SERENA®

RELEASE AUTOMATION 4.1

Powered by Nolio

Installation and Administration Guide

Contents

Revisions

Ver.	Date	Description
4.1	Aug 2012	Updated Server varfile Template information for installing with MS Sql Windows authentication.
		Added MS SQL 2012 to list of supported databases.
		Updated note that application installed without license file is evaluation copy.

Preface

The *Serena Release Automation Installation and Administration Guide* accompanies *Serena® Release Automation* V4.1 release.

About this Document

This document provides the necessary information for:

- Installation and configuration of each of the modules that comprise Serena Release Automation
- Ongoing Serena Release Automation administration

Intended Audience

This documentation is intended for the following audience:

- IT technicians who will manage the Serena Release Automation data centers and personnel who are responsible for the operational side of your organization's multi-tier applications, on all servers and in all data centers. These readers are in charge of Serena Release Automation installation.
- Serena Release Automation administrators who are responsible for the initial setup and the ongoing administration of Serena Release Automation.
- Serena Release Operations Center administrators and operators who need to understand how Serena Release Automation works.

Related Documentation

The Application Release Automation Documentation Set also includes the following:

- *Serena Release Automation* | *Actions Reference Guide* - Serena-installed action templates and categories
- *Serena Release Automation* | *Applications Support Matrix for Actions* - Supported applications for Serena-installed actions
- *Serena Release Automation* | Application Interface Guide - Using Serena APIs for Command Line Interface (CLI), REST and SOAP. Supersedes *Command Line Interface (CLI) Reference Guide* as of version 4.1.

- *Serena Release Automation* | *Custom Actions SDK* - Implementing a custom actions library
- *Serena Release Automation* | *Introduction to Release Automation* - Getting started with Release Automation
- *Serena Release Automation* | *Release Notes* - Details of new features, enhancements, resolved issues, and how to upgrade to latest version of Serena Release Automation
- *Serena Release Automation* | *Security Description* - Information on Serena Release Automation security and certificates
- *Serena Release Automation* | *System Upgrade Guide* - Upgrading to the latest version of Serena Release Automation
- *Serena Release Automation* | *User Guide* - Using Serena Release Automation to develop and execute automated release processes
- *Serena Release Automation Dashboard* | *User Guide* - Using Serena Release Automation Dashboard
- *Serena Release Operations Center* | *User Guide* - Using Serena Release Operations Center
- *Zero Touch Deployment* | *Plugin for Hudson/Jenkins Continuous Integration (CI) Server* - Installing and running the Serena Plugin for Hudson/Jenkins CI servers
- *Zero Touch Deployment* | *Plugin for Microsoft Team Foundation Server (TFS)* - Installing and running the Serena Plugin for Microsoft TFS

Technical Support

For comprehensive support options, contact Serena Support at http://support.serena.com.

Chapter 1

Introduction

In This Chapter

The *Serena Release Automation Installation and Administration Guide* describes the installation process of Serena Release Automation version 4.1.

This chapter introduces Serena Release Automation's architecture and details system requirements and platform compatibilities for each of the Serena Release Automation modules.

Serena Release Automation Architecture

Serena Release Automation is a platform for the automation of complex, multi-tier applications. It includes the following modules:

- **Serena Data Management Server** (referred to as **Data Management Server**) - to which clients connect in order to access Serena Release Automation
- **Serena Execution Server** (referred to as **Execution Server**)
- **Serena Agents** (referred to as **Agents**)
- **Serena Client UI** (referred to as **Client UI**)

Execution Servers and **Agents** are the modules that work in the background to enable the management and automation of such multi-tier application environments.

When a process is executed in Serena Release Automation, the appropriate data and instructions are channeled from the **Data Management Server** to the **Execution Server** that manages a specific Data Center. For the purpose of managing the data center activities, the Serena Release Automation also includes any number of **Agent** modules, each of which controls a specific physical server involved in the execution of a process. Each **Agent** receives the appropriate instructions passed to it by the **Execution Server** and implements them on the server to which it is linked.

The **Client UI** is the user interface application that connects to and interacts with the **Data Management Server**. It is downloaded and installed automatically upon accessing the Serena Release Automation Web server. It is installed using the Java Web Start technology which requires administrative user privileges on the client machine.

System Requirements

The system requirements depend on the Serena Release Automation components being installed.

Hardware Requirements

In the case where Serena components are installed on the same machine, the MEMORY and DISK-SPACE values should be summed.

The following values are the minimum requirements for the installation phase.

Note: Based on your system configuration and load, additional memory, disk space, and processors might be required.

Component	RAM	Processors	Disk Space (See Note)
Complete Installation	2 GB	2 CPUs	2 GB
Data Management Server	1 GB	1 CPU	1 GB

Component	RAM	Processors	Disk Space (See Note)
Complete Installation	2 GB	2 CPUs	2 GB
Execution Server	512 MB	1 CPU	300 MB
Agent	256 MB	1 CPU	300 MB
Client UI	512 MB		
Release Deployment Dashboard	512 MB		

Note: The above disk space requirements represent the minimum for the installation phase. For the operational phase, you will need to consider additional disk space where file transfers are being made. The additional disk space required is calculated at 1.5 times the total size of files transferred per hour.

Recommended Hardware Configuration

For systems which need to a handle load of 30 concurrent steps per second, it is recommended that customers split the Serena Data Management service and its database onto separate machines. For such a configuration, it is recommended to allocate at least 2 CPUs and 4 GB RAM for the Data Management server and to allocate 4 CPUs and 4 GB RAM for the Database server.

Systems that need to manage less than 30 concurrent steps can allocate fewer resources.

For low-level load systems, an All-In-One configuration (Data Management, Execution Server and Serena Agent) running on a single machine with 2 CPUs and 2 GB RAM will suffice.

Note: The above recommended requirements depend heavily on the specific configuration and load of the Serena system.

Platform / Release Automation Component Compatibility Matrix

The following table identifies the platforms which are supported for each Release Automation component:

Platform/Component Compatibility Table

Platform	Release Automation Component			
	Serena Data Management (Center)	Serena Execution Server	Serena Agent	Client UI
AIX 6.1			Supported (Requires IBM Java JRE 6)	

Platform	Release Automation Component			
	Serena Data Management (Center)	**Serena Execution Server**	**Serena Agent**	**Client UI**
Linux (Red Hat Enterprise Linux 5.2 or higher, CentOS, Fedora, Ubuntu, SUSE, Oracle Linux 5 update 6)	Supported (32 and 64bits Kernel 1.6 or higher) **Note**: 32-bit application support is required for installer.	Supported (32 and 64bits Kernel 1.6 or higher)	Supported (32 and 64bits Kernel 1.6 or higher)	Supported (JRE 6 update 16 or higher)
zLinux s390 SUSE 10 ES			Supported (s390x - 64bits *only)*	
Solaris 9 x86 Solaris 9 SPARC			Supported	
Solaris 10 x86 Solaris 10 SPARC	Supported	Supported	Supported	
Solaris 11 x86	Supported	Supported	Supported	
Windows 2000			Supported	Supported (JRE 6 update 16 or higher)
Windows 2003			Supported (32 and 64 bits)	Supported (JRE 6 update 16 or higher)
Windows 2003 R2	Supported (32 and 64 bits)	Supported (32 and 64 bits)	Supported (32 and 64 bits)	Supported (JRE 6 update 16 or higher)
Windows 2008 Windows 2008 R2	Supported (32 and 64 bits)	Supported (32 and 64 bits)	Supported (32 and 64 bits)	Supported (JRE 6 update 16 or higher)
Windows 7 Windows VISTA Windows XP			Supported	Supported (JRE 6 update 16 or higher)

Database Prerequisites

Serena Release Automation requires a database to store data, changes, and configurations.

During installation you will be able to select one of the following database vendors:

- MySQL
- SQL Server
- Oracle

Note: If you select SQL Server, the Data Management Server must be also installed on a Windows machine.

Supported database versions are as follows:

MySQL	Oracle	MS SQL Server
Versions: 5.1.30 and higher **Note**: InnoDB Storage Engine must be enabled.	Versions: 10g and higher	Versions: SQL 2005 and higher, including SQL 2012

MySQL Installation Requirements

A database user must be created for use by the Serena Release Automation application. The user has the option to decide which database user Release Automation will use. The database user may be **root** and it can be 'regular' user with the privileges described below.

The following scenarios are supported:

- Install with root user
  - No special configurations are required.
  - User will need to supply the connection details for the MySQL database.
- Install with non-root user

  Two options are supported:
  - A database will be created by the database administrator.
    - A database will be created by MySQL DBA with UTF-8 character set.
    - InnoDB storage is enabled in this MySQL instance.
    - A database user should be created with the following privileges:

      ```
      GRANT ALL PRIVILEGES ON <db_name>.* TO <username>@<hostname>
      IDENTIFIED BY '<password>';

      For example: GRANT ALL PRIVILEGES ON SERENA_DB.* to serena@localhost
      IDENTIFIED by 'serena';
      ```
  - A database will be created by Serena as part of the installation.

- A database user should be created by the MySQL DBA. The user should be granted privileges to create a database and its objects, as follows:

  ```
  GRANT CREATE, DROP, REFERENCES, EVENT, ALTER, DELETE, INDEX, INSERT,
  SELECT, UPDATE, CREATE TEMPORARY TABLES, LOCK TABLES, TRIGGER, CREATE
  VIEW, SHOW VIEW, ALTER ROUTINE, CREATE ROUTINE, EXECUTE ON *.* TO
  <username>@<hostname>;
  ```

During installation, connection details will be requested.

The existence of a running database instance is a mandatory requirement for Serena Release Automation.

Note: If the user selects to use an existing database, the installer will check if the database is empty. If the database is not empty, or contains a non-Serena schema, the installation will not complete and a relevant message will display.

Microsoft SQL Server Installation Requirements

Serena Release Automation requires a dedicated database for its use. A dedicated database can be created during the Release Automation installation or previously by the DBA.

For Serena to use MS SQL Server, the SQL Server configuration:

- TCP Protocol should be in Enabled mode
- SQL Server Browser Service should be started and in automatic startup mode

If the database is not empty, or contains a non-Serena schema, the installation will not complete and a relevant message will display.

In the database configuration phase of the Release Automation installation, the following will be required:

- Full MS SQL instance name: <HOSTNAME>\<INSTANCENAME>
- MS SQL DBA username which will be used to create the:
  - Connection to the MS SQL instance
  - Required dedicated database
  - Specific database login
- Dedicated database name
- Login name for database ownership

Windows and SQL authentication methods are supported during and after installation. During the installation, you will have the option to select the desired authentication method.

Note: Selecting Windows authentication method for the initial MS SQL instance connection implies that the logged-in user, running the installation, is permitted to log in to this instance and has DBA rights.

Alternatively, you may define the Serena Server service owner to have the required permissions to connect to the MS SQL Server instance.

For additional details, refer to *Using MS SQL Server as Database for Serena* (on page 115) in the *Installation and Administration Guide* Appendix.

Oracle Installation Requirements

Serena Release Automation requires a dedicated database user for its use. It is advisable that Serena database user will be assigned with a dedicated TABLESPACE.

The installation enables the creation of the database user and tablespaces. Alternatively, a DBA may create the dedicated database user and tablespace before the installation.

During the installation, the user will be verified. If the user exists and already holds database objects which are not part of Serena schema, the installation will not complete and a relevant message will display.

It is also recommended that the Oracle instance to be used by Release Automation will use a UTF encoding (UTF8 or UTF16).

The installation will require the following inputs for the creation of the Serena schema on Oracle:

1. Hostname or IP address of the Oracle Database
2. Oracle SID name, or Service name, as described in *Using Oracle as Database for Serena* (on page 123) in the Appendix.
3. Oracle Listener Port
4. Oracle Database user with DBA privileges, required for the initial connection verifying Oracle version and creating the Serena DB user, if required.

   The DBA privileged username and password will not be stored anywhere.
5. Tablespace name to be used by the Serena DB user

For additional details, refer to *Using Oracle as Database for Serena* (on page 123) in the Appendix.

Web Browsers and Flash Player

The Serena Release Deployment Dashboard is a Web-based application that requires Adobe Flash Player 10 to be installed on each client machine.

If Adobe Flash Player was not previously installed on the client machine, a message will display warning that the option to view portal and report graphs is disabled.

The supported Web browsers are:

- Microsoft Internet Explorer 8.0 and higher

  When using Serena Release Operations Center, IE 9 in compatibility mode is not supported.

- Google Chrome
- Mozilla Firefox 7.0 and higher

Ports and Protocols for Serena Release Automation

The following ports and protocols will be used by Serena Release Automation:

Ports and Protocols

Source / Protocol	Direction	Target	Target Port	Reason
Serena UI (HTTP/HTTPS)	From To	Release Automation	8080/8443	All communication between main Serena UI (and Dashboard) and Serena Server (default port)
Release Automation (HTTP/HTTPS)	From To	SES	8080/8443	Initial connection established between Release Automation and SES
SAG (TCP/SSL)	From to + Bidirectional	SES	6600	Transfer of process results back to SES at end of execution. Also used during file transfer during a process (default port)
SAG (TCP/SSL)	From To	SES	6900	If a SAG is installed on SES, recommended to open up traffic from all SAG to SES on 6900 (default port). In this case, all SAG to SES on 6600 must be bidirectionally enabled.
SES (TCP/SSL)	Bidirectional	SES	6600	If multiple SES exist AND SAG is connected to various SES to work together in a single process run, bidirectional communication between those SES is required.
Additional ports:				
SES	To	SAG	135 and 445	Remote agent installation on Windows platforms
SES	To	SAG	22	Remote agent installation on Unix via SSH
Release Automation	To	LDAP/ LDAPS	389/636 (default)	Importing and authenticating users from an LDAP source
Release Automation	To	SMTP	25 (default)	Sending e-mail notifications.

Source / Protocol	Direction	Target	Target Port	Reason
Release Automation	To	Database	DB listening port	If database resides on difference machine than the

Note: All port numbers are configurable. All source ports are random.

Required User Credentials for Installation of Serena Server

The following describes the user credentials required for installing Serena Server:

- Windows
  - For the installation process, the logged in user must have administrative privileges in order to enable creation of the required Serena Service.
  - The owner of the Serena Server service is configured by default to run using the Local System account.

    However, during the installation phase, the Serena Server service owner can be assigned to a different user. The Server service owner user should:
    - Be part of the Administrator group on the Windows machine
    - Have 'Log on as a Service' right
    - Have write, read and execute permissions on the Serena Release Automation installation folder
- Linux and Solaris
  - Serena Release Automation can be installed by any UNIX user that has permissions to create and update files under the installation directory.
  - Installation files should be extracted, using the installer, to a dedicated folder that will include only Serena Release Automation files.

    For example, if the target location for Serena Release Automation will be under /opt, the Serena Release Automation files will be extracted by the installer to /opt/SerenaReleaseAutomation. The UNIX assigned user should have write permissions to the /opt folder.

Considerations if Using Custom Actions Library

If you have developed a **Custom Actions** library using the *Serena Release Automation Custom Actions SDK*, you must ensure that the developed Custom Actions are stored in the **Serena Release**

Automation /customerActions folder and *not* the **/actionslib** folder as the **/actionslib** folder is overwritten during installations and upgrades.

Security

Serena uses its own certification file (JKS) to ensure secure communication between Serena components.

If required, self-signed certificates can be used. For instructions on how to create self-signed certificates and how to use them within Serena, refer to *Serena Release Automation Security Description*.

Chapter 2

Installing Serena Release Automation

In This Chapter

The installation of the Data Management Server and the Execution Server is performed through a single installation executable, **Serena Server Setup**.

The installation is performed by invoking the executable using a command line and passing the required inputs by one of the following methods:

- In the command line
- In a server input varfile

  The template for the server input varfile is available in Installing Server using varfile (on page 127) in the Appendix.

The Server Setup executable supports two types of installations: Complete and Custom.

Installing only an Agent is performed with a dedicated executable: nolio_agent_<OS>.

Complete Installation

Selecting the Complete Installation type installs all the components of the Serena Release Automation platform (Data Management Server, Execution Server, Agent, and Demo application) on a single server.

In the inputs for the installation process, specify **0** for the complete installation type.

Service Credentials on Windows

By default, the Server service is configured to run with the Local System account. For some advanced functions, such as remote installation of agents on Windows platforms, the service must log on as a user with administrative privileges.

If you selected MS SQL Server as the application database and Windows authentication, you will be required to change the service owner to the supplied login account.

To modify the service logon account:

1. Go to Windows Services.
2. Open the **Properties** dialog of the **Serena Server** service entry.
3. Click the **Log On** tab and set the appropriate account credentials.
4. Restart the service.

Server Service Management on Non-Windows Platforms

The following commands are available on non-Windows platforms for the Server Service startup and shutdown:

- Service startup: ./bin/startup.sh
- Service shutdown: ./bin/shutdown.sh

Note: All commands refer to paths relative to the <Serena Datamanagement HOME DIRECTORY>.

Installing Serena Release Automation Server

Installation may be performed with or without a varfile containing required values.

Installing with varfile

To perform an installation using varfile:

1. Copy the content of the varfile template from Installing Server using varfile (on page 127) in the Appendix and save it in a file called `varfile response.varfile`.
2. Update the required inputs in varfile response.varfile.
3. Transfer the installation file to the target machine.
4. Transfer the varfile response.varfile to the target machine.
5. For non-Windows machines, grant "a+x" permission to the installation file:

   ```
   chmod a+x nolio_server_<OS>_4_1_0_b<#>.sh
   ```

6. Execute the installation file:

   ```
   ./nolio_server_<OS>_4_1_0_b<#>.sh –q –varfile response.varfile
   ```

   Note: Additional flags can be added to the command line for logging more information.

   ```
   ./nolio_server_<OS>_4_1_0_b<#>.sh –q –varfile response.varfile
   –Dinstall4j.alternativeLogfile=[path] -Dinstall4j.keepLog=true
   ```

The server response.varfile template is available the Appendix as Server varfile Template (on page 128).

Installing without varfile

Running the installation without an input varfile requires that the Server executable will be invoked together with all inputs listed and filled with values in the same command line.

Before invoking the installation:

- Get a list of all required input parameters from the varfile template found in the Appendix as Installing Server using varfile (on page 127).
- Identify the parameter values needing update.
- When entering the parameters in the command line, prefix each parameter name with a hyphen and upper case letter 'V' (-V).

To perform an installation without varfile:

1. Transfer the installation file to the target machine.
2. For non-Windows machines, grant "a+x" permissions to the installation file:

   ```
   chmod a+x nolio_server_<OS>_4_1_0_b<#>.sh
   ```

3. Invoke the installation executable:

```
./nolio_server_<OS>_4_1_0_b<#>.sh –q –V<PARAMETERS>
```

The server response.varfile template, which contains all required parameters, is available in the Appendix as Server Response varfile Template (on page 134).

Notes:

- Default port for the Execution Server is 6600 and for the agent 6900.
- Execution Server Node name should be unique.
- Fill in the database variables according to the selected database type. Refer to database name definition procedure below.

To define the database name required for the installation input:

1 If MSSQL was selected for the database:

The value of the parameter **nolio.db.database.name** should be <SERVERNAME>\<INSTANCENAME> if this is a named instance, or just <INSTANCENAME> if the instance is a default.

If Windows Authentication was selected and the supplied login does not have Administrator privileges on the Data Management Server, the service will not start automatically and you will be prompted to start the service manually after the files were extracted and the schema populated.

2 If Oracle was selected for the database:

The value of the parameter **nolio.db.database.name** should be either the SID name or the Service Name. To determine the correct value, refer to Using Oracle as Database (on page 123).

The installer will check that the given schema does not exist. If it exists, the installer will check whether it is empty, and if empty, the Serena schema will be populated.

If the given tablespace and user are already assigned with non-Nolio objects, the installer will stop the installation process.

Note: To configure the Agent to startup automatically under Solaris or Linux systems, refer to Configuring Linux/Solaris for Automatic Startup of Agent (on page 38).

Chapter 3

Performing Custom Serena Server Installation

In This Chapter

The following topics describe how to perform a custom server installation.

Custom Installation

The Custom Installation installs selective components, generally either the Data Management Server or the Execution Server separately, thus supporting distribution of the Serena components across multiple servers.

In the inputs for the installation process:

1 Specify **1** for the custom installation type.

2 Select the components to install by marking the components as **true**.

Custom Installation supports the following component combinations:

- Data Management Server
- Data Management Server and Execution Server
- Data Management Server, Execution Server and Agent
- Data Management Server, Execution Server and Agent
- Data Management Server and Execution Server with Skip Database Configuration option checked.

  Note: Selecting Skip Database Configuration will configure Serena Release Automation to not use a database and will prevent much of the product functionality. This configuration should be performed in specific use cases. For further information, contact Serena support (on page 10).

- Execution Server
- Execution Server and Agent

For understanding the custom installation for any combination of Data Management and Execution Server, without the Database, refer to When to Perform Custom Installation (on page 26).

When to Perform Custom Installation

Custom installation is intended for users who want to:

- Add an additional Execution Server, or Servers, after the initial complete installation.

The following restrictions apply to custom component selection:

- If only **Agent** is selected, use the appropriate Agent Installation package.
- If **Skip Database Configuration** is selected, the Agent cannot be installed and product functionality will be limited.

When running a custom installation that includes Data Management and Execution Server, no manual configuration is required, unless additional Execution Servers are being added.

If additional Execution Servers are being added, the following steps are required:

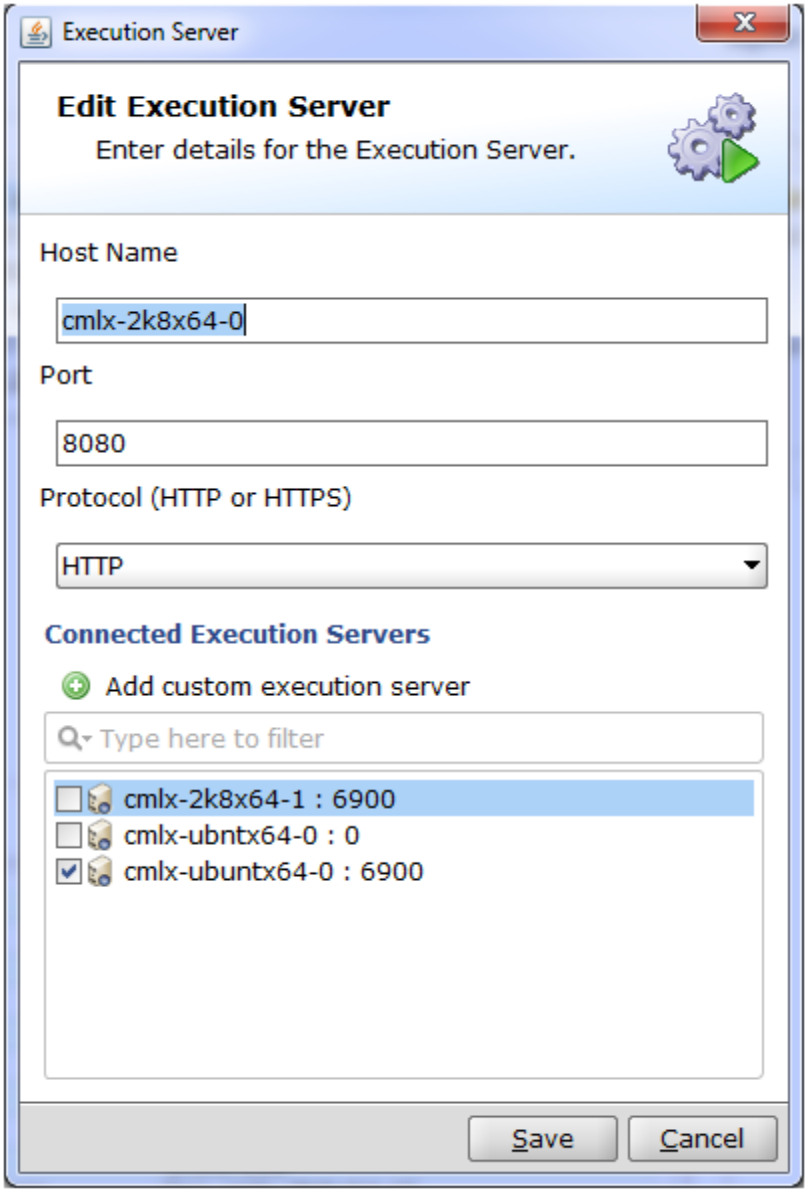
To connect an Execution Server to Serena Release Automation system:

1 In the **Administration** tab of the Navigation Panel, click **Agents Management**.

The **Agents Management** page opens.

