling licensing activities. The following are disableLicensingActivities action entries:

Entry	Optional	Value
action	No	disableLicensingActivities
arguments	Yes	
reason	Yes	A reason for disabling of licensing activities

Sample JSON for a disableLicensingActivities action would be:

```
[
  {
    "action":"disableLicensingActivities",
    "arguments":{
      "reason":"The server is on maintenance for about an hour"
    }
  }
]
```

4.7 Shutting Down Licensing Server

The shutdown action allows to shutdown a Licensing Server. The following are shutdown action entries:

Entry	Optional	Value
action	No	shutdown
arguments	Yes	

Sample JSON for a shutdown action would be:

```
[
  {
    "action":"shutdown"
  }
]
```

4.8 Checking Licensing Server

The `check` action allows to determine if a Licensing Server is up and running. The following are `check` action entries:

Entry	Optional	Value
action	No	check
arguments	Yes	

Sample JSON for a `check` action would be:

```
[
  {
    "action":"check"
  }
]
```

A.1 Fetching LDAP License Users

If you experience troubles fetching LDAP license users then you can try to utilize alternative (workaround and less efficient) implementation. This implementation can be turned ON by altering the `bin/server.xml` configuration file and specifying `alternativeLDAPFetch="true"` attribute of the `<module id="licenseUsers"/>` element:

```
<module id="licenseUsers"
        fileName="${config}/licenseUsers.xml"
        alternativeLDAPFetch="true"/>
```

Note: Licensing Server must be restarted to apply changes made to the `server.xml`.

A.2 Encryption and Security Protocols

Protection! Licensing Server makes use of the following encryption methods and security protocols:
1. **RSA-1024** for communication via HTTPS or RMI over SSL
2. **SHA-512** with salt for password digests
3. **PBEWithMD5AndDES** with salt for encryption of not-so-important things
4. **RSA-512** for licenses encryption

A.3 Making Custom Installers

It is possible to make custom installers for Protection! Licensing Server to bundle the licenses, tweak configuration, add additional install actions etc. If you need to do this – please contact us to get the [install4j](#) project.

If you are making your own installer from scratch or you have forked our installer some time ago – please check it to ensure:
1. The launcher for Licensing Server contains the following VM option:

   ```
   -Xmx512m
   ```

   It is also feasible to expose VM options for that launcher so the users can adjust them if needed.

2. The launcher for Management Console contains the following VM option (conditionally, for Java 17 and later):

   ```
   --add-exports java.desktop/sun.awt=ALL-UNNAMED
   ```

As part of the continuing effort to improve our product, we welcome your comments, suggestions and general feedback regarding the product.

If you have questions about Protection!, please feel free to contact us for further information at protection@jproductivity.com, or visit our site at: http://www.jproductivity.com.

If you discover any issues or defects in Protection! - please send the description of them to protection@jproductivity.com.