2 Click  and select **Execution Server**.

The Execution Server details dialog will display.



3 In the **Host Name** box, enter the host name.

4 In the **Port** box, enter the port of the Web server used by the Execution Server. The default port is 8080.

5 In the **Connected Execution Servers** list, select the Execution Servers to connect to.

6 Click **Save**.

Once the Execution Server has been added, you may connect sibling Execution Servers, if required. Refer to Configuring an Execution Server (on page 83) in the Administration section.

Chapter 4

Installing Serena Agents

In This Chapter

As previously indicated, the Serena Release Automation Agent module may be installed on Windows, Linux, Solaris, AIX or zLinux s390x machines.

The installation of Serena Agents is performed by:

- **Local Installation** - Invoking the dedicated agent executable using a command line and passing the required inputs by one of the following methods:
  - In the command line
  - In an agent input varfile, a template of which is available in the Appendix (on page 136)
- **Remote Installation** - Using the Serena Release Automation UI

Note: The installation should be performed to a clean folder that does not hold any files or folders other than those of Serena Release Automation.

Note: For the installation package to be invoked on a Linux system, support for 32-bit applications must be enabled.

Important Note for Installation of Agents on AIX Platforms ONLY:

When installing Serena Agents on IBM AIX platforms, IBM JAVA JRE 6 32-bit must be installed on the target server prior to Agent setup.

Required User Credentials for Installation of Serena Agent

The following describes the user credentials required for installing the Agent:

- Windows
  - For the installation process, the logged in user must have administrative privileges in order to enable creation of the required service.
  - The owner of the Agent Service is configured by default to run with the Local System account.
  - For some advanced functions, such as access to resources on the network, the service must log on as a user with administrative privileges.
  - The owner of the Agent Service should have:
    - Write permissions to the file system
    - Permission to start a service
- Linux and Solaris
  - The Agent can be installed by any UNIX user that has permissions to create and update files under the installation directory.
  - The installation files should be extracted, using the installer, to a dedicated folder that will include only Serena Release Automation files.

    For example, if the target location for Serena will be under /opt, the Serena files will be extracted by the installer to /opt/SerenaAgent. The UNIX user assigned should have write permissions to the /opt folder.
  - To enable automatic startup in case of machine restart, the installation will try to add an entry to the runlevel scripts.

    If the installed user is not ROOT and does not have permissions to update the runlevel scripts, a message will be displayed during the installation.

    Refer to Configuring Linux/Solaris for Automatic Startup of Agent (on page 38) for a specific script that can be executed by the ROOT user after installation.

Local Agent Installation in Silent Mode

Installing Serena Agent is performed by running the Nolio Agent executable. Running the installation in silent mode requires the agent installation executable to be located on the target machine.

The installation can be performed by passing all required installation flags through the command line, or by using an agent input varfile.

To perform an agent installation using command line flags:

1 Copy the Serena Agent installation file to the target machine.

2 For non-Windows machines, grant “a+x” permission to the installation file:

chmod a+x nolio_agent_<OS>_4_1_0_b<#>.sh

3 Invoke the installation executable file with the following flags:

./nolio_agent_<OS>_4_1_0_b<#>.sh –q –V<PARAMETERS>

Following is an example of the command line syntax for installing an agent in silent mode:

```
./nolio_agent_<OS>_4_1_0_b<#>.sh –q
-Vsys.installationDir/opt/Serena/SerenaAgent
-Vnolio.nimi.node.id=myserver -Vnolio.nimi.port=6600
-Vinstall.service.lsa$Boolean=true -Vnolio.nimi.secured$Boolean=true
-Vnolio.execution.name=192.168.168.4 -Vnolio.execution.port=6900
-Vnolio.hiddenport$Boolean=false -Vsys.programGroupDisabled$Boolean=false
-Vsys.component.336$Boolean=true -Vsys.programGroupName=Nolio
-Vsys.programGroupAllUsers$Boolean=true -Vsys.languageId=en
```

To perform an agent installation using an agent input varfile:

1 Copy the Serena Agent installation file to the target machine.

2 Copy the content of the varfile template from Installing Agent using varfile in the Appendix and save it in a file named `deployer.silent.varfile`.

3 Update the deployer.silent.varfile to include appropriate values.

4 For non-Windows machines, grant "a+x" permission to the installation file:

```
chmod a+x nolio_agent_<OS>_4_1_0_b<#>.sh
```

5 Invoke the installation executable file with the following flags:

```
./nolio_agent_<OS>_4_1_0_b<#>.sh –q –varfile deployer.silent.varfile
```

Remote Agent Installation

To remotely install a new Agent, you may use the remote installation feature.

The remote installation enables the user to add agents on remote machines and connect them to existing Execution Servers. The ability to perform remote agent installation is dependent on the install base of the Execution Server.

Users can map a newly installed agent to the required Application, Environment, and Server Types, as defined in Serena, during remote agent installation, eliminating the need to map the agents to the appropriate configuration after all agents are installed from the Environments tab in the Serena client UI.

The following configurations are supported for Execution Server running on:

- Windows
  - Remote installation of agents on Windows platforms
  - Remote installation of agents on Linux, Solaris, AIX and/or zLinux platforms
- Linux or Solaris
  - Remote installation of agents on Linux, Solaris, AIX and/or zLinux platforms

Prerequisites

To enable remote agent installation, the Execution Server to which the new agents will be connected should have the target OS Agent's executable in its scripts folder.

For example, if the new agent to be installed should run on a Solaris x86 platform, the agent's installation file (nolio_agent_solaris-x86_4_1_0_b<#>.sh) should be copied to the Execution Server' script folder located at:

```
<Execution Server Home Directory>/scripts
```

The target machine must be configured properly to allow remote installation. In the case of Windows, the Server Service owner must have administrative privileges and 'Log on as a service' rights on the machine.

Windows

- **Workstation** and **Server** services are running
- **Admin$** share is available
- **Windows Network** is running
- **Printer and File Sharing** is activated
- Incoming network Users authenticate as themselves
  - Simple File Sharing is turned off
  - Network Users Identify as Guests is turned off
- Firewall allows incoming traffic through ports 135 and 445

Linux / Solaris

- **SSH** must be enabled

Dynamic Agent Mapping

By default, the ability to map agents to Application, Environment and Server Type when performing a remote installation is not enabled, and the input boxes for Agent Mapping Details will not be displayed in the Agent Installation window.

To enable dynamic agent mapping during remote agent installation:

1. In the Serena Release Automation Client UI, select the **Administration** tab.
2. Select **System Settings**.
3. Click the  icon.

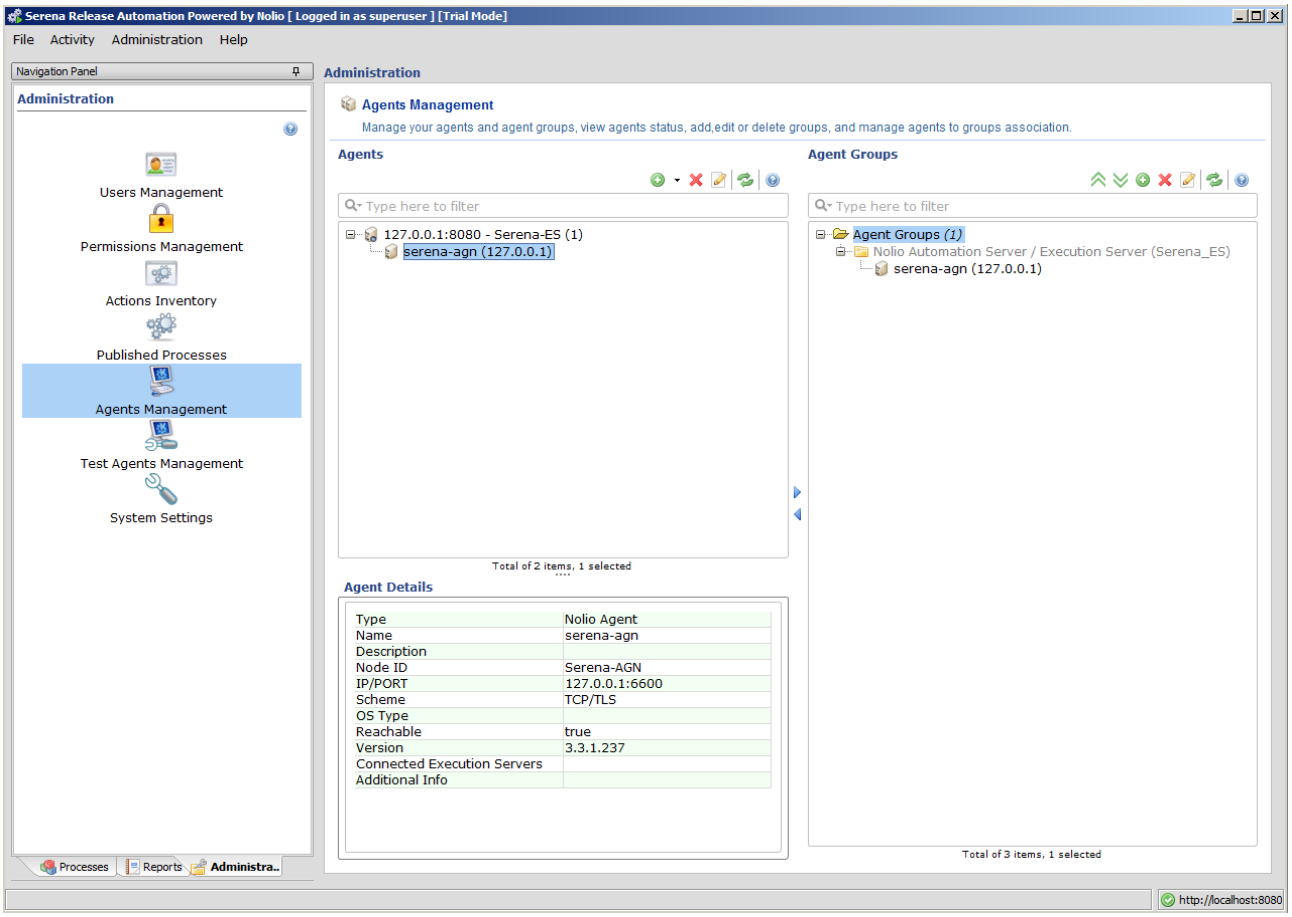
   The Add System Settings dialog displays.
4. In the **New key** box, enter **DYNAMIC_AGENT_MAPPING_ENABLED**.
5. In the **New value** box, enter **true**.
6. Click **Save**.

It is not necessary to restart the Serena Server service or the client UI. The next time you open the remote agent installation wizard, the agent mapping detail fields will display.

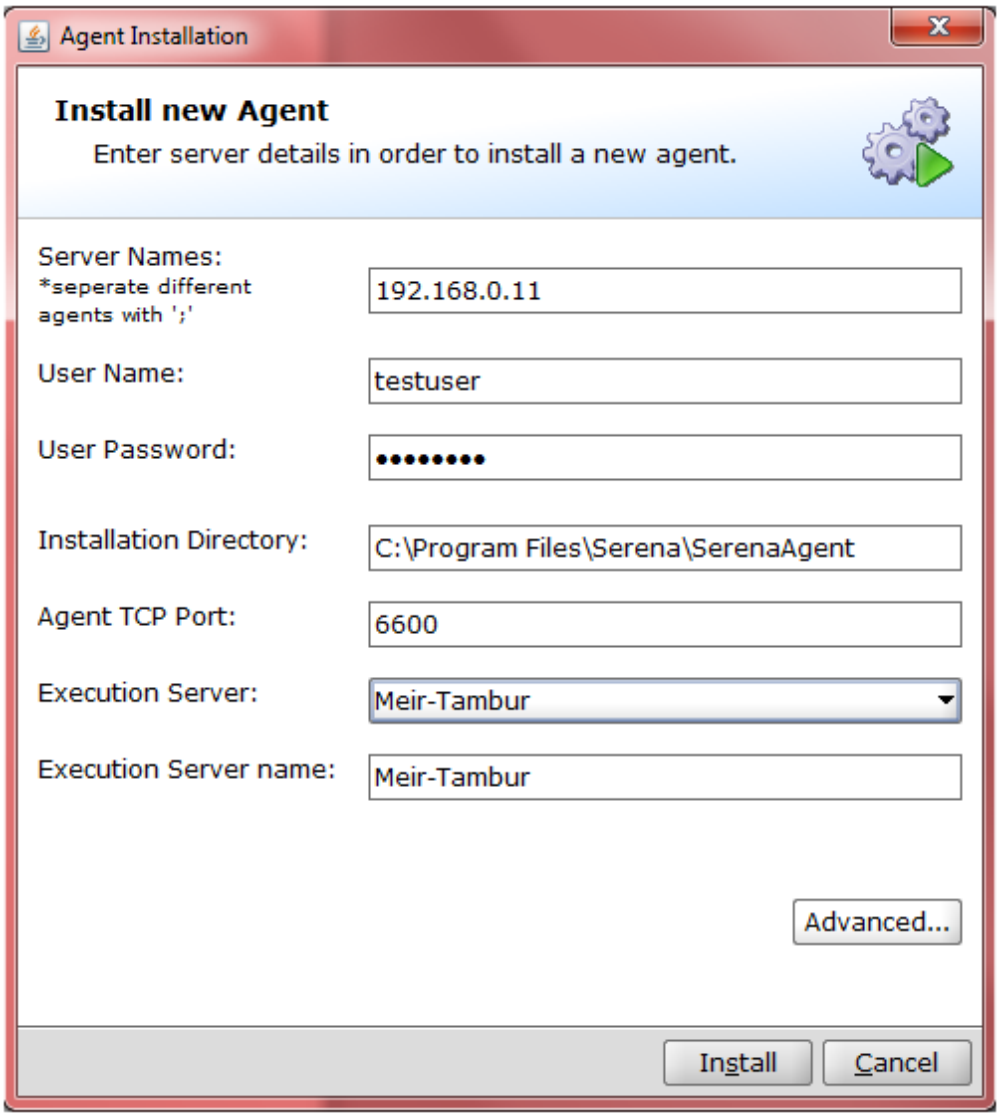
Running Remote Installation

To run a remote installation:

1 In the Serena Release Automation **Client UI**, select **Administration > Agents Managements**. The **Administration Agents Management** window opens.



2 Click , and on the dropdown menu select **Install Serena Agent** to install the agent on the relevant platform. The **Install New Agent** window opens.



3 In the **Server Names** box, enter the DNS names or IP addresses of the servers on which the new agent.

For multiple servers, separate agents with a semicolon (;).

4 In the **User Name** box, enter the user name of an account that has administrative privileges on the remote machines.

For Linux/Solaris platforms, the **User Name** supplied should have the same user credentials as described in Linux and Solaris section of Required User Credentials for Installation of Serena Agent (on page 30).

5 In the **User Password** box, enter the password for the user account.

6 In the **Installation Directory**, enter the folder in which to place the installation.

In the **Agent TCP Port** box, enter the number of the TCP port on which the agent will listen.

Note: If the port is not available, a new window will open with a prompt to provide new ports.

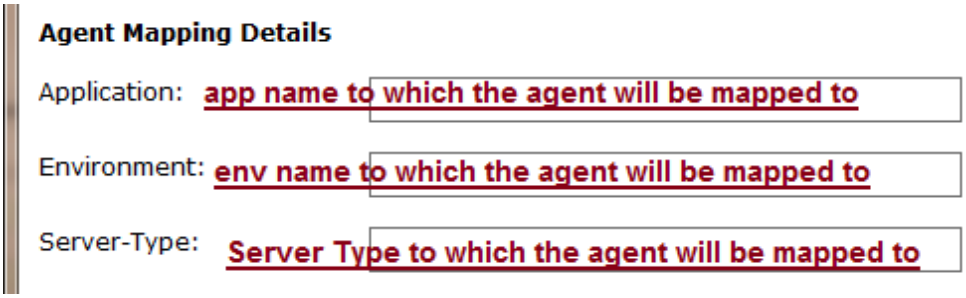
7 In the **Execution Server** list, select IP address or DNS name of the Execution Server with which these agents will be associated.

8 In the **Execution Server name** box, enter the Execution Server name as known to the agents, which is the same as in the Execution Server drop-down list, or a different hostname or IP address for the machine with which agents will be able to communicate.

9 For Windows platforms, you can click the **Advanced** button to define different credentials for remote servers:

a. Enter the credentials of a user account that will be used to run the Agent service. The user must have administrative privileges and 'Log On As A Service' rights.

10 If the System Setting parameter **DYNAMIC_AGENT_MAPPING_ENABLED** is set to **TRUE**, the Install new Agent window will also display boxes for entering the Application, Environment, and Server Type to associate with the server agents.

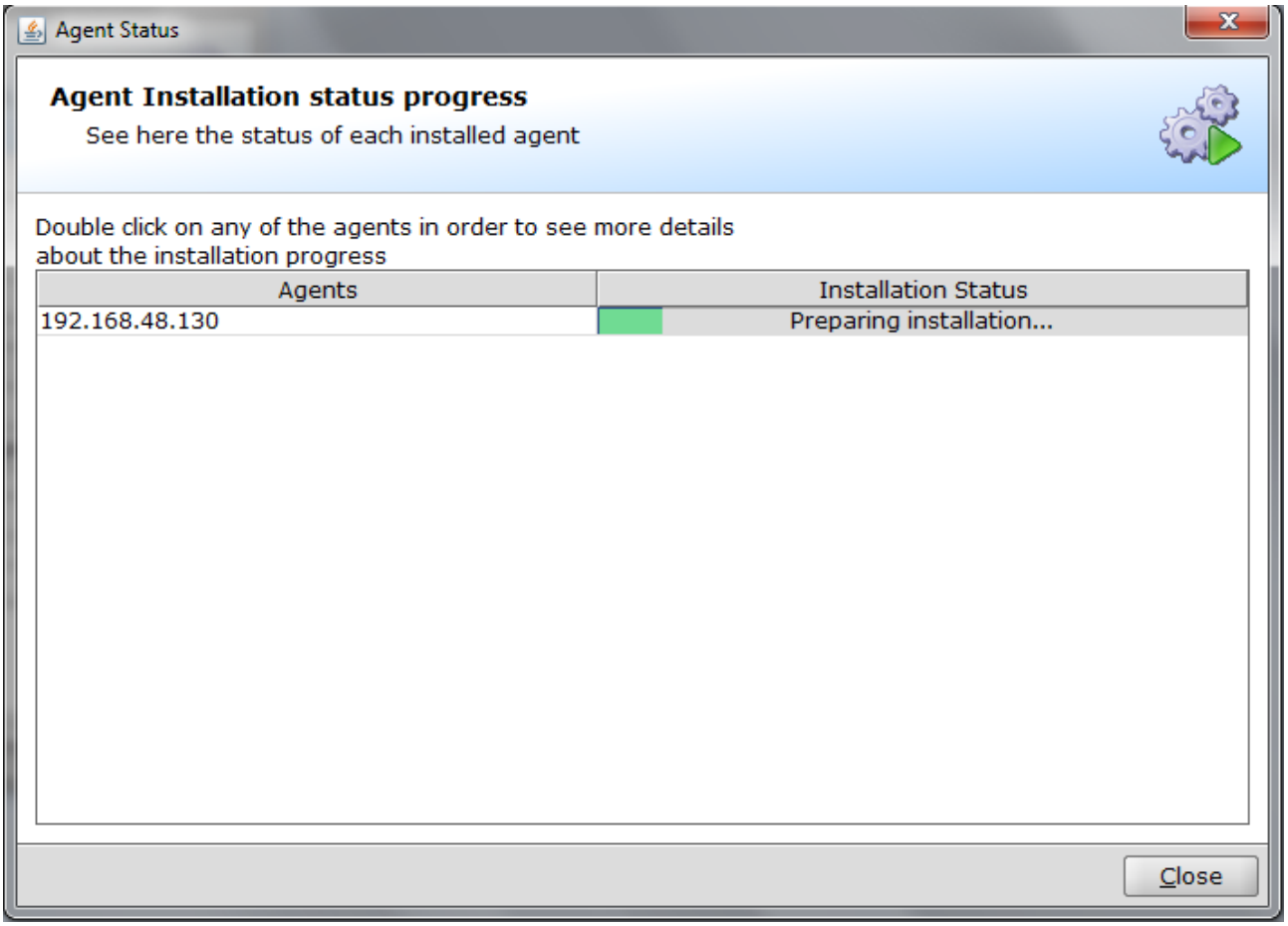


a. In the **Application** box, enter the name of the application to which the agent will be mapped.

b. In the **Environment** box, enter the name of the application environment to which the agent will be mapped.

c. In the **Server Type** box, enter the server type to which the agent will be mapped.

11 Click **Install**. The **Agent Installation Status Progress** window opens.



12 When the installation completes, click **Close** to close this window.

As part of the installation process, the Agent services will be set to start automatically after the server's boot. If the installer will not be able to do so, you will be notified. Failure might occur due to permissions issues that will prevent the installation process from copying the required files to the needed location. For instructions on how to set automatic startup, see Configuring Linux/Solaris for Automatic Startup of Agent (on page 38).

Note: On Windows platforms, in the case where the installer fails to install or start the Serena Agent service, the installation will be rolled back and the installation files will be removed.

Advanced Agent Configuration

This section describes advanced configuration options for the Agent.

Agent Service Management on non-Windows Platforms

Note: All commands refer to paths that are relative to the <SERENA AGENT HOME DIRECTORY>.

- Service startup: ./deployer_deamon.sh start
- Service shutdown: ./deployer_deamon.sh stop
- Service restart: ./deployer_deamon.sh restart
- Service status: ./deployer_deamon.sh status

Configuring Linux / Solaris for Automatic Startup of Agent

Note: The following configuration procedure should only be performed if you installed the Server or the Agent with a non-root user.

To configure automatic Agent startup on Linux/Solaris:

1. Open conf/S70nolioagent for editing.
2. Find the RUN_AS_USER=root entry.
3. Uncomment the line and replace 'root' with the user that owns the Agent installation.
4. For Solaris machines only:
   a. Comment out the following line:

      ```
      su -m $RUN_AS_USER -c "\"$REALPATH\" $2"
      ```

   b. Uncomment the following line:

      ```
      su  $RUN_AS_USER -c "$REALPATH $2"
      ```

5. Save the file.
6. Connect as ROOT user and run the following script from Agent root folder:

   ```
   install_service_to_runlevels.sh ./conf S70nolioagent
   ```

   After server reboot, the Agent Service will start with the specified user.

Automatically Assigning Serena Agents to Application Components for Cloud Support

It is possible to automatically assign Serena Agents to predefined applications, environments, and server types when replicating images as part of cloud support.

Add an agent mapping file, according to the Agent Mapping File Template (on page 139), to the host image. The mapping file will be used to automatically assign Serena Agents to the predefined application, environment, and server type components in Serena when the agent connects to its Execution Server.

Chapter 5

Post-Installation Activities

In This Chapter

In the event it is necessary to uninstall Serena Release Automation, use the following procedures.

Controlling Serena Server Service on Non-Windows Platforms

Serena Server service can be manipulated by running a single script with different attributes.

Execute the script from the Serena Release Automation Home Directory and by the user that owns Serena system.

To add Serena Server service to the services list for automatic start after reboot:

At the Command prompt, run:

```
./nolio_server.sh install
```

If executed, the **install** command will enable the Serena Server service to start automatically in case the host was rebooted similar to automatic start for the Agent.

If Serena Server service needs to be started with a user other than root, edit the `nolio_server.sh` script as follows:

1. Open shell file
2. Search for RUN_AS_USER
3. Type the username

Note: User should have write, execute and read permissions on the Serena Release Automation root.

To start the Serena Server service:

At the Command prompt, run:

```
./nolio_server.sh start
```

To stop the Serena Server service:

At the Command prompt, run:

```
./nolio_server.sh stop
```

To restart the Serena Server service:

At the Command prompt, run:

```
./nolio_server.sh restart
```

To query the status of the Serena Server service:

At the Command prompt, run:

```
./nolio_server.sh status
```

To remove Serena Server service from the services list:

At the Command prompt, run:

```
./nolio_server.sh remove
```

Uninstalling Serena Release Automation

To uninstall Serena Release Automation from a machine running under Windows:

Invoke the specific uninstall executable supplied:

1. Choose **Start > All Programs > Serena** .
2. In the menu that opens to the right, choose **Serena Uninstaller**.
3. When queried whether to remove the installation, click **Yes**.
4. Follow the uninstall wizard instructions.

To uninstall Serena Release Automation from non-Windows machine:

1. Go to the <Serena Server Home Directory>.
2. Run the uninstall task (./uninstall) and follow the instructions.

Note: The schema in the database will not be removed.

Uninstalling an Agent

This section provides instructions for uninstalling an Agent from Windows and Linux systems.

Windows

Uninstalling a Serena Agent from a machine running under Windows is performed by invoking its uninstall executable.

To uninstall SerenaAgent:

1. Choose **Start > All Programs > Serena**.
2. In the menu that opens to the right, choose the **Serena Agent Uninstaller**.
3. When queried whether to remove the installation, click **Yes**.
4. Follow the uninstall wizard instructions.

Linux / Solaris

Uninstalling an Agent on a machine running on non-Windows platform is performed by running the Serena Agent uninstall shell script.

- If the target machine has a GUI, uninstall is performed in the same manner as for Windows (on page 41).
- If the target machine does not have a GUI, the uninstall process is run in text mode:
  a. Go to <Serena Agent Home Directory>
  b. Run the uninstall task (./uninstall) and follow the instructions.

Chapter 6

Running Serena Release Automation

In This Chapter

This chapter explains how to invoke the Serena Release Automation **Client UI**.

Note: Sun Java version 6 update 16 or higher must be installed on your computer.

Launching the Client UI

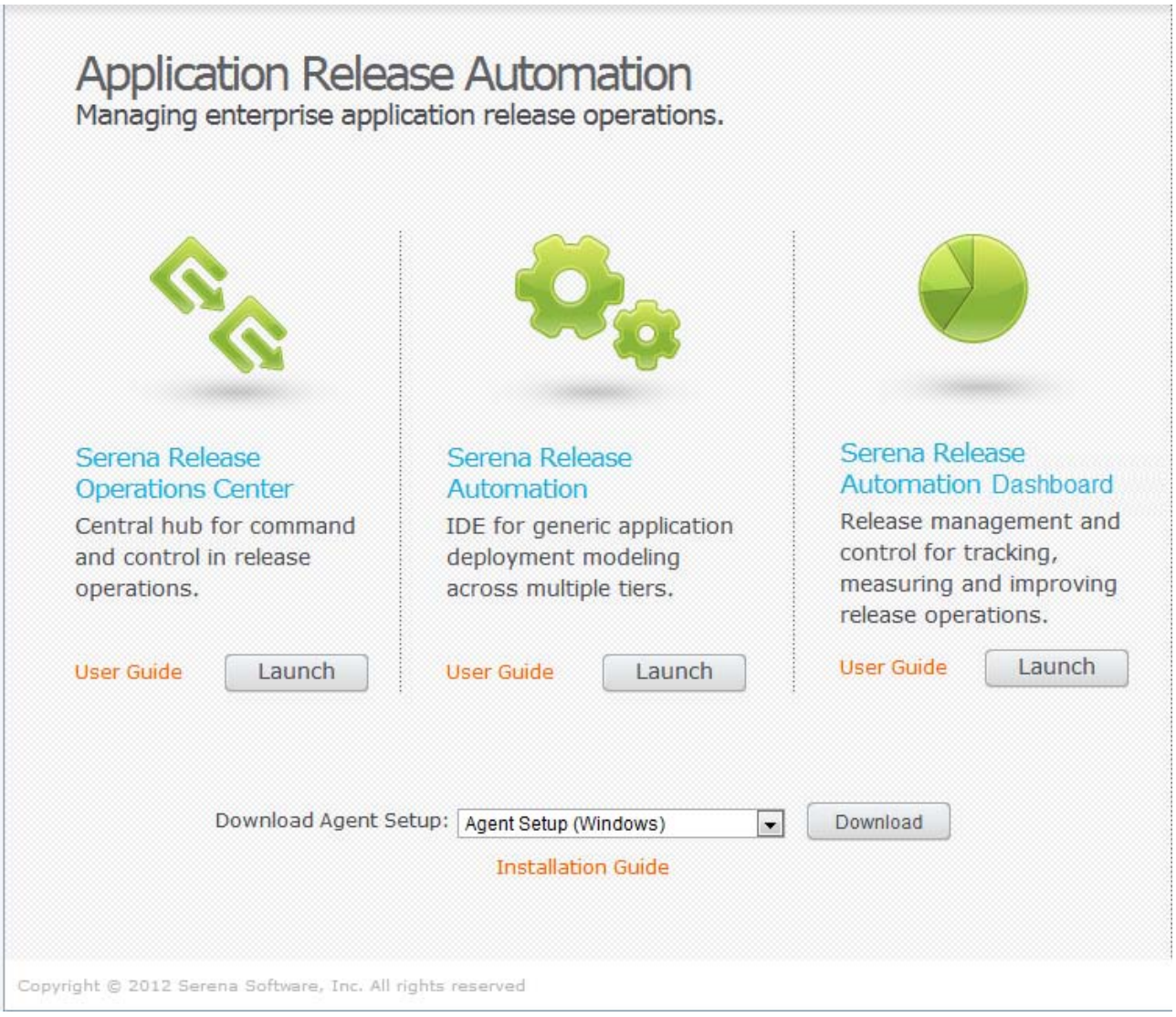
The Serena Release Automation UI uses Web Start technology.

To download and install the Client UI:

1 In your browser, enter:

http://<hostname or IP address of center>:8080/serena-app

If Sun Java 6 update 16 or higher is not installed, you will be provided with a link to install Java. You will need to update your client machine with this JRE version.



2 Click the Launch button under **Serena Release Automation**. The **Download Application** window opens. If this is the first time you are running the application, the **License Message** window opens.

3 Click the **Log to Serena Management Console** button to log into the Release Deployment Dashboard.

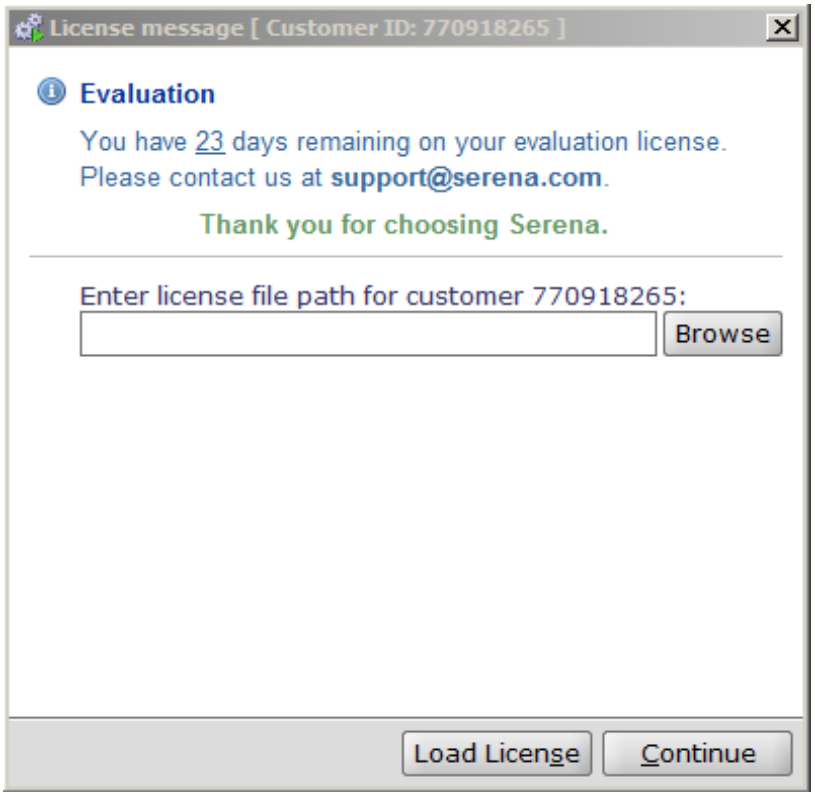
Note: The credentials mechanism for the Release Deployment Dashboard is based on the mechanism defined within the Release Automation User Management component.

4 If you have obtained a L**icense** file, you may now apply it. Enter the path to the L**icense** file or click **Browse** and click **Load License**.

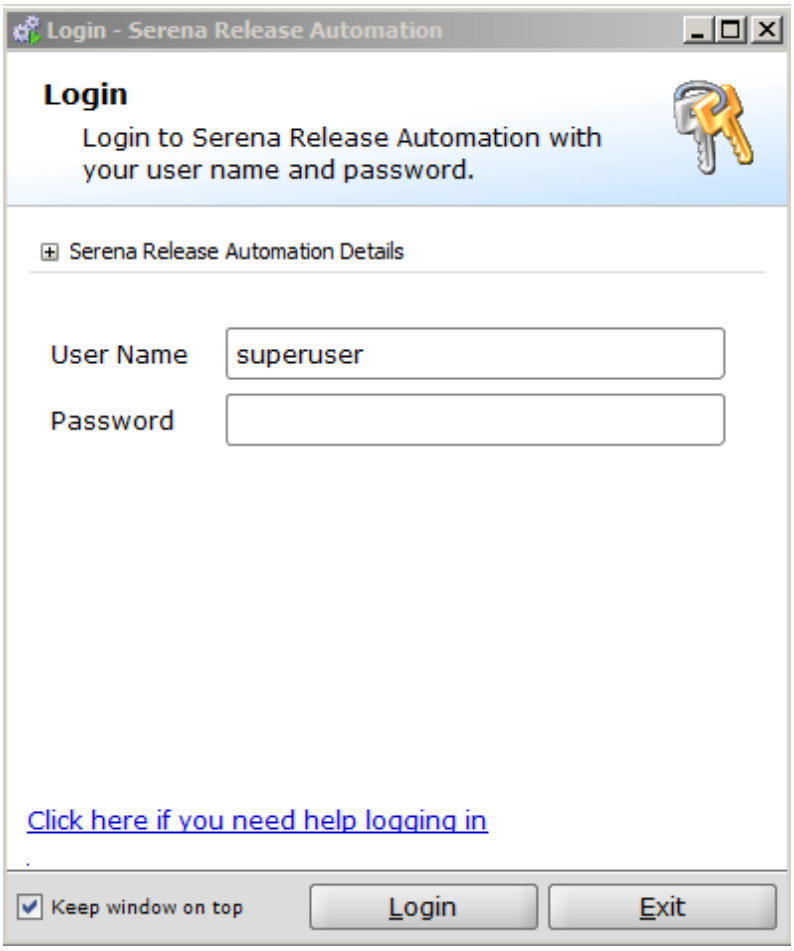
5 If you do not have a License file at this time, you may continue your work and apply the License at a later time by clicking **Continue.**

For instructions on how to update the **License** file, see Updating Serena Release Automation License (on page 112).

Note: If you do not supply a **License** file, the installed product will be enabled as an evaluation kit.



The application will now load.



Setting Up Serena Release Automation

This section describes the steps required for setting up Serena Release Automation for use.

Overview

During the Serena Release Automation installation, a default administrative user called **superuser** is created. You must use this user on the first login to the Serena system in order to set up Serena Release Automation for use.

Note: For information on changing the default administrative user's details, refer to Modifying the Default Administrative User's Details (on page 71).

Workflow

The workflow for Serena Release Automation setup is as follows:

1. Run Serena Release Automation Client UI (on page 43).
2. Add Serena Execution Servers (on page 81).
3. Create Agents Groups (on page 87).
4. Optional: Add agents to the Test Agents Group (on page 93).
5. Add Agents to Agents Groups (on page 88).
6. Add **Users** and give them Permissions (on page 55).

Chapter 7

Performing Serena Release Automation Administration

In This Chapter

This section provides an overview of the tasks that the Serena Release Automation Administrator **must** perform for setup and regular use of Serena Release Automation.

This section also introduces the Serena Release Automation **Administration** tab, where administrative tasks are performed.

These tasks require you to have **superuser** role.

For information on roles, see Managing Serena Release Automation Users and Permissions (on page 55).

Overview

The following administrative tasks are necessary for setting up Serena Release Automation and performing the various tasks necessary for the ongoing and continuous use of Serena Release Automation:

- Managing Serena Release Automation users and user groups
- Granting application, environment, and server group permissions to Serena Release Automation users and user groups
- Managing the actions inventory
- Managing the Published Processes
- Managing the agents installed on the organization's servers
- Managing test agents on which flows and processes can be tested
- Managing the System Settings

All administrative tasks are performed in Serena Release Automation's **Administration** tab.

Note: This section focuses on the **Administration** tab only. For an explanation of the Serena Release Automation Client UI, its elements, and their use, refer to *Introduction to Serena*.

For information on performing non-administrative tasks in Serena Release Automation, refer to *Serena Release Automation User Guide*.

Serena Release Operations Center users should also consult *Serena Release Operations Center User Guide*.

Logging in to Serena Release Automation

To log in to Serena Release Automation Client UI:

1 If you have performed a "Complete Installation" from the client machine, double-click  on your desktop.

OR

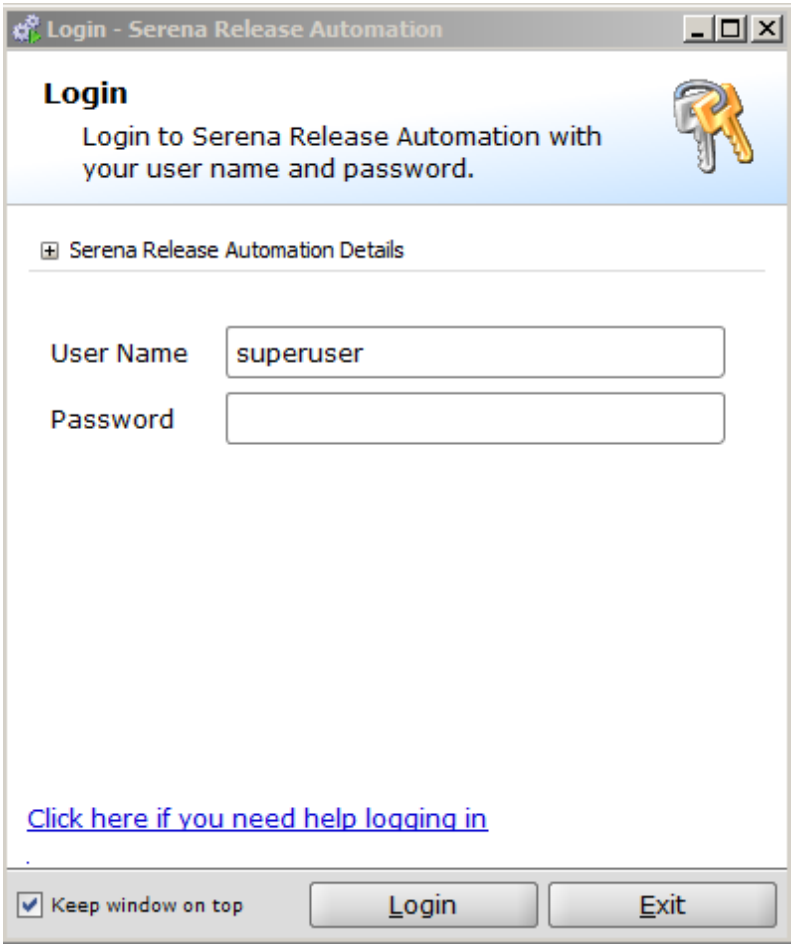
In your Web browser, type the Serena Release Automation URL:

http://<IP>:<port>

where **<IP>** is the IP address of the Data Management Server and **<port>** is 8080 (unless you have changed the port).

The Data Management Server Web page opens. Click the **Serena Release Automation** button.

The Login window opens.



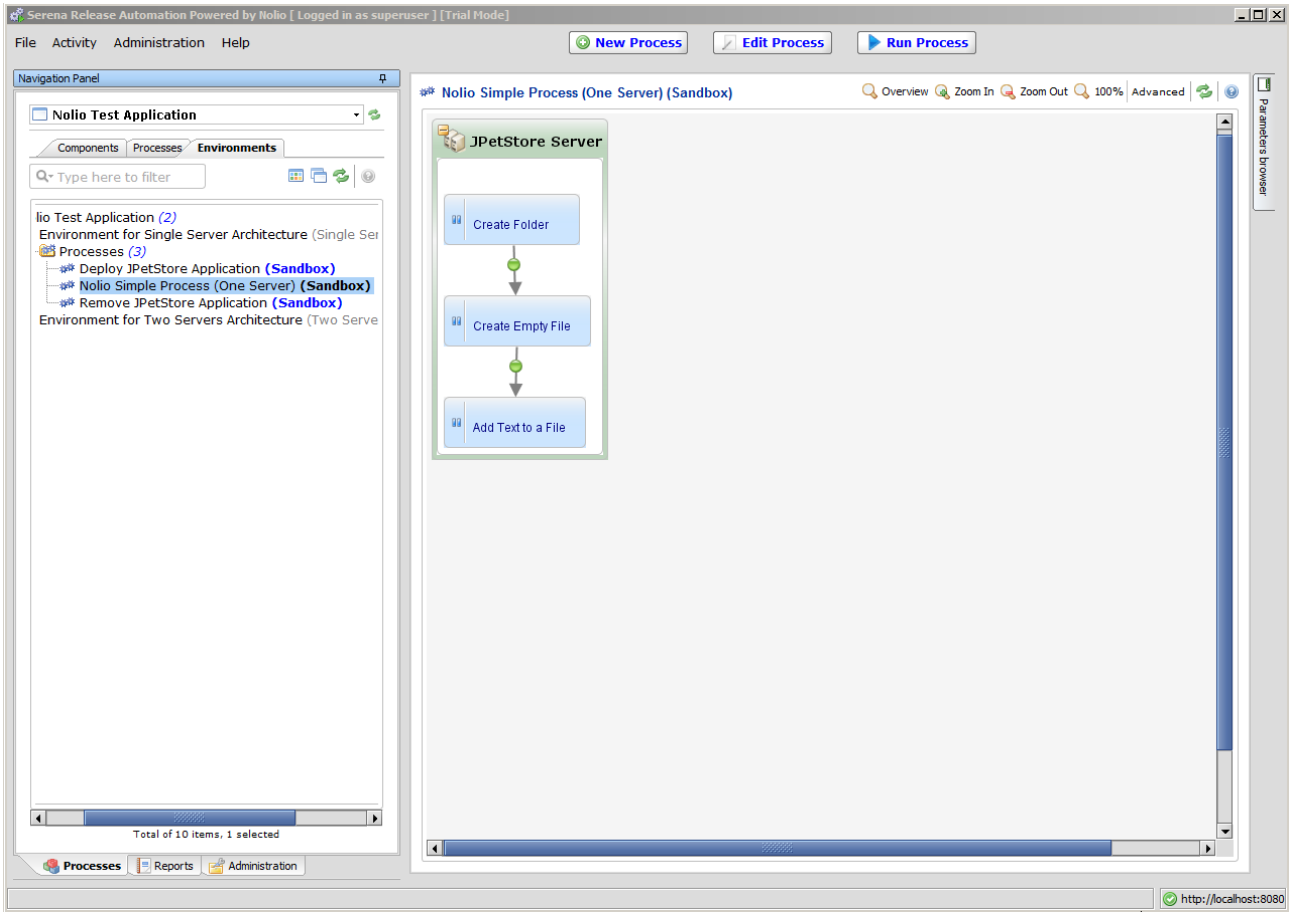
2 In the **Serena Release Automation Address** field, type the URL of the Serena Release Automation server, or select the relevant URL from the dropdown list.

3 In the **User Name** field, type your user name, or leave the default 'superuser'.

4 In the **Password** field, type your password.

If using the default user name, enter the default password 'suser'. To change the default password, refer to Adding and Editing Users (on page 59).

5 Click **Login**. Serena Release Automation opens displaying the **Environments** tab of the **Processes** window.



Navigating to the Administration Window

To navigate to the Administration window:

- At the bottom of the **Navigation Panel**, click the **Administration** tab.

  OR

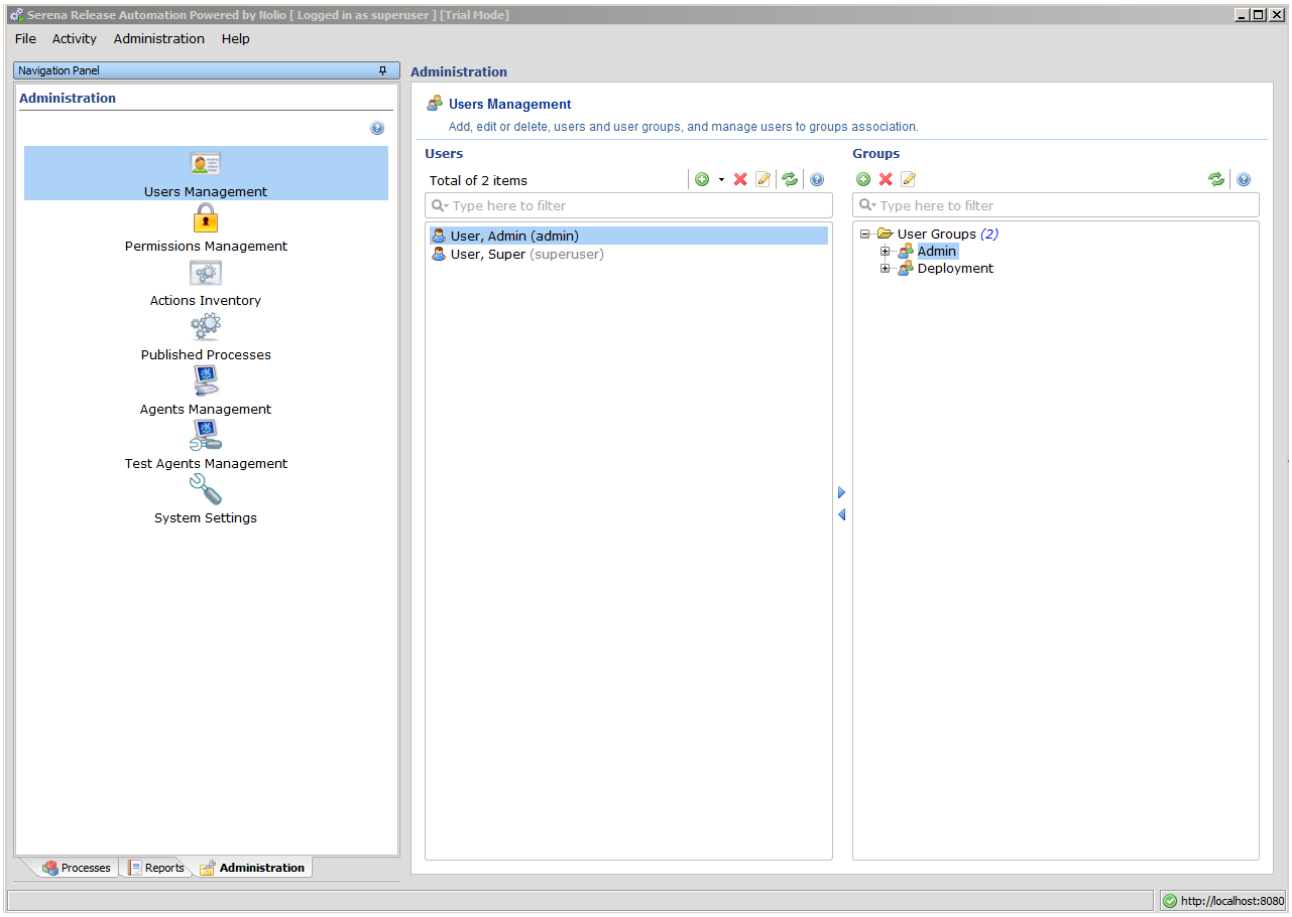
- In the toolbar, click **Administration**, then click the relevant function.

Using the Administration Window

The **Administration** window includes the following components:

- **Administration Panel**: Appears on the left and includes labels for different types of administrative tasks to be performed.

- **Central pane**: Displays content that is determined by the selected Administration task.



Chapter 8

Managing Release Automation Users and Permissions

In This Chapter

This section explains how to add, edit, and delete Serena Release Automation users and user groups, and how to assign them permissions for specific server groups and environments.

Note: These tasks require you to have **superuser** role.

Overview

Serena Release Automation offers three methods for adding users:

- Adding a single user through the Add User dialog. See Adding and Editing Users (on page 59).
- Adding multiple users using a batch file. See Adding Users Using a Batch File (on page 62).
- Importing users from an LDAP directory. See Importing Users from LDAP (on page 66).

In addition, Serena Release Automation offers a high level of granularity in defining users. There are assignable attributes by which you can control a user's access and operational capability:

- **Roles** - Understanding User Roles (on page 56)

  Roles are set when a user is added.

- **Permissions** - Understanding Permissions

  Permissions are set after a user is added and are based on control at the application, environment and process levels.

Understanding User Roles

When adding a new user, assignment of a user role, which determines administrative level, is required.

User accounts are assigned one of the user level roles:

- **User:** A non-administrative user who:
  - Can access the **Reports** panel
  - May be granted permissions on server groups
  - Cannot view applications other than the applications to which they have **Can View Application** permission
  - Additional permissions are granted at the environment and process levels for an application
  - Cannot access the **Administration** panel
- **Superuser**: A user who can perform administrative tasks, such as:
  - Managing Serena Release Automation users, permissions, available actions, published processes, agents, test agents, and system settings
  - Accessing all panels and tabs.
  - Can view all applications and perform all activities on any application. Therefore, there is no need to assign application level permissions
  - Operating on all server groups, applications and environments

Understanding Permissions and Roles

When a regular user is added to Release Automation, there are no permissions attached to the user role for access to screens and functions.

The first level of permission to grant a user is to view an application (**Can View Application**). This is an initial permission that if not selected, for any user other superuser, no data will be shown in Release Automation or Serena Release Operations Center. Once a user has **Can View Application** permission, additional permissions can be layered according to the three Release Automation levels: application, environment, and process.

For application level permissions, there are **Application Owner**, **Execute in All Environments**, **Application Publisher,** which also includes **Application Designer**, and finally, the permission of **Release Template Designer** for use in Release Operations Center.

- **Application Owner**

  Release Automation administrators can now grant users or user groups with **Application Owner** permissions. Defined in the application level, **Application Owner** grants full permissions (superuser) on a specific application and all its environments.

- **Execute in All Environments**

  Defined in the application level. This new permission enables a user to execute all processes and releases in all environments under the current application.

For environment level permissions, there are **Environment Admin**, **Can Execute All Processes** under this environment, and finally, the permission of **Release Designe**r for use in Release Operations Center.

For processes level permissions, individual processes may be selected for users who are not granted **Can Execute All Processes** for the environment.

Each user is tested for specific permission before access is given to him. Users with role "Superuser" have all rights.

Combinations of user permissions form functional roles.

- **Design Template** is granted to a user who has **Release Template Designer** or **Application Owner** permission on application level.
- **Design Release** is granted to a user who has **Release Designer** permission on environment level, **Environment Admin** on environment level or **Application Owner** on application level.
- **Release Executor** is granted to a user who has:
  - **Can Execute All Processes** permission on environment level
  - Or **Execute In All Environment**s on application level
  - Or **Environment Admin** on environment level
  - Or **Application Owner** on application level

Understanding User Authentication

Release Automation provides a users management system that supports login and authentication for its users. When a user is added to Release Automation, their details are added to the users table. These details are used later on to authenticate users upon login. The 'username' attribute is a unique identifier of the user that is used to lookup that user in the database.

Release Automation supports two authentication methods:

- Basic

  The Basic authentication method is a proprietary user management mechanism that stores the user's password in the database and authenticates the user against it upon login.

- LDAP

  Serena supports authentication of Release Automation users against any LDAP server.

  When using LDAP authentication, the user password is not stored in Release Automation. Whenever an LDAP user logs in, the password that they provided during login is authenticated against an LDAP server. The required LDAP attributes are stored in the user's record in the database. See Understanding LDAP Authentication Attributes (on page 58).

  Release Automation supports authentication against any LDAP server, including Active Directory.

Understanding LDAP Authentication Attributes

The following values are used to authenticate the LDAP user:

LDAP Authentication Attributes

Name	Required	Description
Username	Required	Unique identifier of the user in the Release Automation system. If Security Context is not provided, the username is used as both the identifier in the Release Automation system and the Security Context for the purposes of LDAP authentication. In this case the username format should adhere to the specifications of the Security Context as defined below.
LDAP Host	Required	Hostname or IP address of the LDAP Server
LDAP Port	Optional	Ports that the server uses for the LDAP/LDAPS protocols. If left blank then defaults will be used. This setting will also be used when authenticating a user that was created through this import operation. If not provided, the port usually resolves to: • 389 for LDAP • 636 for LDAPS

Name	Required	Description
Search Context	Required for LDAP	Context in which to search for the user, for example, the user's domain. The Search Context should be provided in a Qualified Name format: dc=mycompany,dc=com.
Security Context	Optional	User's security context used to authenticate against the LDAP server. If blank, the username is used as the LDAP security context, in which case it must follow the same guidelines as described here. The Security Context should be an identifier that is recognized and supported by the LDAP server. Common options for Active Directory Security Contexts: • UserPrincipalName: Usually formatted as user@domain, for example, john.doe@mycompany.com • SamAccountName: Usually formatted as <domain short name>\samaccountname" mycompany\jdoe • Distinguished Name: The FQDN of the user, for example, CN=John Doe,CN=Users,DC=mycompany,DC=com
SSL	Optional	Boolean indicating whether LDAP or LDAPS should be used. Default is LDAP.

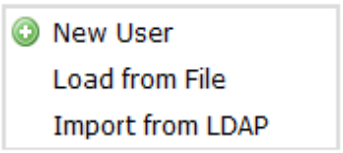
Adding and Editing Users

To add a user:

1 In the **Administration** tab of the **Navigation Panel**, click the **Users Management** tab.

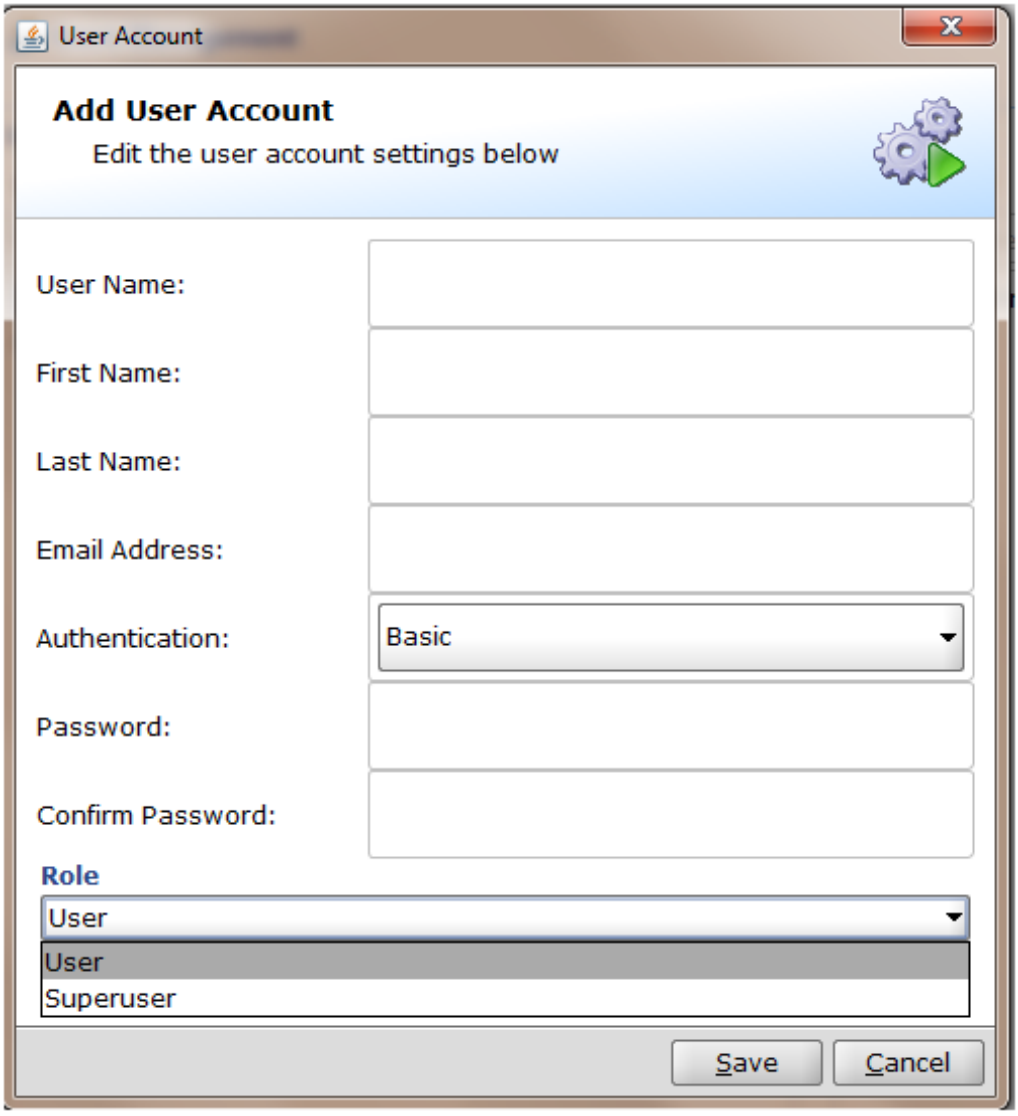
2 To add a new user, click  in the **Users** list.

A selection window opens.



3 Select **New User**. You may also click  .

4 The **Add User Account Settings** window opens.



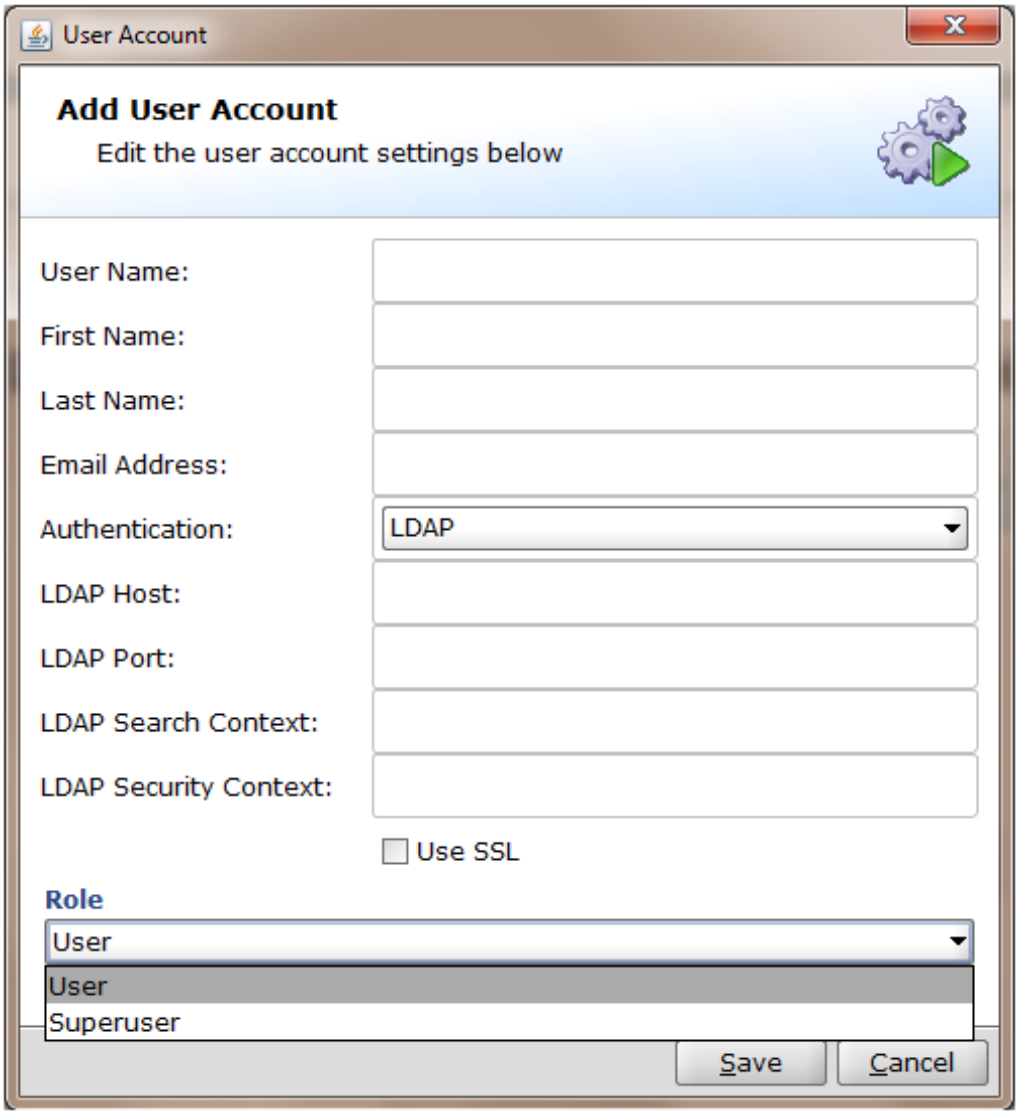
5 Complete the fields according to the User Account Settings table.

6 Click **Save**.

To edit an existing user:

1 Select the relevant user in the **Users** list, and click , or double-click the **User Name**.

The edit **User Account** window opens.



2 Edit according to the information in the User Account Settings table.

3 Click **Save**.

User Account Settings

Field	Description	Example
Username	Type a user name for the user. In the **Edit User details** window, this field is read-only.	JohnSmith
First Name	Type the user's first name.	John
Last Name	Type the user's last name.	Smith
Email Address	Type the user's email address.	johnsmith@mycompany.com

Field	Description	Example
Authentication	Select the authentication method for the user: • BASIC (User name and password) • LDAP (includes Active Directory) See Understanding Release Automation User Authentication.	LDAP
Role	Select the relevant role for the user. For details, see Understanding User Role (on page 56)s.	USER
If Basic **authentication method is selected, the following fields display:**		
Password	Type a password.	mypassword
Confirm Password	Retype the password.	mypassword
If LDAP **authentication method is selected, the following fields display:**		
LDAP Host	Resolvable name or address of the LDAP server. Must contain a valid value if Active Directory or LDAP authentication methods are used.	
LDAP Port	Port number through which the LDAP server serves the LDAP/LDAPS protocol. Leave blank to use LDAP defaults.	
LDAP Search Context	Path to an LDAP entry that is an ancestor of the LDAP user.	
LDAP Security Context	Security context that will be used to authenticate the user. The security context is the value of the **userprincipalname** LDAP attribute. If blank, the username attribute will be used as the LDAP security context.	
Use SSL	Select to use SSL in connection to LDAP.	

Adding Users Using a Batch File

Multiple user accounts can be created in Release Automation in a single operation. Individual user attributes are provided through a tab-delimited file.

To Create a Batch File for Adding Users

Create a text file with the following characteristics:

- Each line in the tab-delimited file corresponds to a user account.
- Each line *must* contain *all* of the user account attributes listed in the Add Users Batch File Layout (on page 63), even if the values are empty strings.

Add Users Batch File Layout

Add Users Batch File Layout

	Entry	Description	
1	User Name	Identifier of the user in Release Automation. Must be unique across all users. Allowed characters: alphanumeric, backslash, period, underscore, hyphen and '@'.	
2	Last Name	Last name of the user.	
3	First Name	First name of the user.	
4	Email	Email address used by Release Automation server to send notifications.	
5	Password	User's password when using Basic authentication. Leave blank when using non-Basic authentication.	
6	Authentication Method	Method used to authenticate user. • BASIC: Release Automations proprietary authentication method which requires storing a password in Release Automation. • LDAP: Authentication is performed again an LDAP server. LDAP authentication required configuration of a valid LDAP server and a valid LDAP Search Context attribute for looking up users.	
7	LDAP Host	Resolvable name or address of the LDAP server. Must contain a valid value if Active Directory or LDAP authentication methods are used.	
8	LDAP Port	Port number through which the LDAP server serves the LDAP/LDAPS protocol. Leave blank to use LDAP defaults.	
9	LDAP SSL	LDAP: Use the LDAP protocol (no encryption). LDAPS: Use the LDAPS protocol (SSL encryption).	
10	Role	SUPERUSER:	Release Automation administrator can perform all applicative and administrative operations. Only a SUPERUSER can perform administrator operations.
			USER role requires that either the OPERATOR (Run Processes) or DESIGNER (Design Processes) role be assigned, at a minimum. User is **not** a valid batch file value.
		OPERATOR:	Can execute processes.
		DESIGNER:	Can design processes.
		PUBLISHER:	Can perform design operations and publish processes.
		OPERATOR-DESIGNER:	Can perform design operations and publish processes.
		OPERATOR-PUBLISHER:	Can execute processes, perform design operations, and publish processes.
11	LDAP Search Context:	Path to an LDAP entry that is an ancestor of the LDAP user.	

	Entry	Description
	LDAP Security Context:	Security context that will be used to authenticate the user. The security context is the value of the **userprincipalname** LDAP attribute. If blank, the username attribute will be used as the LDAP security context.
12	Row termination character	Constant: '#'

Example

Following is an example of the layout for a batch file to add users:

```
guy.basic<tab>ginzburg<tab>guy<tab>guy@abc.com<tab>guypw<tab>BASIC<tab><tab><tab><tab>SUPERUSER<tab><tab><tab>#<cr><lf>
guy.basic2<tab>ginzburg<tab>guy<tab>guy@abc.com<tab><tab>LDAP<tab>monster<tab><tab>LDAP<tab>SUPERUSER<tab>CN=Users,DC=serena<tab><tab>#<cr><lf>
```

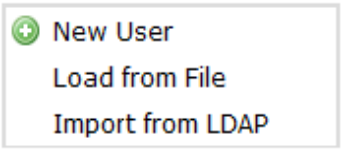
Loading Users Using Batch File

To Load Users Using a Batch File

1 In the **Administration** tab of the **Navigation Panel**, click **Users Management**.

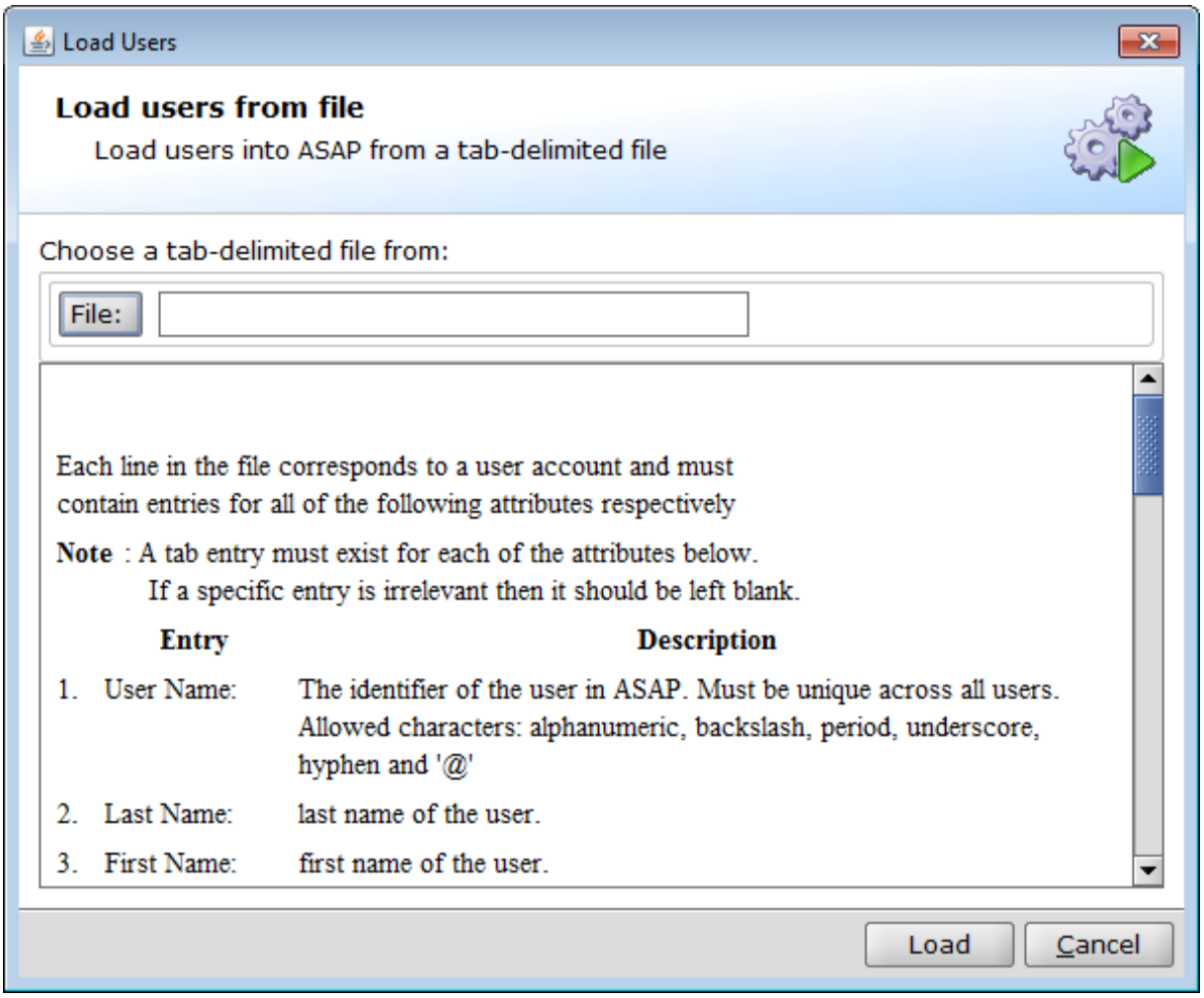
2 In the **Users** panel, click **Users** list.

The load user method menu opens.



3 Select **Load from File**.

The **Load Users** window opens.



4 Click **File** to open an Open dialog box for selecting an input file.

The Open dialog box opens.

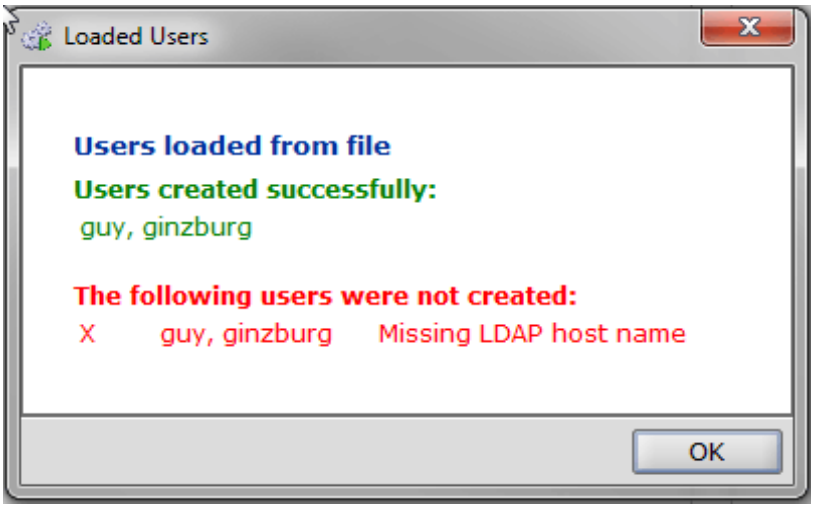
5 Select an input file and click **Open**.

Control returns to the **Load Users** window.

6 Click **Load** to load the selected file.

The load process begins.

7 When the load completes, the **Loaded Users** message will display with detailed information about the load:



8 Click **OK** to close the Loaded Users window.

Importing Users from LDAP

Users can easily be imported from an LDAP directory to Release Automation. The process consists of:

1 Browsing an LDAP server for a list of users.

2 Selecting the LDAP users that should be added to the Release Automation system.

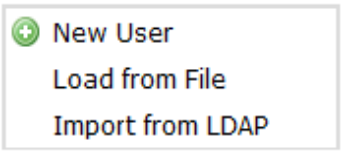
Note: Before you start, contact your Serena administrator for the list of attributes required to connect and browse the LDAP directory. Or, use an external LDAP browser to look up LDAP attribute values.

To import users from an LDAP directory:

1 In the **Administration** tab of the **Navigation Panel**, click **Users Management**.

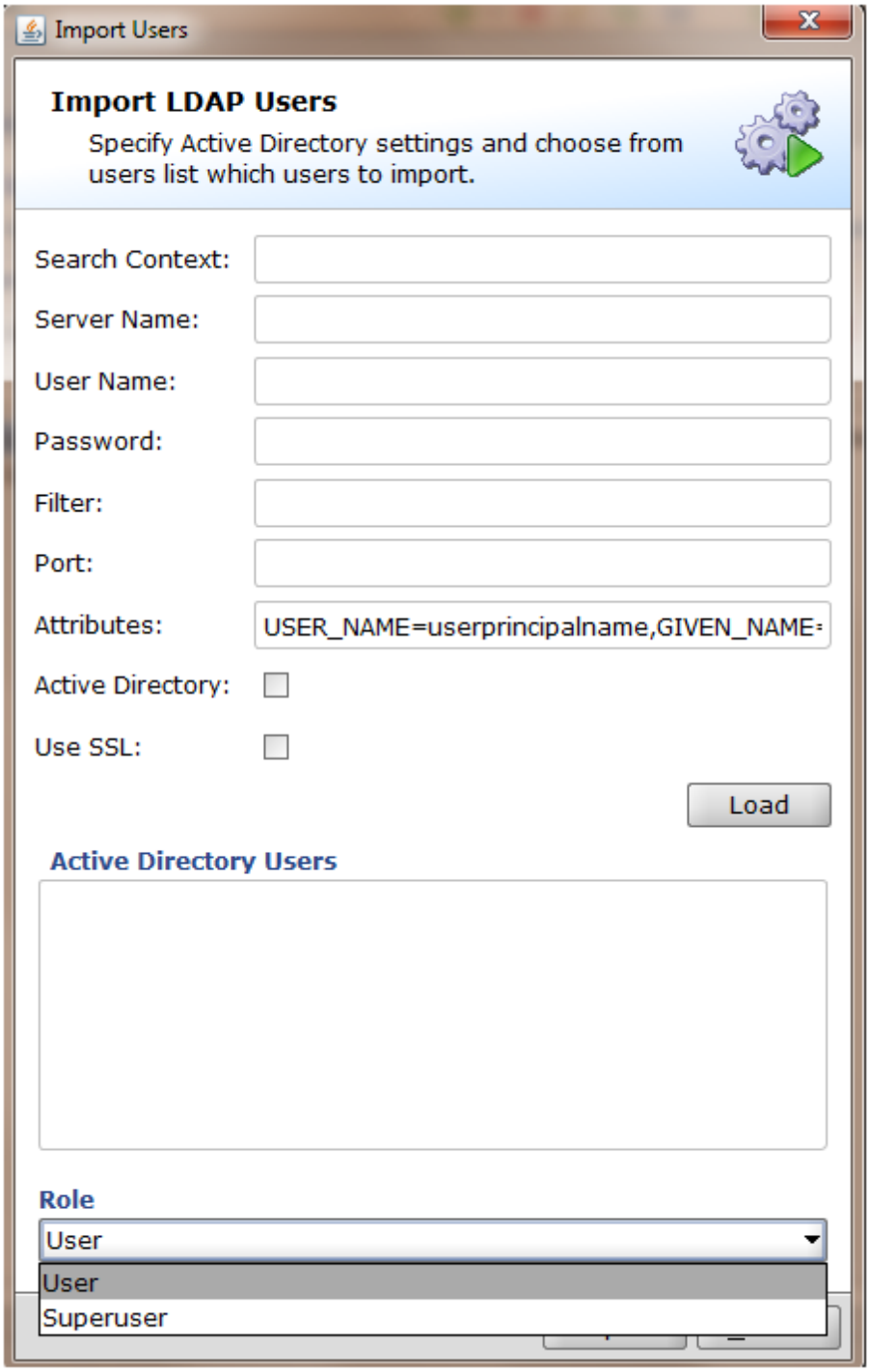
2 In the **Users** panel, click  in the Users list.

A selection menu opens:



3 Select **Import from LDAP**.

The **Import Users** window opens.



4 Enter the field data as required according to the LDAP Browse Criteria Table (on page 68).

5 Click **Load** to import the users meeting the search criteria.

Users meeting the browse criteria will display in the Active Directory Users pane.

6 To accept the list of loaded users for importing:

a. In the **Role** list, select **LDAP User**.

b. Click **Import**.

7 To adjust the browsing criteria and start over, click **Cancel**.

8 Click **Save** to store the imported users.

Note: Custom LDAP accounts are verified when the user logs in, not at account creation.

LDAP Browse Criteria Table

Field	Description	Example
Search Context	LDAP directory path to search for the users. The Search Content path is used during user authentication.	DC=mycompany
Server Name	Resolvable alias or address of the domain controller. The Server Name value is used during user authentication.	
User Name	Security context of the browsing user that will be used to connect to and browse the directory.	admin@mycompany.com
Password	Password of the browsing user.	
Filter	LDAP query that will be used to filter the users that reside in the subtree under the search context. If blank, all users are returned.	userprincipalname=*@mycompany.com
Port	Enter ports that the server uses for the LDAP/LDAPS protocols. If blank, defaults will be used. The Port value is used during user authentication.	
Attributes	Enter the descriptor for the attributes that will be looked up in the LDAP user entry. The attributes should be provided as a comma-delimited list of pairs: USER_NAME=uid,GIVEN_NAME=givenname,SURNAME=sn,EMAIL=email,USERNAME_DECORATION=@mycompany.com See LDAP Attributes Table (on page 69).	
Active Directory	Indicates whether Active Directory standards should be used when looking up the users. In an Active Directory server using this attribute will reduce the number of returned records.	True
Use SSL	Indicates whether to use SSL (LDAPS) or not (LDAP) when connecting to the directory. The Use SSL setting is used during user authentication.	True

LDAP Attributes Table

Option	Description
USER_NAME	Required. Name of the attribute that stores the user's login name: USER_NAME=userprincipalname. The lookup value can be decorated using the USERNAME_DECORATION option: • For Active Directory, the user name attribute will usually be either userprincipalname or samaccountname. • For other LDAP servers, a common username attribute is uid.
GIVEN_NAME	Name of the attribute that stores the user's given name: GIVEN_NAME=givenname
SURNAME	Name of the attribute that stores the user's surname: SURNAME=sn
EMAIL	Name of the attribute that stores the user's email address: EMAIL=email
USERNAME_DECORATION	An expression that supports appending a prefix or suffix to the retrieved user name: USERNAME_DECORATION=mycompany.com\\ or USERNAME_DECORATION=@mycompany.com. Formats: • Prefixes always end with '\\'. • Suffixes always begin with '@'.

LDAP Groups

An option exists to import LDAP user groups into the Serena User Management console. LDAP organizations can preserve control of the user management process within their LDAP systems while allowing user access to Serena applications.

When a LDAP group user tries to connect to Serena, Serena will access the LDAP server for authentication and authorization.

When the LDAP group user is using Serena Release Automation, access to tabs, applications, and functions is controlled by the permissions assigned in Serena to the user's LDAP group.

Any user which is created in LDAP and assigned to a LDAP group which was imported to Serena will enjoy all Serena permissions assigned within Serena to the group. Domain users not part of the imported LDAP group will only view a blank screen in Serena.

To enable LDAP integration, system administrators need to manually update the `distributed.properties` file, located on the Serena Server machine under the webapps/datamanagement/WEB-INF folder. For implementation details, refer to Enabling LDAP Integration (on page 141).

Importing LDAP Groups

To import an LDAP group:

1 Verify that LDAP Integration is enabled (on page 141).

2 In the **Administration** tab of the **Navigation Panel**, click **Users Management**.

3 In the **Groups** panel, click .

A selection menu opens:

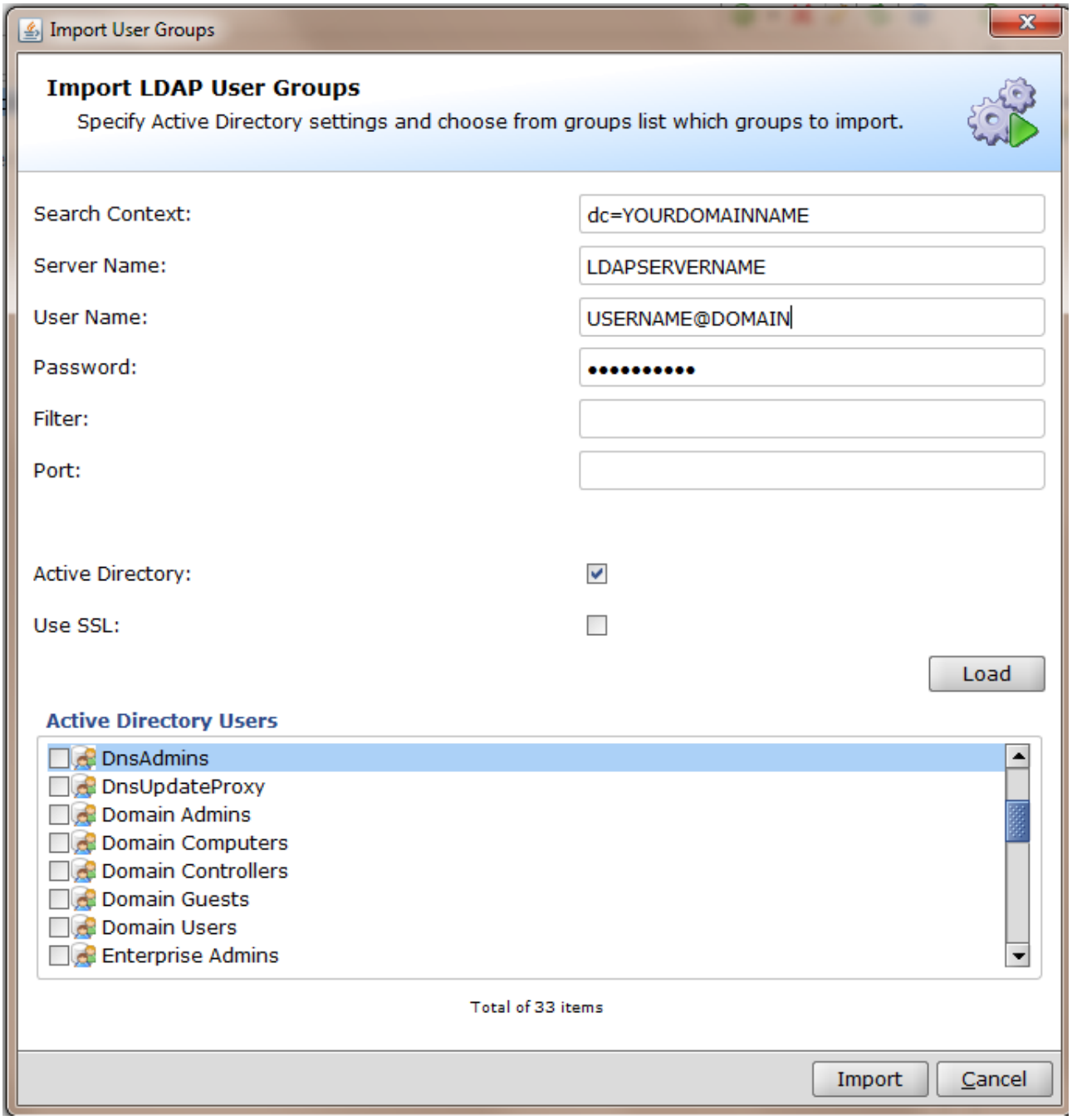
4 Select **Import from LDAP**.

The Import User Groups dialog displays.

5 Enter the field data as required, except for Attributes, according to the LDAP Browse Criteria Table (on page 68).

6 Click **Load** to import the users meeting the search criteria.

Users meeting the browse criteria will display in the Active Directory Users pane.



7 To accept the list of loaded users for importing, click **Import**.

8 To adjust the browsing criteria and start over, click **Cancel**.

9 Click **Save** to store the imported group.

10 Grant permissions to the imported group. See Granting Permissions (on page 73).

Modifying the Default Administrative User's

Details

During Serena Release Automation installation, a default administrative user called "superuser" is created. If necessary, you can change the default administrative user's details (on page 59).

Deleting Users

To delete a user from Serena Release Automation:

1 In the **Administration** tab of the **Navigation Panel**, click **Users Management**. The **Users Management** page opens.

2 In the **Users** list, select the relevant user, click , and confirm the operation. The **user** is deleted.

Adding and Editing User Groups

To add or edit a user group in Users Management:

1 In the **Administration** tab of the **Navigation Panel**, click **Users Management**. The **Users Management** page opens.

2 To **add** a new **user group**, click in the Groups list. A **New Users Group** window opens, where you can add the name for the group.

3 To **edit** an existing **user group**, select the relevant user group and click . The **Edit Users Group** window opens.

4 Complete the fields using the information in the following table.

5 Click **Apply**.

User Group Fields

Field	Description	Example
Group Name	Type a user name for the user group. In the **Edit Users Group** window, this field is read-only.	Operations Team
Description	Type a description of the user group.	This is a group for the operations team.

Deleting User Groups

To delete a user from Serena Release Automation:

1 In the **Administration** tab of the **Navigation Panel**, click **Users Management**. The **Users Management** page opens.

2 In the **Groups** list, select the relevant user group, click  , and confirm the operation. The **user group** is deleted.

Adding Users to User Groups

To add a user to an existing user group:

1 In the **Administration** tab of the **Navigation Panel**, click **Users Management**. The **Users Management** page opens.

2 In the **Users** list, select the relevant user.

3 In the **Groups** list, select the user group to which you want to add the user and click  . The user is added to the user group.

Note: An LDAP user group can be managed *only* from the LDAP server. Users cannot be added to an LDAP group from the User Management panel.

Removing Users from User Groups

To remove a user from a user group list:

1 In the **Administration** tab of the **Navigation Panel**, click **Users Management**. The **Users Management** page opens.

2 In the **Groups** list, under the relevant user group, highlight the user you want to remove and click  . The user is removed from the user group.

Granting Permissions

Permissions can be granted per **user** or **group** for operations on a **Server Group** and an **Application**. Within a permitted **Application**, permissions can be granted at the **Environment** and **Process** level.

The ability to perform operations within the permitted level is dependent on having the appropriate application role permissions. See Understanding Application Roles.

In the Permissions Management panel there are four types of icons representing users and groups:

-  - Serena user
-  - LDAP user
-  - Serena user group
-  - LDAP user group

Granting Permissions for Server Groups

To grant permissions on Server Groups:

1 In the **Administration** tab of the **Navigation Panel**, click **Permissions Management**. The **Permissions Management** page opens.

2 In the **Users and Groups** list, select the **user** or **user group** for which you want to grant permissions.

3 In the **Permissions** area, click the **Server Groups** tab. The **Server Groups** tab displays a tree containing all defined server groups and the servers they contain.

4 Select the check box next to each server group on which the **user** or **user group** should be granted permissions and click **Save.** The user or user group are granted with permissions on the servers belonging to the selected server group.

Granting Permissions for Applications

To grant, view, or modify application level permissions, the administrative user should have superuser role.

Note: In order to grand Environment permissions, it is necessary to grant Application permission for the target Environment first.

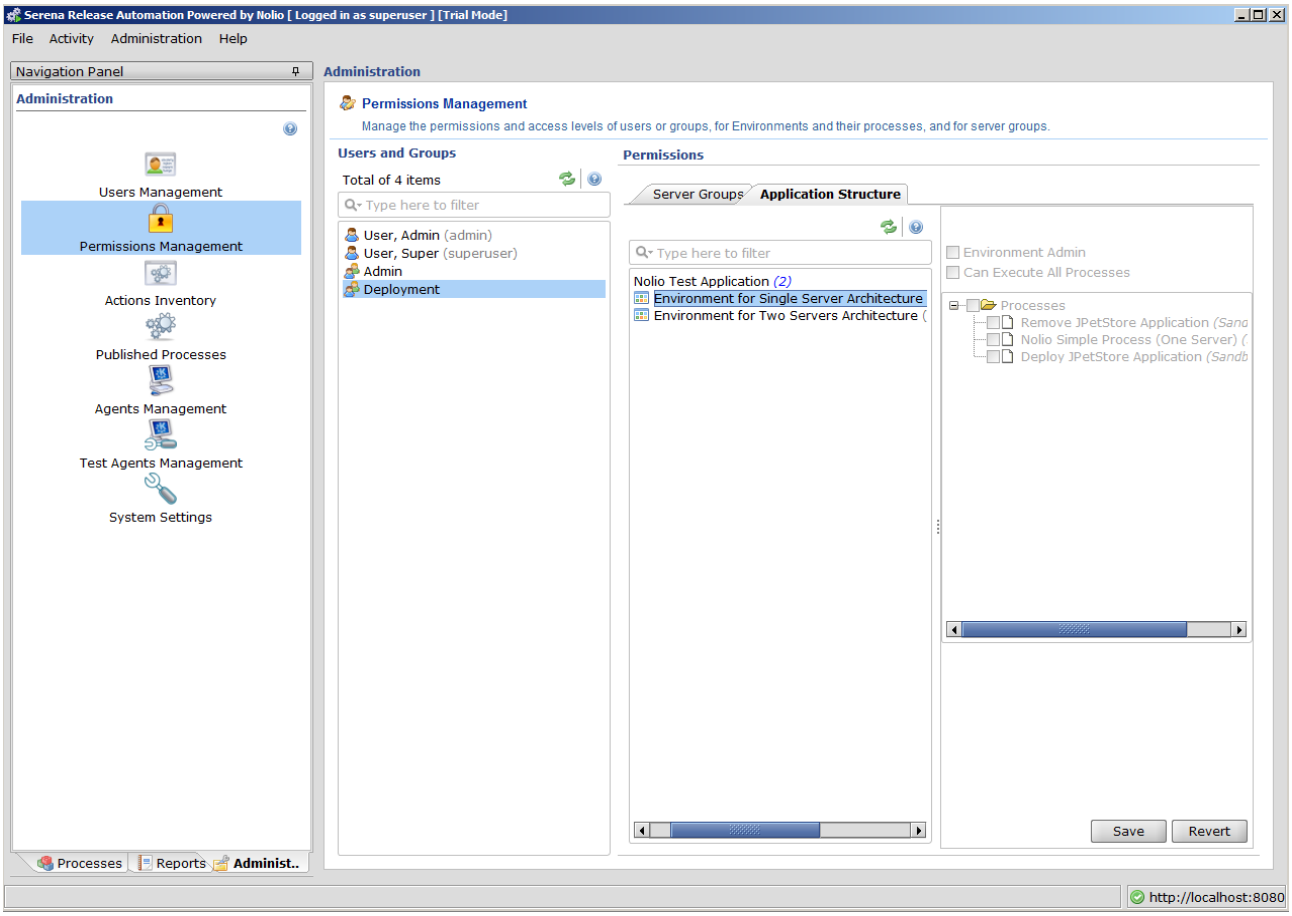
To grant application level permissions

1 In the **Administration** panel, select **Permissions Management**.

2 In the **Users and Groups** panel, select the target user or group.

3 In the **Permissions** panel, select the **Application Structure** tab.

The application and environment tree will display.



4 To allow the user to view an application:

a. Select the target application.

The **Can View Application** check box will display in the top of the right panel.

b. Select the **Can View Application** check box.

5 Click **Save** to store the selections, or **Revert** to cancel selections.

6 For information on application roles and additional permissions, refer to Understanding Application Roles.

7 To allow the user to execute selected processes or administer an application environment, refer to Granting Permissions for Environments (on page 75).

Granting Permissions for Environments

Note: In order to grant Environment permissions, it is necessary to grant Application permission for the target Environment first. See Granting Permissions for Applications (on page 74).

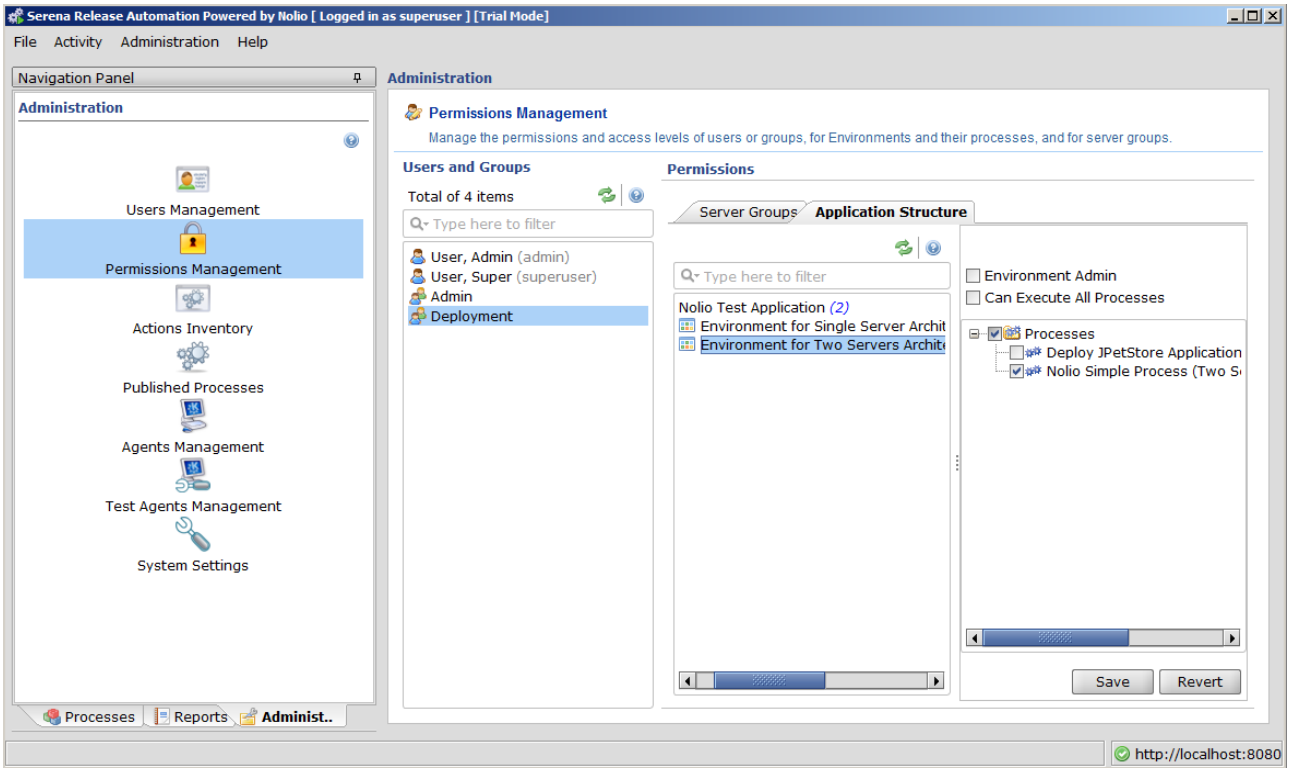
To grant permissions on Environments:

1 In the **Administration** tab of the **Navigation Panel**, click **Permissions Management**. The **Permissions Management** page opens.

2 In the **Users and Groups** list, select the **user** or **user group** for which you want to grant permissions.

3 In the **Permissions** area, click the **Application Structure** tab.

The **Application Structure** tab displays a tree containing all defined applications and environments.



4 Select the Application to which the target Environment belongs.

5 Verify that the **Can View Application** check box is selected for this Application.

a. If not, select the **Can View Application** check box.

b. Click **Save** to save the granted Application permission.

6 **Mark** the environment on which the **user** or **user group** should be granted permissions.

7 You may now grant the user or user group with:

a. **Environment Admin**: to enable the user or user group to **add, modify,** and **delete** all **processes** of the selected environment, to assign Server Types, and to manage instances and links. This also enables the user to **execute** all **processes** of this environment.

b. **Permissions on specific processes**: to enable the user or user group to **execute** the **checked processes** of the specific environment.

c. **Can Execute all Processes**: to enable the user or user group to **execute** all of the processes in the marked environment.

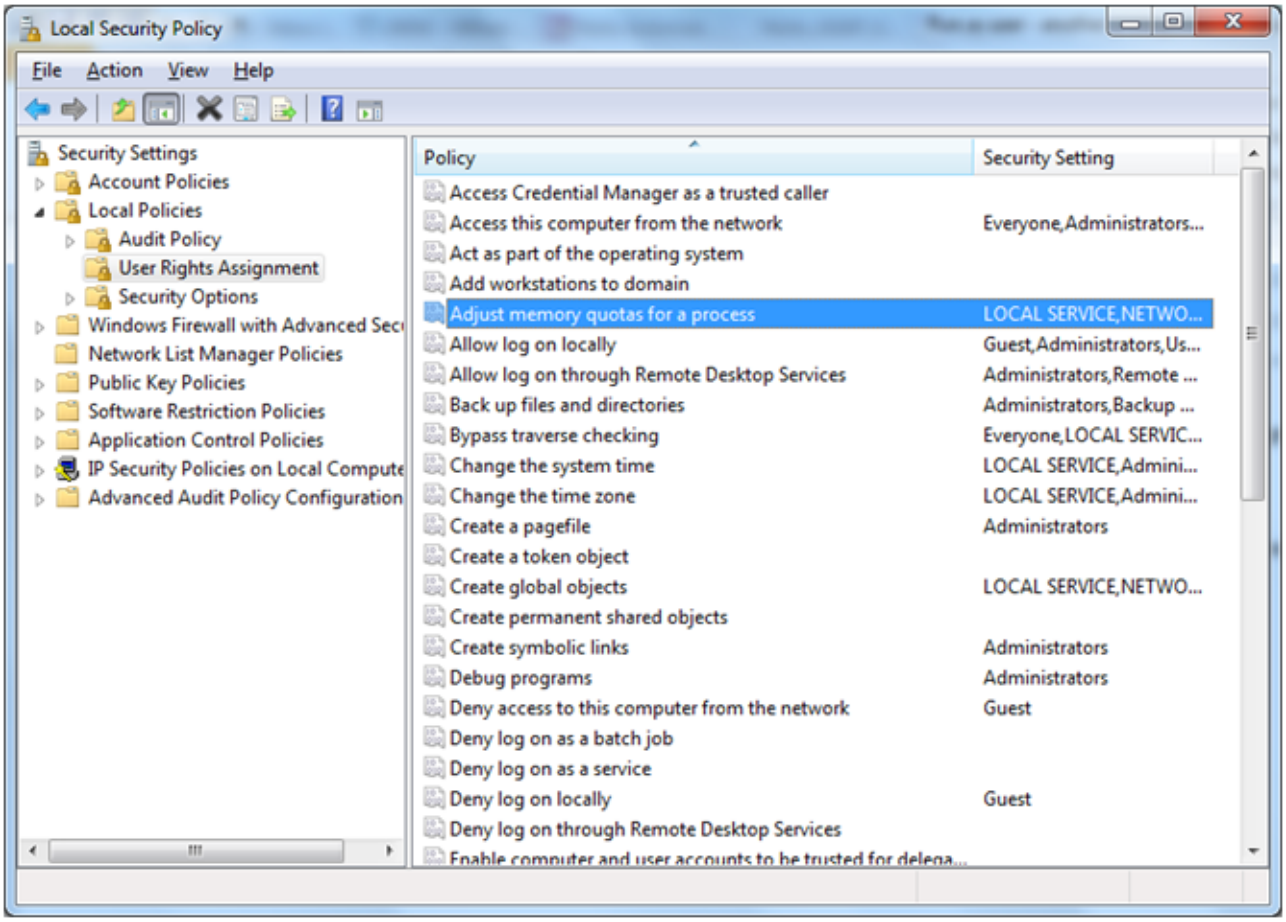
8 Click **Save**. The user or user group are granted with permissions on the processes belonging to the selected environments.

Assigning User Rights for 'Run as User' Option on Windows

On Windows, users who wish to run an action, process, or process subset, under the system privileges of a non-default user must have certain user rights assignments made.

Upon request, set the following assignments in Administrative Tools> Local Security Policy> Security Settings> Local Policies> User Rights Assignments for the non-default user:

- **Replace a process level token**
- **Adjust memory quotas for a process**



Chapter 9

Managing Agents and Test Agents

In This Chapter

This section explains how to add, edit, and delete **Agents** and **Test Agents**.

These tasks require you to have **superuser** role.

For information on roles, see Managing Serena Release Automation Users and Permissions (on page 55).

Overview

Serena Release Automation manages the automation of multi-tier applications, by channeling data and instructions to Serena Execution Servers installed at Data Centers. The Execution Servers then provide the information to the Serena Agents installed on each of the Data Center servers.

To enable Serena Release Automation to channel data to Execution Servers and agents, you must first define the Execution Servers and agents in Serena Release Automation.

Once you have defined the necessary Execution Servers and agents, you can organize the agents, by adding them to **agent groups**. Each **agent group** represents a group of agents that have commonality.

In addition, you can specify which of the defined agents will serve as **test agents**. **Test Agents** are used for testing actions, flows, and processes while modeling applications in Serena Release Automation.

In order for an agent to serve as a test agent, it must belong to the **test agents group**.

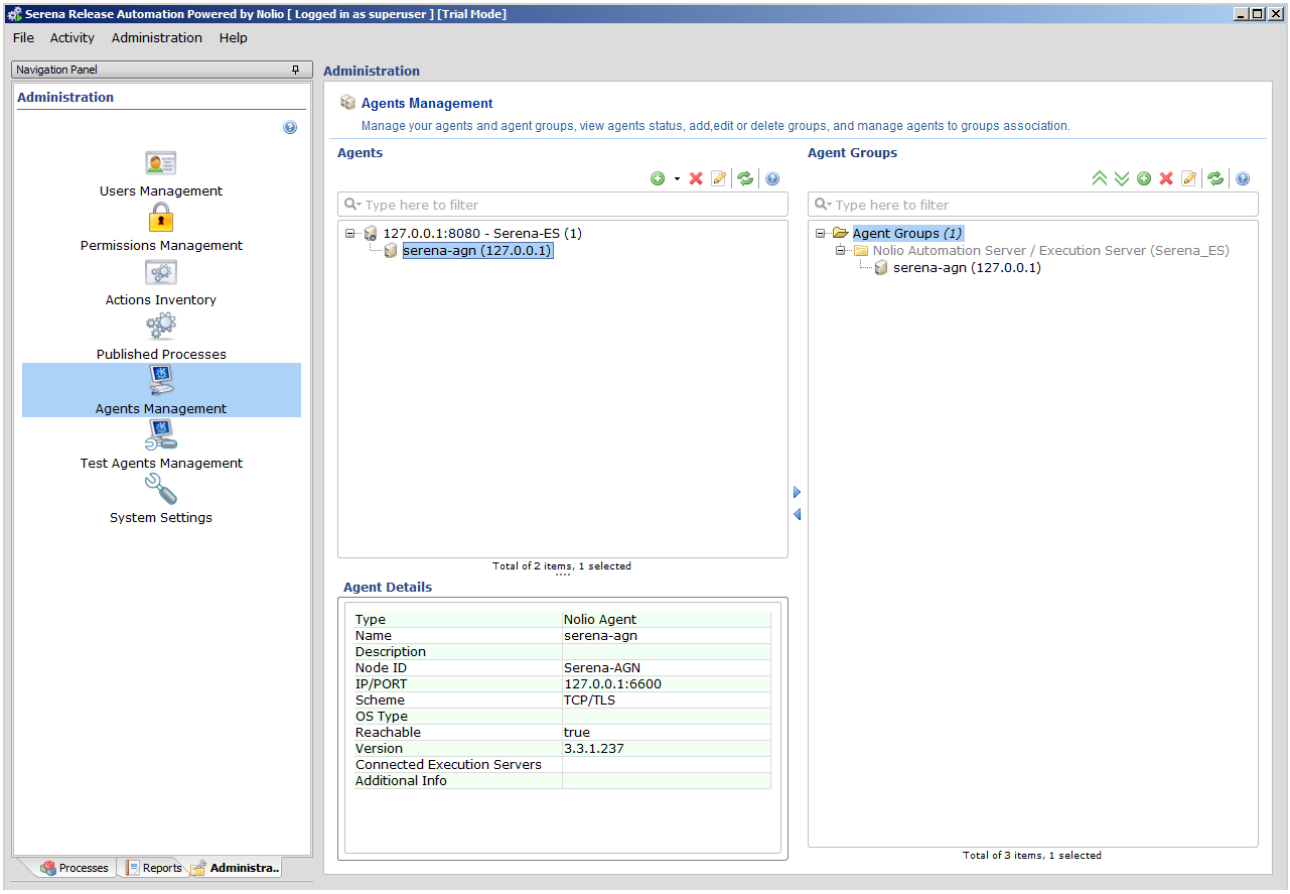
Execution Servers

This section describes how to add, edit, and delete Agent Execution Servers.

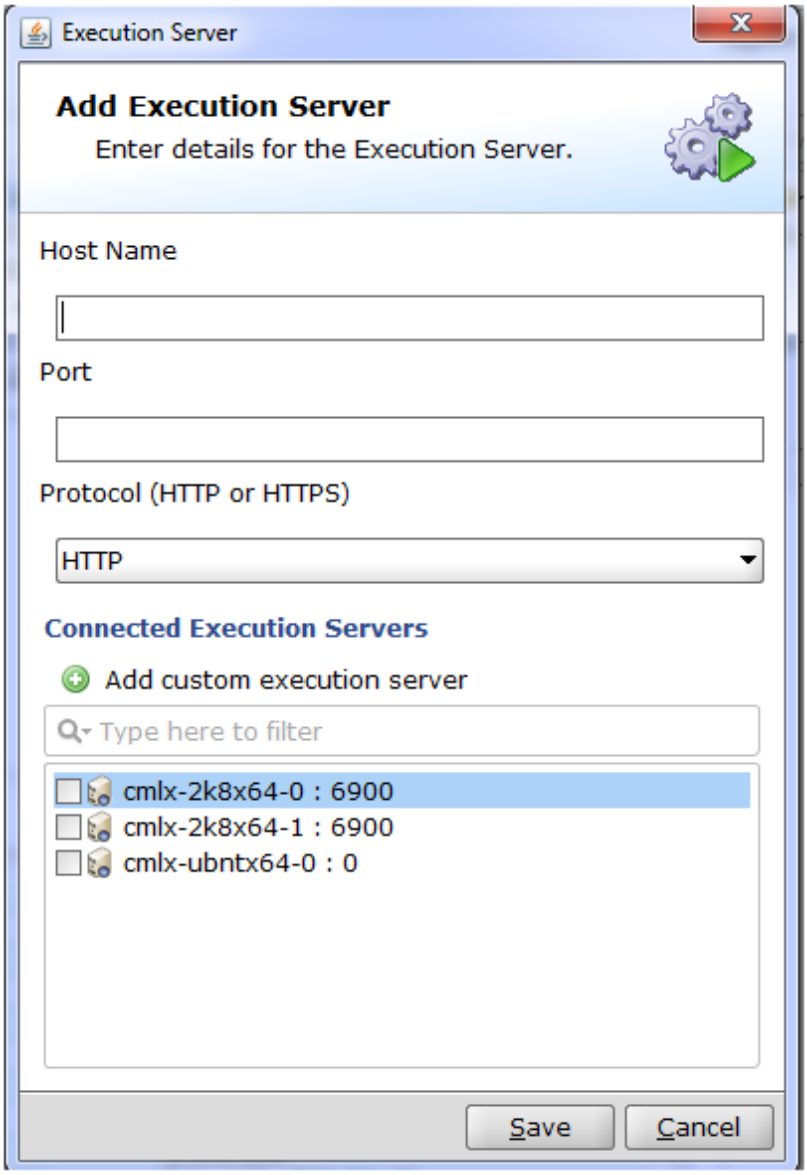
Adding and Editing Execution Servers

To add or edit an agent server:

1 In the **Administration** tab of the **Navigation Panel**, click **Agents Management**. The **Agents Management** page opens.



2 To **add** a new **Execution Server**, click in the **Agents** list, and then select **Execution Server**. The **Add Execution Server** window opens.



3 Complete the fields using the information in the following table and click **Save.**

4 Wait two minutes.

Execution Server Fields

Field	Description	Example
Host Name or IP	Type the host name of the Execution Server	Myhost1
Port	Type the port that should be used for communications between the Execution Server and the Serena Release Automation Data Management	8080

Field	Description	Example
Protocol (HTTP or HTTPS)	Select the protocol to use for communication with Serena Release Automation	HTTPS
Connect Execution Servers	Select the Execution Servers you want this Execution Server to connect to	

To edit an existing Execution Server:

1 Select the desired **Execution Server** from the list, and click  . The **Edit Execution Server** window opens.

2 Edit the required parameters and click **Save**.

Configuring an Execution Server

Typically, Serena Agents reporting to an Execution Server do not recognize other Agents that are reporting to other Execution Server in the system.

In configurations requiring Agents to communicate with Agents that are reporting to different Execution Servers, update the Execution Servers by creating sibling connections.

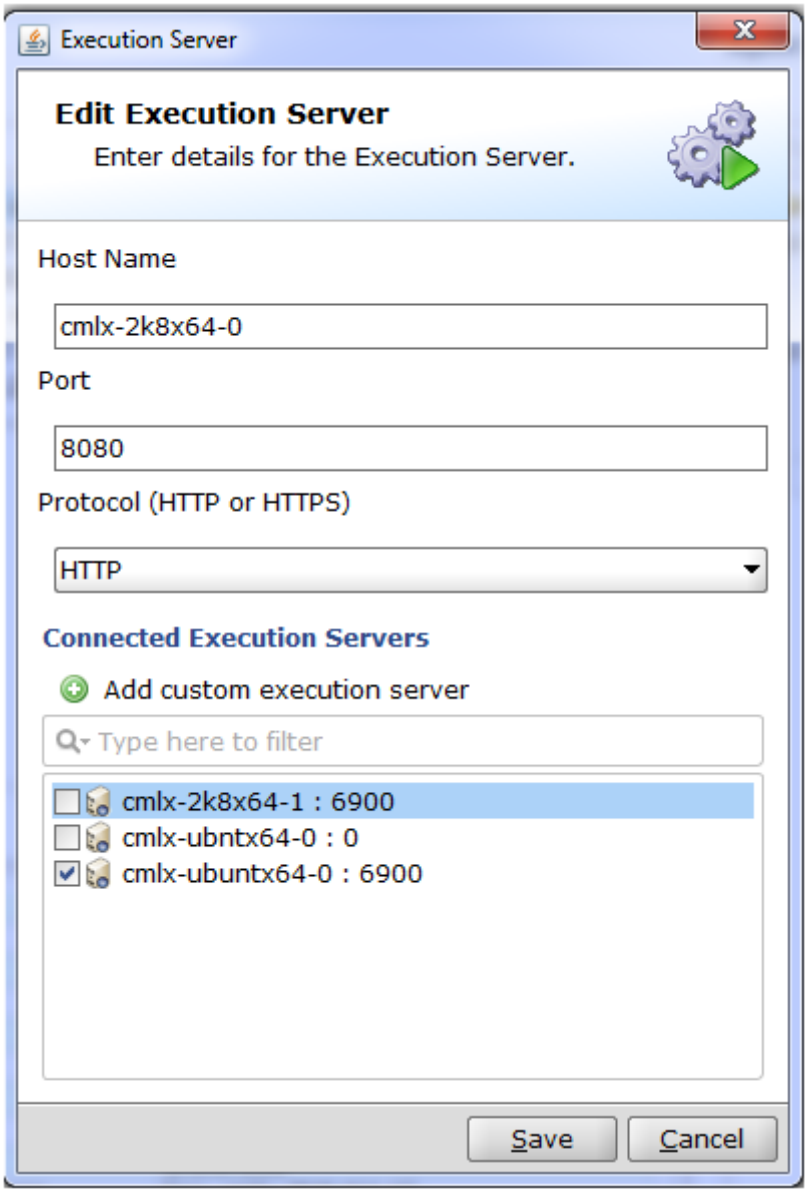
To configure an Execution Server to communicate with Agents reporting to different Execution Servers:

1 In the **Administration** tab of the Navigation Panel, click **Agents Management**.

2 Select the target Execution Server.

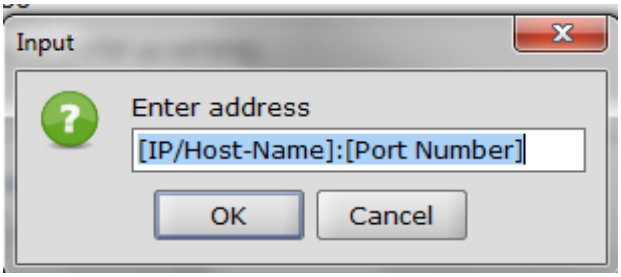
3 Right-click and select **Change execution server properties**.

The Edit Execution Server dialog displays.



4 Click  **Add custom execution servers**.

The address Input dialog displays.



5 Edit the Execution Server's Host-Name, or IP address, and Port Number following the specified format, for example, Serena-ES2:8080. The default port is 8080.

If the input does not conform to the specified format, a 'Malformed execution server Address message' will display.

6 Click **OK** to return to the Edit Execution Server dialog.

7 Click **Save** to save the configuration.

The Agent detail panel will include a list of the connected Execution Server.

Deleting Execution Servers

To delete an Execution Server:

1 In the **Administration** tab of the **Navigation Panel**, click **Agents Management**. The **Agents Management** page opens.

2 In the **Agents** list, select the relevant Execution Server and click . The **Execution Server** is deleted. No confirmation message displays.

Serena Agents

This section describes how to perform various operations on Serena Agents and Agent Groups, including adding, editing, viewing, deleting, installing, updating, and upgrading.

Deleting Agents

To delete an agent from the Serena Release Automation Client UI:

1 In the **Administration** tab of the **Navigation Panel**, click **Agents Management**. The **Agents Management** page opens.

2 In the **Agents** list, select the desired agent and click . The **Agent** is removed from the Client UI.

Note: No confirmation message appears.

If you do not uninstall an agent, it will continue to be detected by the system, even after you have removed it.

Viewing Execution Server and Agent Details

To view details for an Execution Server or an Agent:

1 In the **Administration** tab of the **Navigation Panel**, click **Agents Management**. The **Agents Management** page opens.

2 In the **Agents** list, select the relevant **Execution Server** or **Agent**. The **Component Details** pane displays information on the Execution Server or Agent, as described in the following table.

Execution Server Details Screen

Type	Nolio Execution Server
Name	cmlx-2k8x64-0
Description	Nolio Execution Server
Node ID	es_cmlx-2k8x64-0
IP/PORT	8080
Scheme	HTTP
OS Type	Windows Vista
Reachable	true
Version	3.3.0.156
Connected Execution Servers	[es_cmlx-2k8x64-1]
Additional Info	

Agent Details Fields

Field	Description	Example
Type	One of the following types of agent server or agent. • **Serena Execution Server** • **Serena Agent**	Serena Execution Server
Name	Name of the agent server or agent.	cmlx-2k8x64-0
Description	Description of the agent server or agent.	Serena Execution Server
Node ID	Node name of the agent server or agent.	es cmlx-2k8x64-0
IP/PORT	• When viewing an agent server, this field displays the port used for communications between the agent server and Serena Release Automation. • When viewing an agent, this field displays the IP address of the agent.	8080
Scheme	Protocol used for communication with Serena Release Automation.	HTTP
OS Type	Operating system of the agent server or agent.	Windows
Reachable	Boolean value indicating whether the agent server or agent is reachable.	true
Version	Version ID of the installed Serena Release Automation.	3.3.0.156
Connected Execution Servers	Execution Servers connected with this specific Execution Server	es cmlx-2k8x64-1
Additional Information	Additional Information.	Additional Information

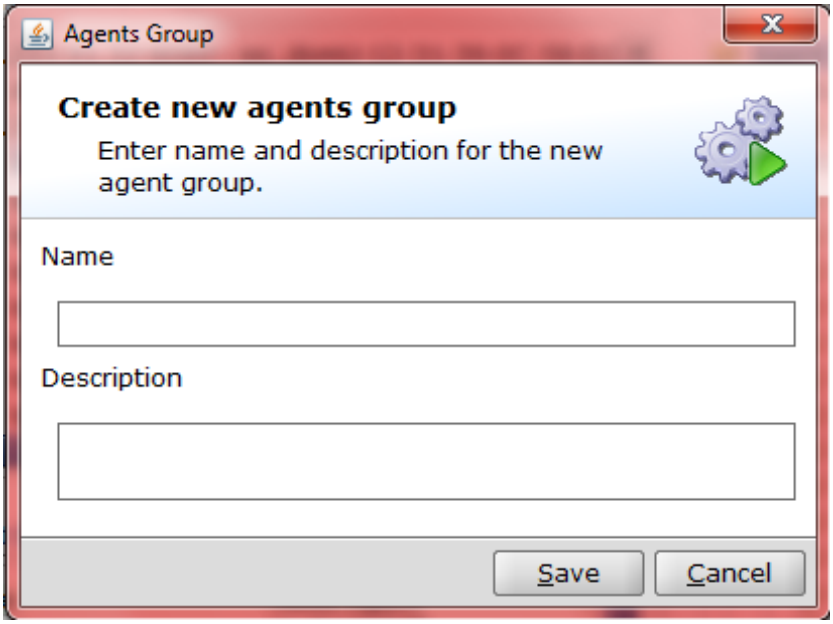
Adding and Editing Agent Groups

Agents are automatically placed in a group based on their Execution Servers. The agents may be moved into different groups.

To add or edit an Agent Group:

1 In the **Administration** tab of the Navigation Panel, click **Agents Management**. The **Agents Management** page opens.

2 To **add** a new **Agent Group**, click in the Agent Groups list. The **Create New Agents Group** window opens.



3 Complete the fields using the information in the following table and click **Save**.

Agent Group Fields

Field	Description	Example
Name	Type a name for the agent group	AgentGroup1
Description	Type a description for the agent group	First set of agents

To edit an existing agent user group:

1 Use the following icons to control the display of agents within the **Agent Groups** list:

a. Click to expand all Agent Groups.

b. Click to collapse all Agent Groups.

2 Select the relevant agent group in the **Agent Groups** list, and click . The **Edit Agents Group** window opens.

3 Complete the fields using the information in the Agent Group Fields table and click **Save**.

Deleting Agent Groups

To delete an Agent Group:

1 In the **Administration** tab of the **Navigation Panel**, click **Agents Management**. The **Agents Management** page opens.

2 In the **Agent Groups** list, select the relevant agent group and click  and confirm the operation. The **Agent Group** is deleted.

Adding Agents to Agent Groups

To add an Agent to an Agent Group:

1 In the **Administration** tab of the Navigation Panel, click **Agents Management**. The **Agents Management** page opens.

2 In the **Agents** list, select the relevant agent.

3 In the **Agent Groups** list, select the agent group to which you want to add the agent and click  . The agent is added to the agent group.

Removing Agents from Agent Groups

To remove an agent from an Agent Group:

1 In the **Administration** tab of the **Navigation Panel**, click **Agents Management**. The **Agents Management** page opens.

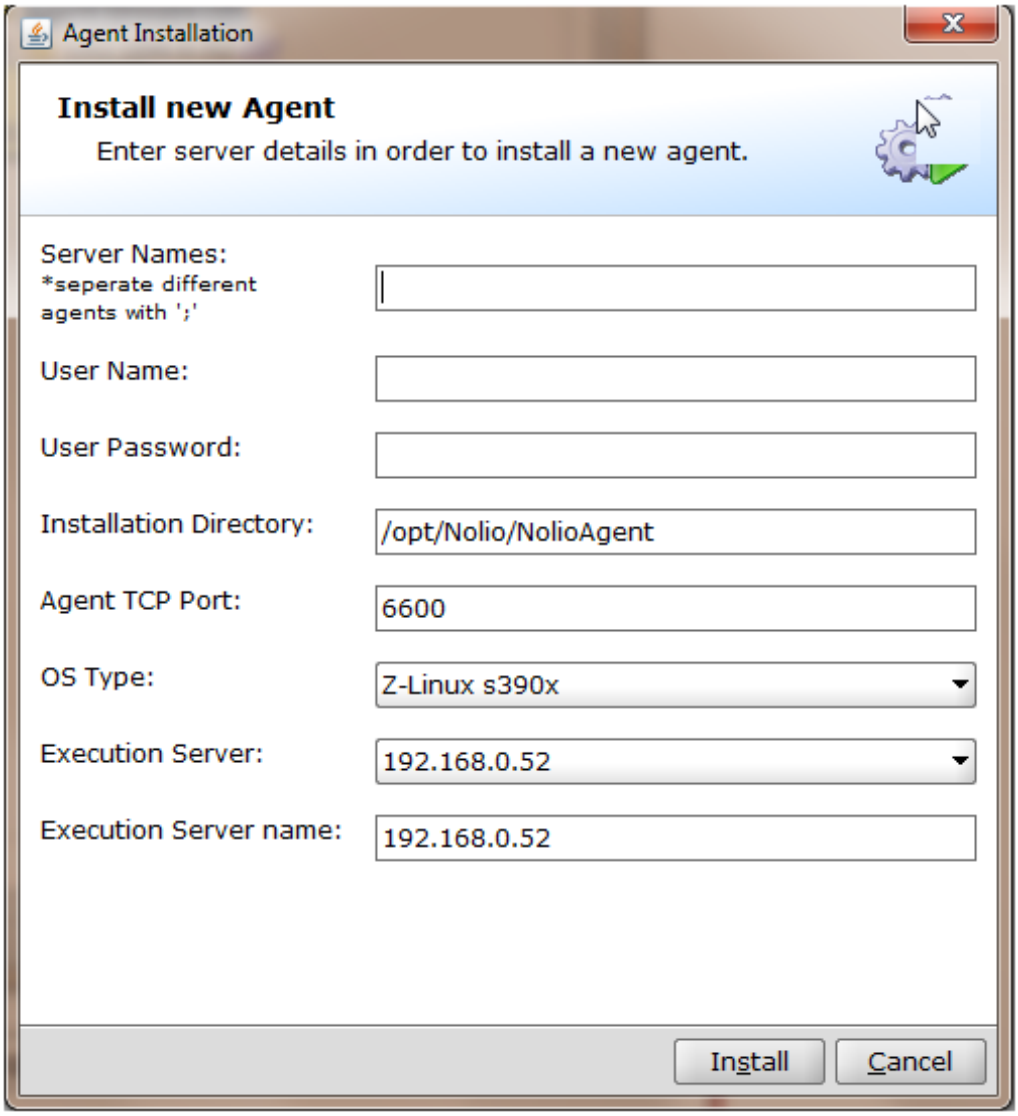
2 In the **Agent Groups** list, under the relevant agent group, select the agent you want to remove and click  . The agent is removed from the agent group.

Installing New Agents and Adding Agents to Execution Server in Remote Install

To install new agents and add them to an Execution Server:

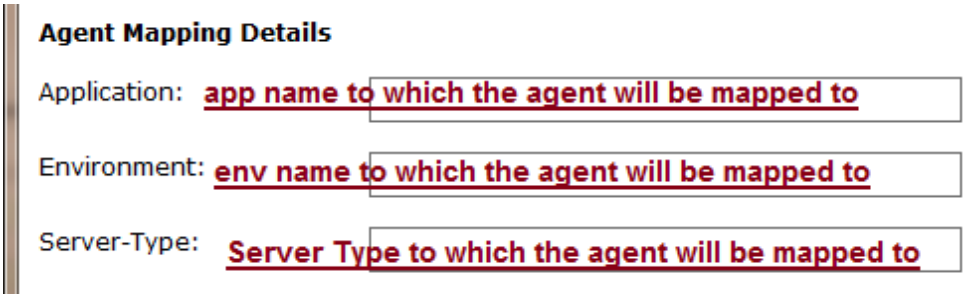
1 In the **Administration** tab of the **Navigation Panel**, click **Agents Management**. The **Agents Management** page opens.

2 Click  and from the dropdown menu select **Install Serena Agent on a Windows Server** or **Install Serena Agent on a Linux/Solaris Server (SSH)**. The **Agent Installation** window opens (depending on the selected platform).



3 Enter the new server required information. You may enter more than one server at a time, separating them with semicolons (;).

4 If the System Setting parameter **DYNAMIC_AGENT_MAPPING_ENABLED** is set to **TRUE**, the Install new Agent window will also display boxes for entering the Application, Environment, and Server Type to associate with the server agents.



5 Click **Install.**

Additional information on how to Install a New Agent can be found in Remote Agent Installation (on page 31).

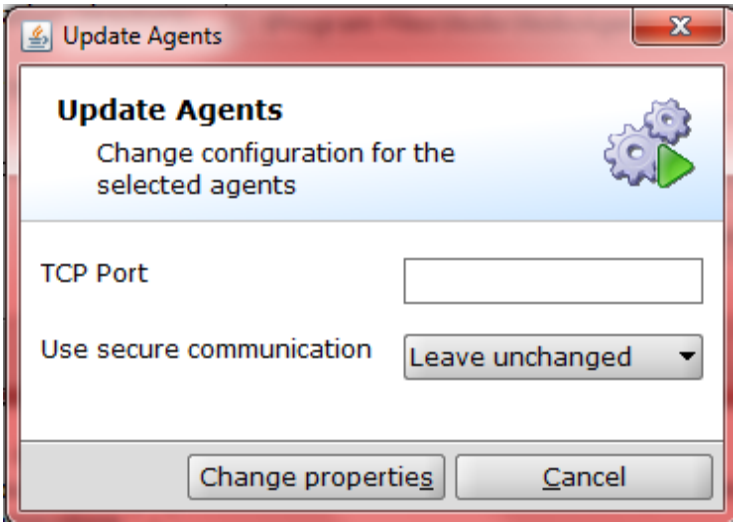
Updating Agents

To update configuration properties of an installed Agent:

1 In the **Administration** tab of the **Navigation Panel**, click **Agents Management**. The **Agents Management** page opens.

2 Select and right-click the agent you want to update.

3 Select **Change Properties of Selected Agents**. The **Update Agents** window opens.



4 Fill in the parameters that you want to change and click **Change properties.** Fields left blank will keep their current settings.

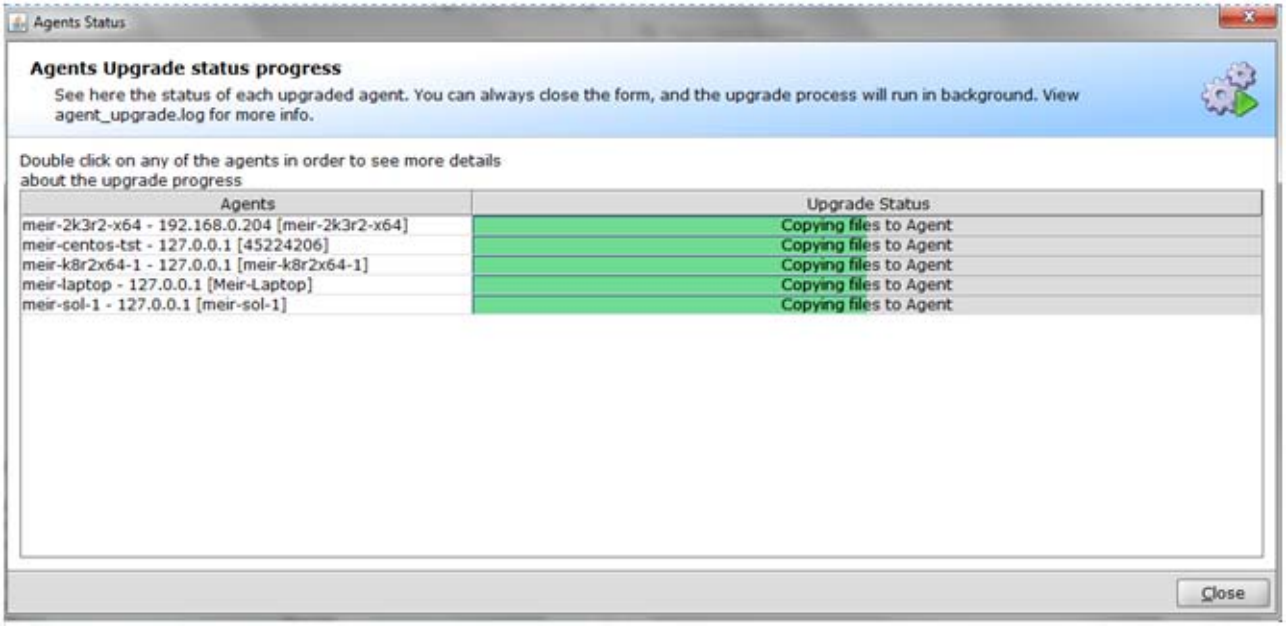
Upgrading Agents

This option will enable customers to upgrade their previous version of the Serena environment. Instead of upgrading each individual client, you may select all agents and upgrade them automatically.

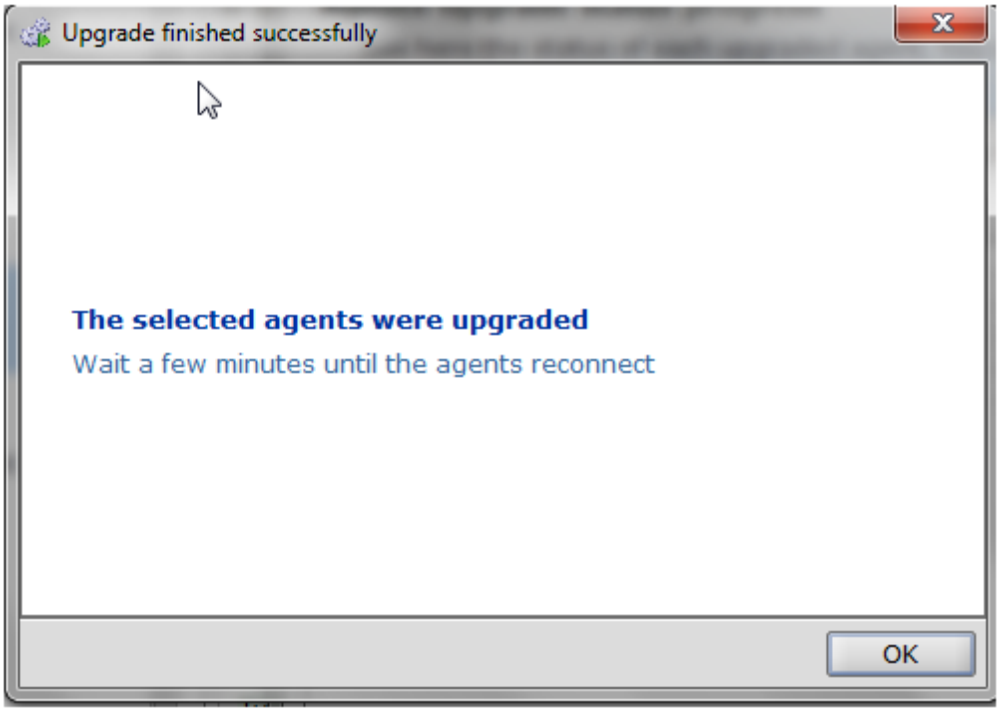
Important Note: Upgrading to Release Automation 4.1 from the UI is supported from versions 3.3.x only. For earlier versions, contact support@serena.com.

To upgrade agents:

1. In the **Administration** tab of the **Navigation Panel**, click **Agents Management**. The **Agents Management** page opens.
2. In the **Agent** list, select the **Agents** you want to upgrade.
3. Right-click and select **Upgrade selected agents**. All selected agents are upgraded to the newest version of the Serena environment.
4. When the Agents Upgrade process begins, an Agents Status window will open and display the upgrade progress. See Understanding the Upgrade Status Progress (on page 92).



5 When the Agents Upgrade process completes, an 'Upgrade finished successfully' message will display.



Important Note: The upgrade process of a previous Serena environment requires an Upgrade of all Serena components: Data management Server, Execution Server, and Serena Agent.

The upgrade should be performed as follows: Data Management Server, Execution Server(s) and Serena Agent(s).

The upgrading process for the Serena Data Management Server and Execution Server is explained separately in the zipped upgrade instructions file.

Understanding the Upgrade Status Progress Display

The list of agents being upgraded displays in the left column. The progress and the status of the upgrade appear in the right column.

The progress is displayed by a green bar which moves across the column row as the upgrade process proceeds through the job steps. The sequence of possible successful job step messages is:

- Upgrade started
- Copying files to Execution Server
- Copying files to agent
- Restarting agent
- Upgrade succeeded

If an upgrade step fails, an error message for the step will display in the status column and the cell will have a red background. The likely sequence of failure messages is:

- Failure to copy file to Execution Server
- Failure to copy files to agent
- Failed restarting agent
- Upgrade failed

If the upgrade includes multiple agents and fails, an additional popup will display with the message 'At least 1 agent failed to upgrade'.

At any time during the upgrade process, you may click on a row and a new popup will display a message with additional information.

You can close the Agents Status window during the upgrade process. The job will continue to run in the background.

Serena Test Agents

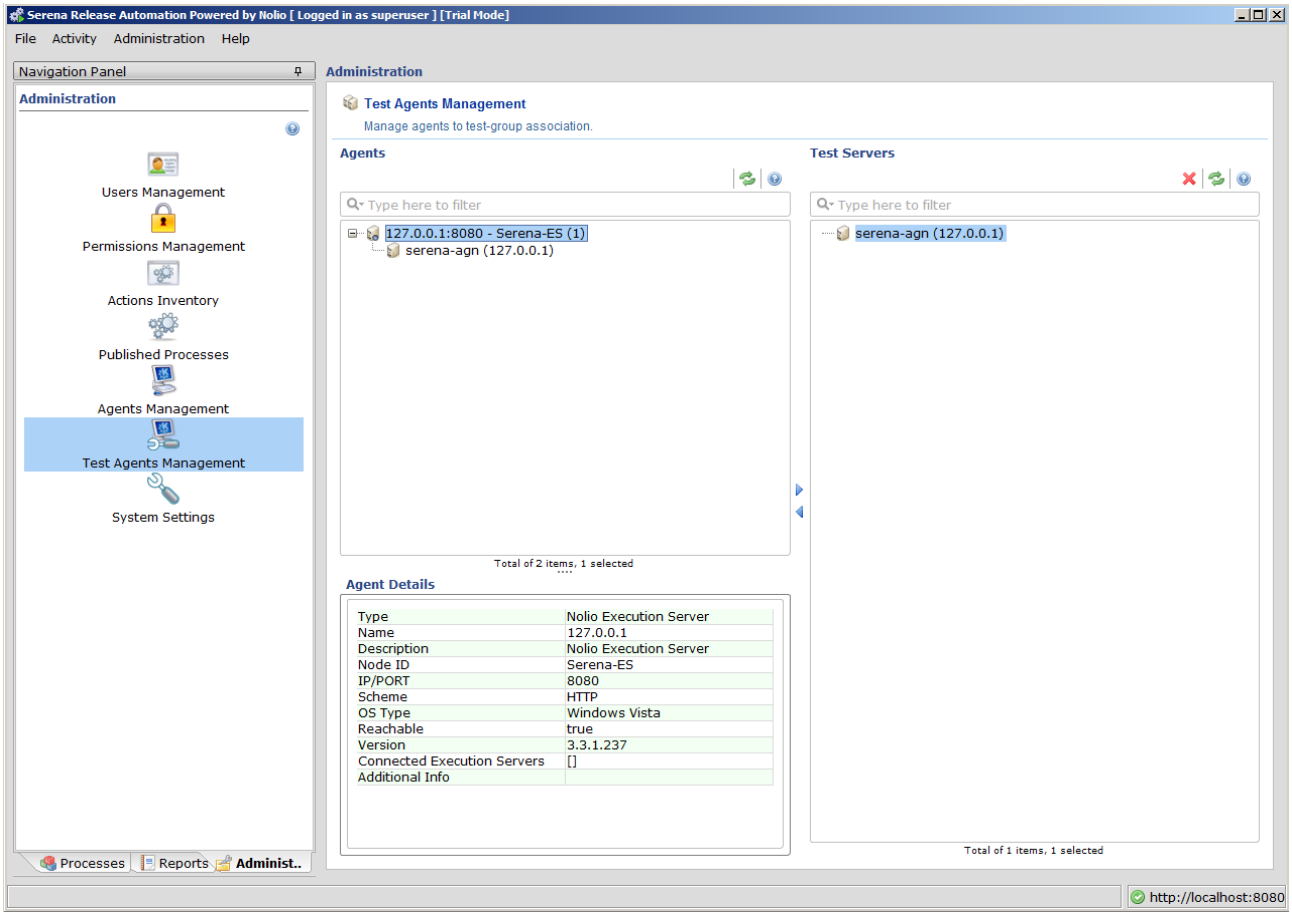
This section describes how to add and remove agents from Test Agent Groups.

Adding Agents to the Test Agents Group

Test Agents are used for testing actions, flows, and processes while modeling applications in Serena Release Automation.

To assign an Agent to Test Agents group:

1 In the **Administration** tab of the **Navigation Panel**, click **Test Agents Management**. The **Test Agents Management** page opens.



2 Select an agent from the **Agents** list, and click . The agent is added to the test agents group.

Removing Agents from the Test Agents Group

To remove an Agent from the Test Agents group:

1 In the **Administration** tab of the **Navigation Panel**, click **Test Agents Management**. The **Test Agents Management** page opens.

2 Select the agent you want to remove from the **Test Agents Group** list and click . The agent is removed from the test agents group.

Administration Tasks on Agents or Execution Servers

The following topics describe general administration tasks on Agents or Execution Servers.

Collecting Execution Server and Agent Logs

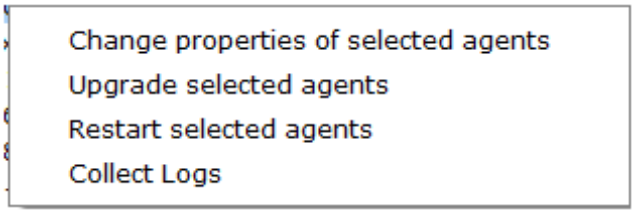
To collect logs from an Execution Server or Agent:

1 In the **Administration** tab of the Navigation Panel, click **Agents Management**.

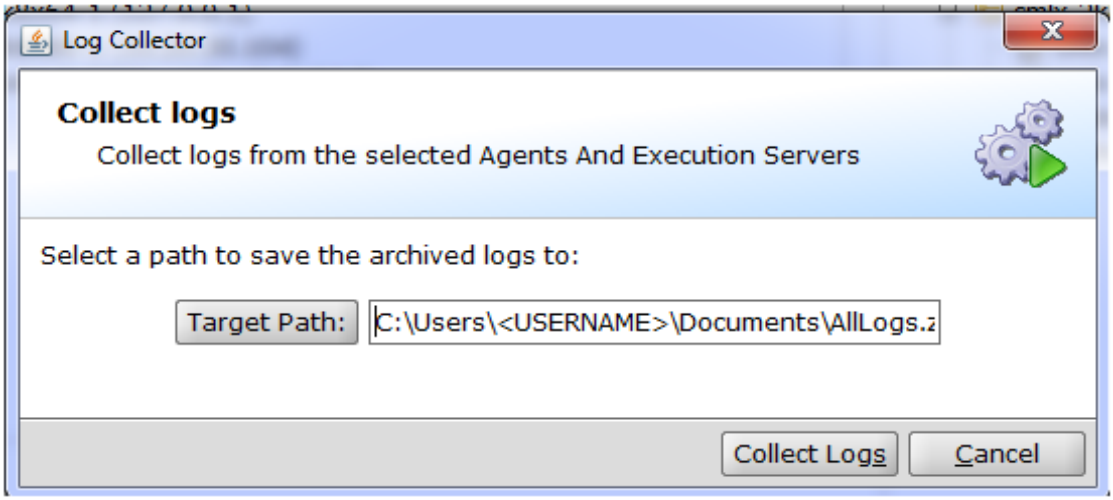
The Agents Management page opens.

2 In the **Agents** list, select an Execution Server or Agent.

3 Right-click and select **Collect Logs**.



The Log Collector dialog will display.



4 Click the **Target Path** button to open a browser to locate and select the path for saving archived logs; or in the **Target Path** box, enter the path.

5 Click the **Collect Logs** button to start the collection process.

Note: Collect Logs may take several minutes.

Restarting an Agent

From the Release Automation Client UI, you can restart a Serena Agent that is installed on a Windows machine.

To restart an Agent on a Windows machine:

1 In the **Administration** tab, click **Agents Management**.

2 Select a Serena Agent.

3 Right-click and select **Restart Agent**.

Note: The right-click restart agent option is **not** available on non-Windows machines.

Chapter 10

Managing Actions Inventory

In This Chapter

This section explains how to manage action templates in the Serena Release Automation actions inventory.

These tasks require you to have **superuser** role.

For information on roles, see Managing Serena Release Automation Users and Permission *(on page 55).*

Detailed information on individual actions can be found in the *Serena Release Automation Actions Reference Guide*.

Overview

When modeling a server-based application in the **Modeling** window, it is necessary to define actions that will be available for modeling the application's deployment.

Defining an action is performed by selecting a predefined **action template** from Serena Release Automation's **Actions Inventory** and then modifying the action as desired.

The **Actions Inventory** displays the action templates organized in **categories**, where each category represents a group of action templates that have in common usage. This enables you to quickly locate the action templates you need. Serena Release Automation automatically includes several predefined categories of action templates.

If desired, you can change the category to which a specific action template is assigned. You can also add new categories, modify existing categories, or delete categories as needed.

Together, the predefined action templates and action categories make up the **actions inventory**.

Serena Release Automation obtains the actions inventory from a default library that is provided with Serena Release Automation and is located on the Serena Center Server. The actions included in the default library are described in detail in the *Serena Release Automation Actions Reference Guide*.

If desired, you can purchase or implement libraries of custom action templates, and use these libraries instead of or in addition to the default library.

For instructions on implementing a custom actions library, refer to *Serena Release Automation Custom Actions SDK*.

New libraries must be loaded in Serena Release Automation. See Reloading Actions Libraries. (on page 103)

Note: When working with action packs, there may be preliminary configuration tasks to be performed prior to using the packs. These preliminary tasks are described in the *Serena Release Automation Action Reference Guide* under the specific action pack. For example, preliminary configuration tasks must be performed on each Serena Agent machine that will execute WebSphere or WebLogic actions.

Managing Existing Actions

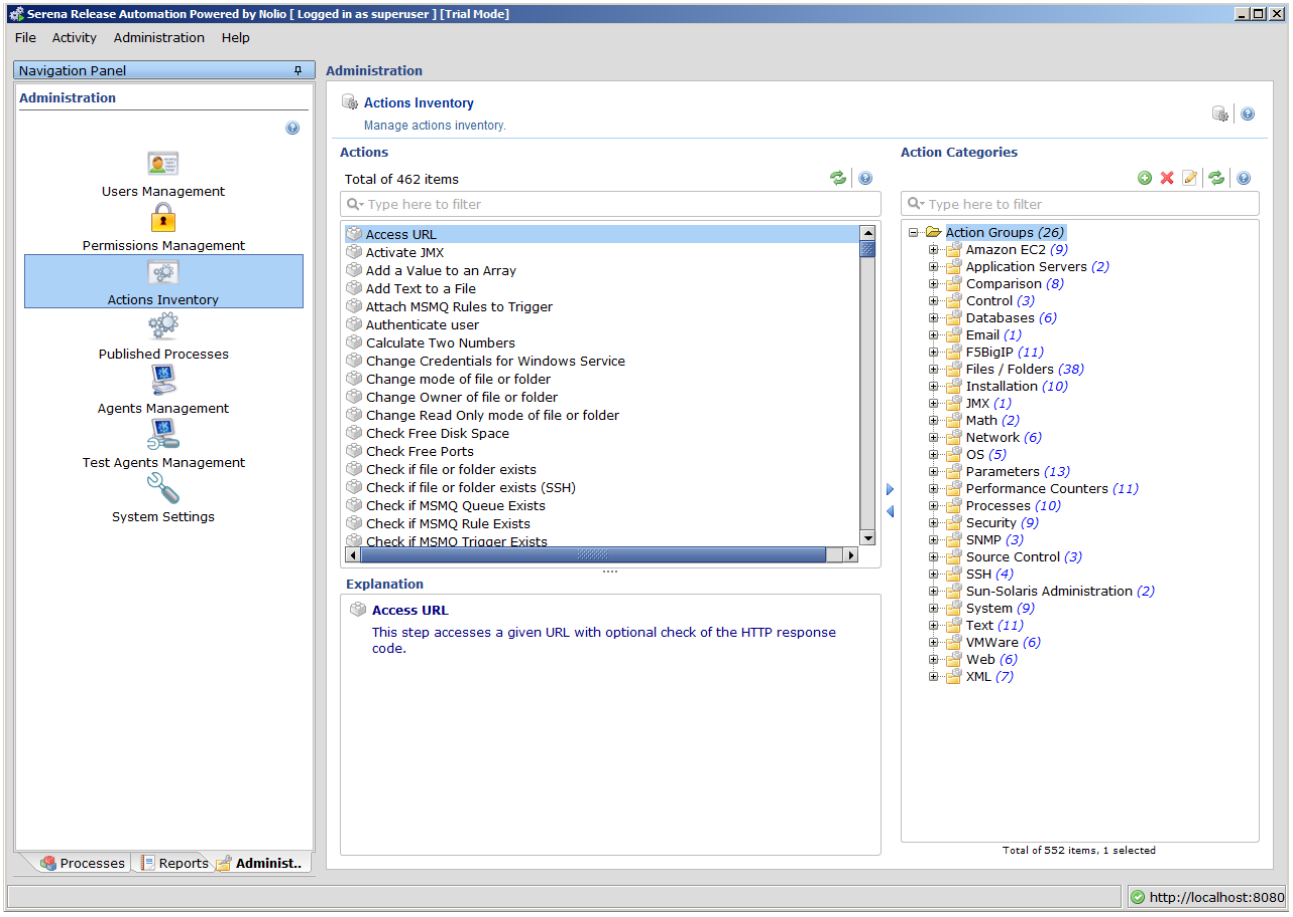
This section describes how to manage existing actions, including adding, editing, deleting action categories and adding and removing action templates from action categories.

Adding and Editing Action Categories

The actions templates are already divided into logical categories. If you prefer to group them differently, they may be moved into different categories.

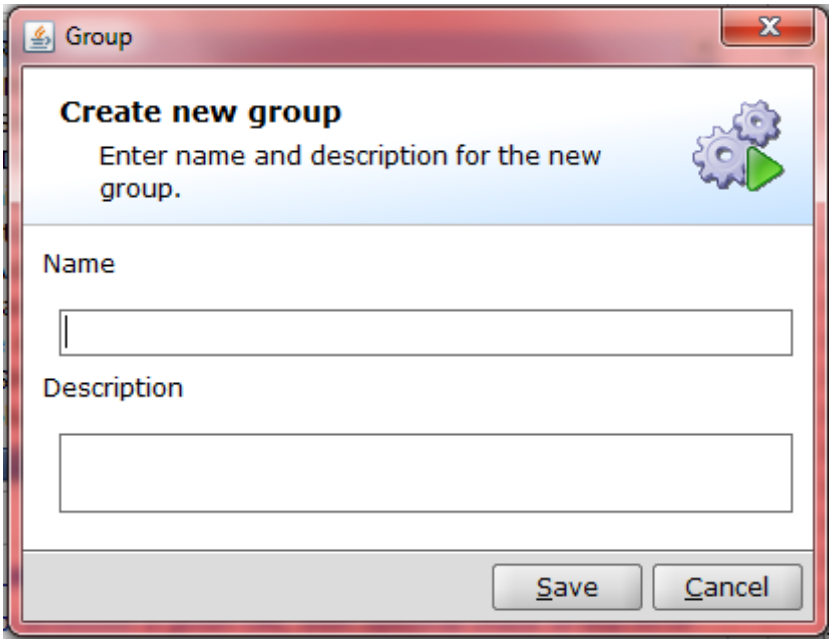
To add or edit an action category:

1 In the **Administration** tab of the **Navigation Panel**, click **Actions Inventory**. The **Actions Inventory** page opens.



2 To **add** a new category, select the relevant parent node in the **Action Categories** list, and click . The **Create new group** window opens.

3 Complete the fields using the information in the following table and click **Save**.



4 To **edit** an existing category, select the relevant category in the **Action Categories** list, and click  . The **Rename group** window opens.

5 Complete the fields using the information in the following table and click **Save**.

Action Category Fields

Field	Description	Example
Name	Type a name for the category	Preliminary Checks
Description	Type a description of the category	Actions related to preliminary checks

Deleting Action Categories

To delete an action category:

1 In the **Administration** tab of the **Navigation Panel**, click **Actions Inventory**. The **Actions Inventory** page opens.

2 Select the category you wish to **delete** from the **Action Categories** list.

3 Click  and confirm the operation. The category is deleted.

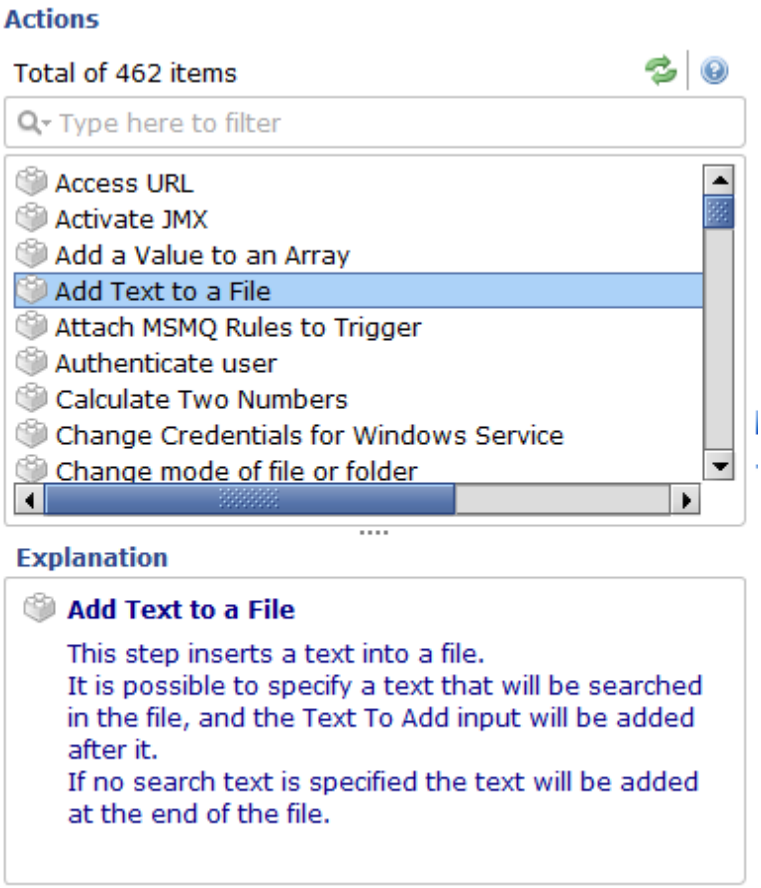
Note: If the category was deleted in error, see Reloading Actions Libraries (on page 103).

Adding Action Templates to Action Categories

To add a new Action Template to existing Actions Categories:

1 In the **Administration** tab of the **Navigation Panel**, click **Actions Inventory**. The **Actions Inventory** page opens.

2 Select the relevant action template from the **Actions** list. The **Explanation** pane, shown in the following figure displays a description of the selected action template.



3 In the **Action Categories** list, select the category to which you want to add the selected action template and click . The action template is added to the category.

Removing Action Templates from Action Categories

To remove an action template from an action category:

1 In the **Administration** tab of the **Navigation Panel**, click **Actions Inventory**. The **Actions Inventory** page opens.

2 In the **Action Categories** list, expand the relevant category, and select the action template you want to remove from this category.

3 Click . The action template is removed from the category.

Adding Actions

Customers can create custom actions to the Serena Release Automation action pack sets using the *Serena Release Automation Custom Actions SDK*.

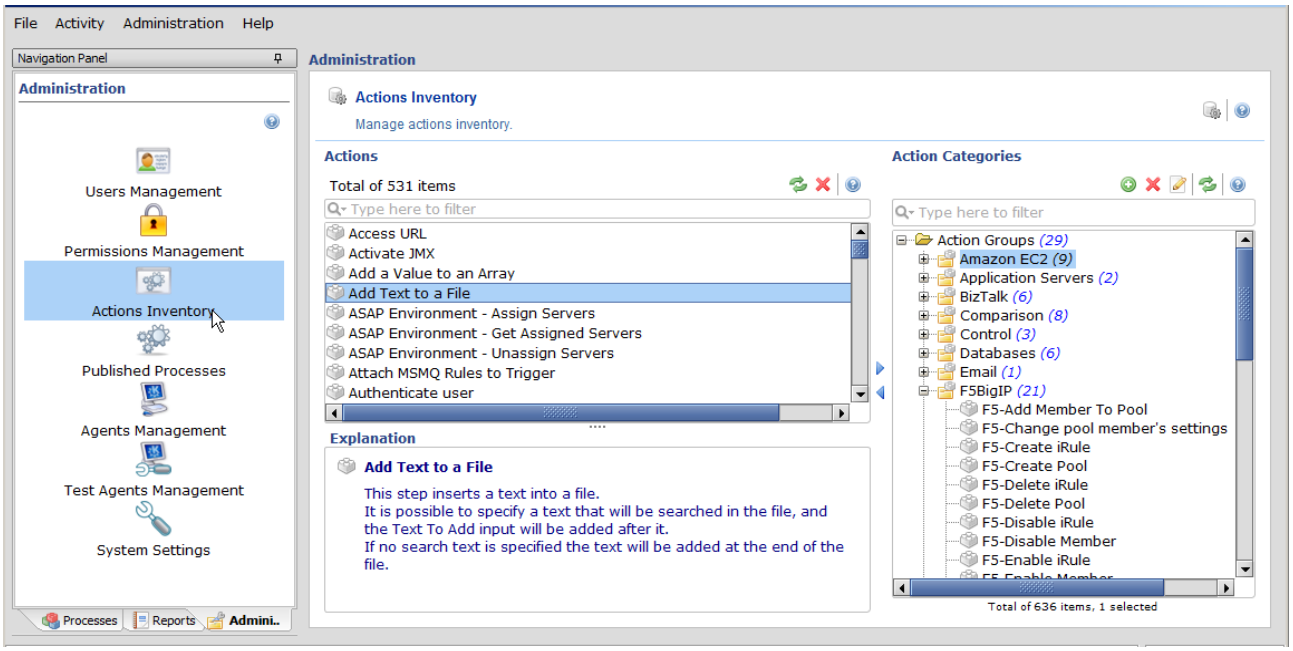
Actions, other than those provided by Serena, are defined in a **customerActions** library.

CustomerActions libraries are loaded into Serena Release Automation using the Manage Locations function. See Reloading Actions Libraries (on page 103).

Deleting Actions

To delete an action:

1 In the **Administration** tab of the **Navigation Panel**, click **Actions Inventory**. The **Actions Inventory** page displays.



2 In the Actions pane, select the action to delete.

3 Click the delete icon.

The Delete Action confirmation dialog will display.

4 Click **Yes** to complete the deletion.

Managing Action Template Libraries

The action inventory is comprised of all action template libraries, including Serena-supplied libraries and customer-developed libraries, defined in Serena Release Automation.

To define a library, you must specify its location on the Serena Center Server. To develop a custom library, refer to *Application Release Automation Custom Action SDK*.

This section describes how to reload libraries.

Reloading Action Libraries

You can reload all of the defined action libraries to Serena Release Automation. This option is used when:

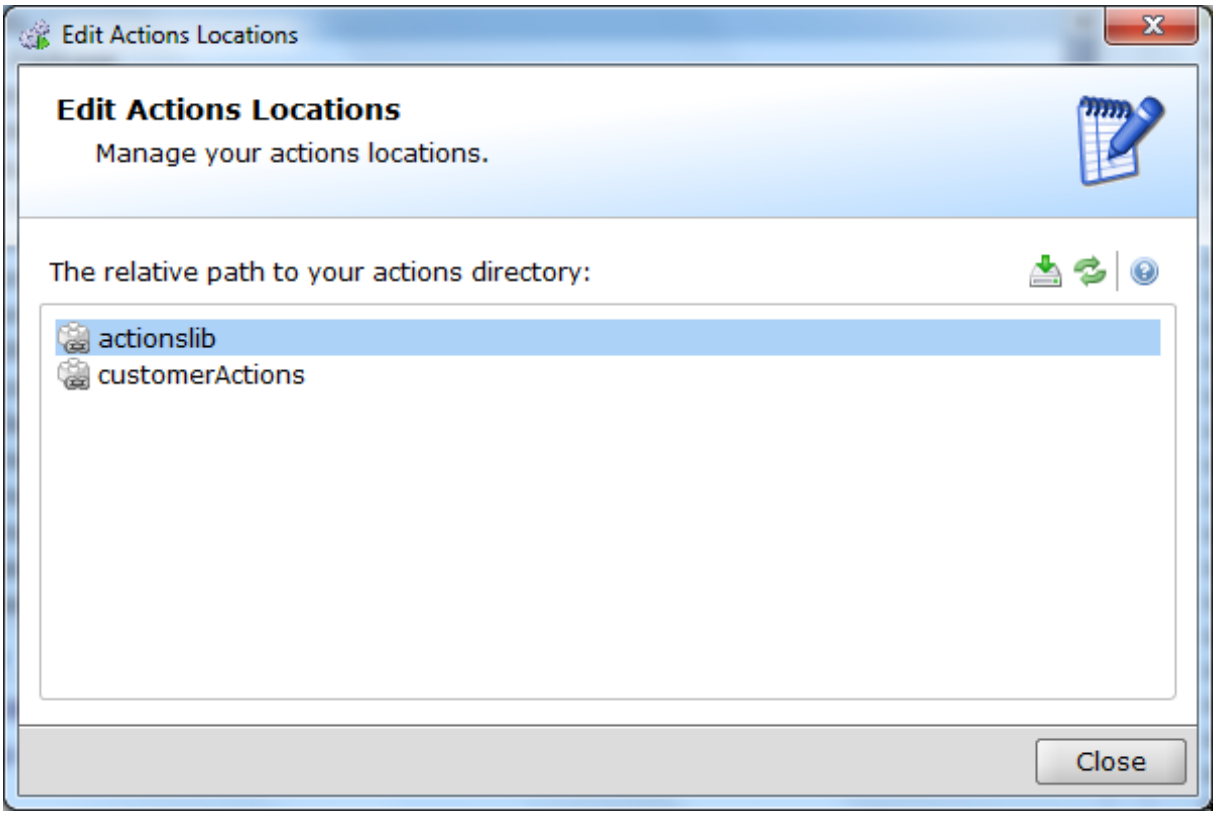
- New libraries with new templates have been received from Serena for the **actionslib**
- New custom templates have been added to the **customerActions** library
- After a system upgrade

Note: Customer-defined action libraries residing in the Serena Release Automation**/customerActions** folder are detected automatically by the Manage Locations function.

To reload all defined libraries:

1 In the **Administration** tab of the **Navigation Panel**, click **Actions Inventory**. The **Actions Inventory** page opens.

2 In the upper-right corner of the page, click . The **Edit Actions Location** window opens with a list of defined library locations.



3 Select the library, or libraries, to reload:

a. To reload a single actions library, select a library.

b. To reload multiple action libraries at once, hold the CTRL key down while selecting libraries.

4 Click . The selected libraries are reloaded to Serena Release Automation.

A message with the location of the currently loading library will display in the Edit Actions Locations will display.

The Loading actions library progress bar and message will display. At the end of a successful reload, the following message will display.

Operation completed successfully

5 Click **Close** to close the Loading actions library window.

6 Click **Close** to close the Edit Actions Locations dialog.

Chapter 11

Managing Published Processes

In This Chapter

This section explains how to view, promote, and delete published processes.

These tasks require you to have **superuser** role.

For information on roles, see Managing Serena Release Automation Users and Permissions (on page 55).

Overview

Once an application has been modeled in the **Modeling** window, the relevant processes must be **published** to the **Control** windows sandbox, where they can be assigned to an environment. The processes are then ready for implementation on your organization's servers.

As part of the working process with Serena Release Automation, the personnel that design and execute actions and flows are able to test their work using Test Agents. Before a process can be implemented, a user with **superuser** role must **promote** the process from the **sandbox** to **public** use.

The **sandbox** is usually referenced as an "internal" location which can be used by those who develop and test the process.

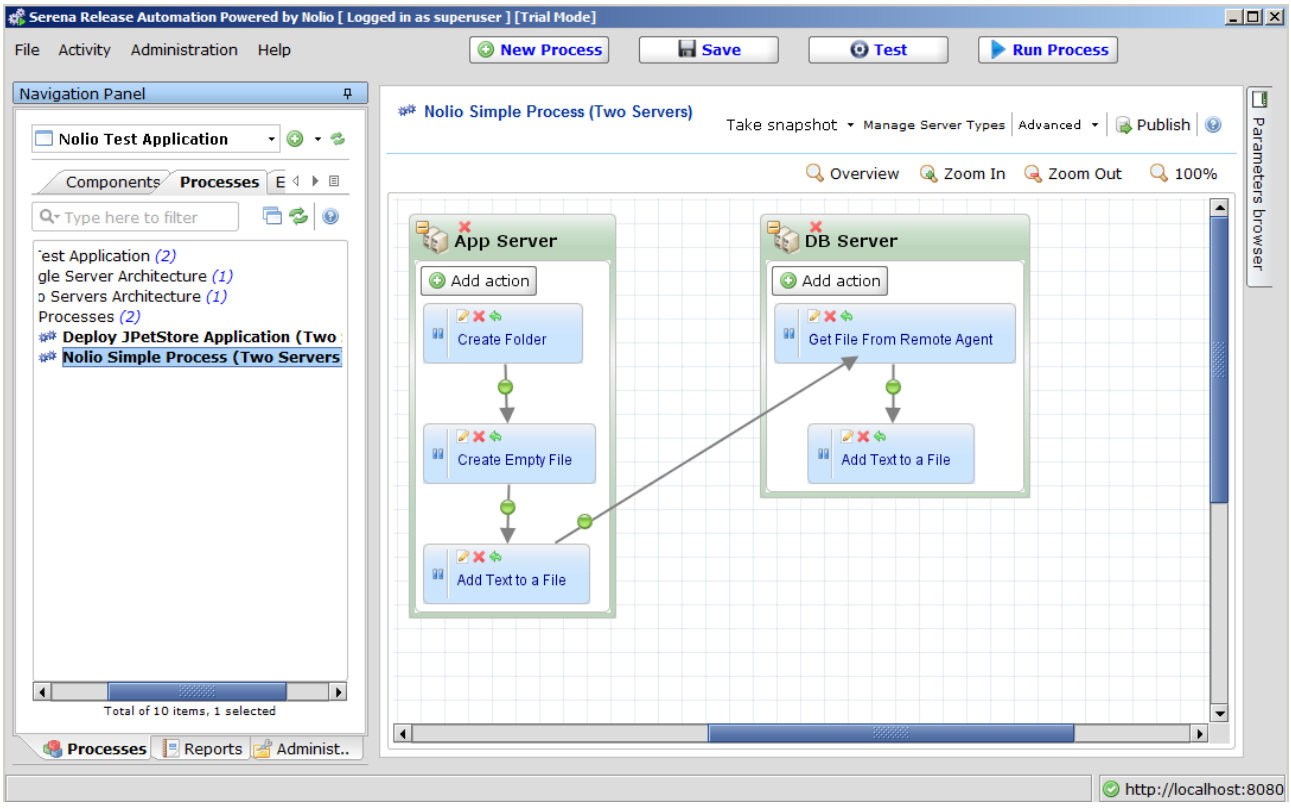
The **public** is used by the general audience of the Serena Release Automation.

Publishing a Process

After you have completed the creation of a process and tested it, you can now **publish** the process and assign it to a specific **Environment**.

To publish a process:

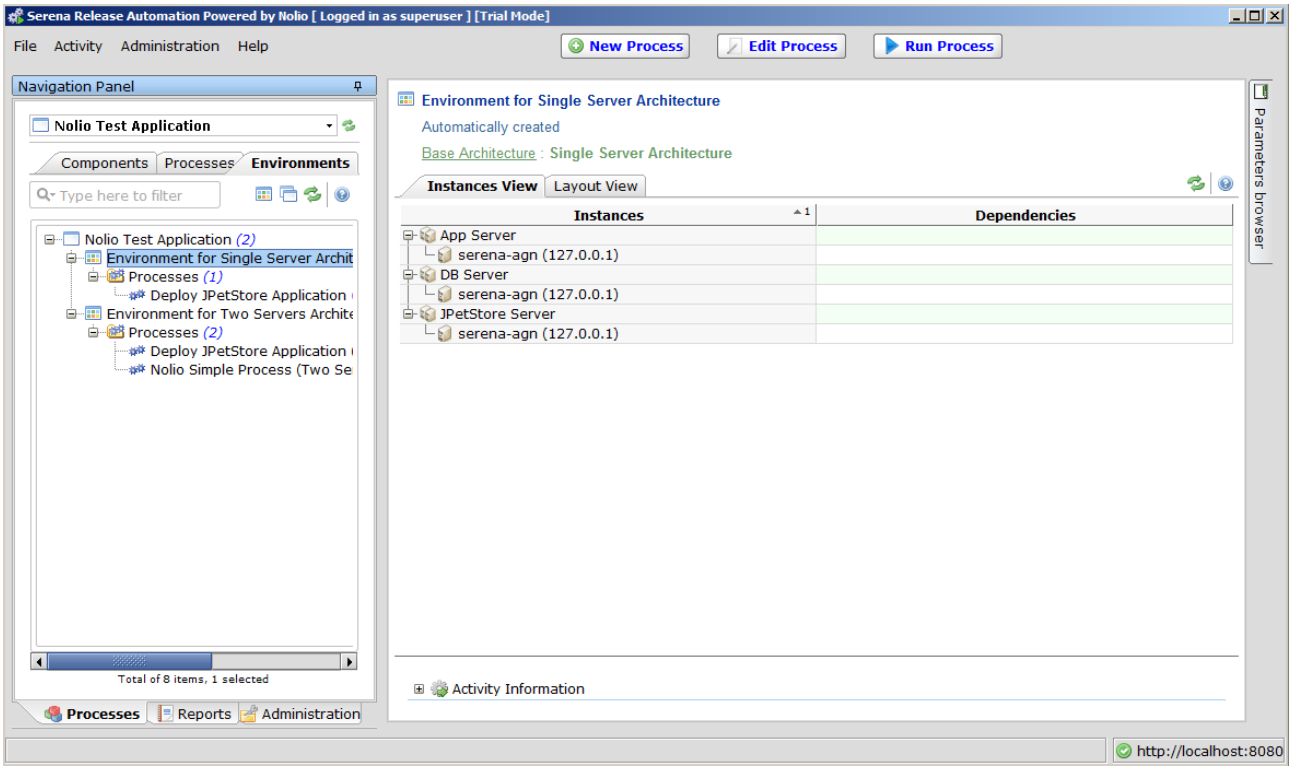
1 In the **Processes** tab of the **Navigation Panel**, click the **Processes** tab. The **Processes** page opens.



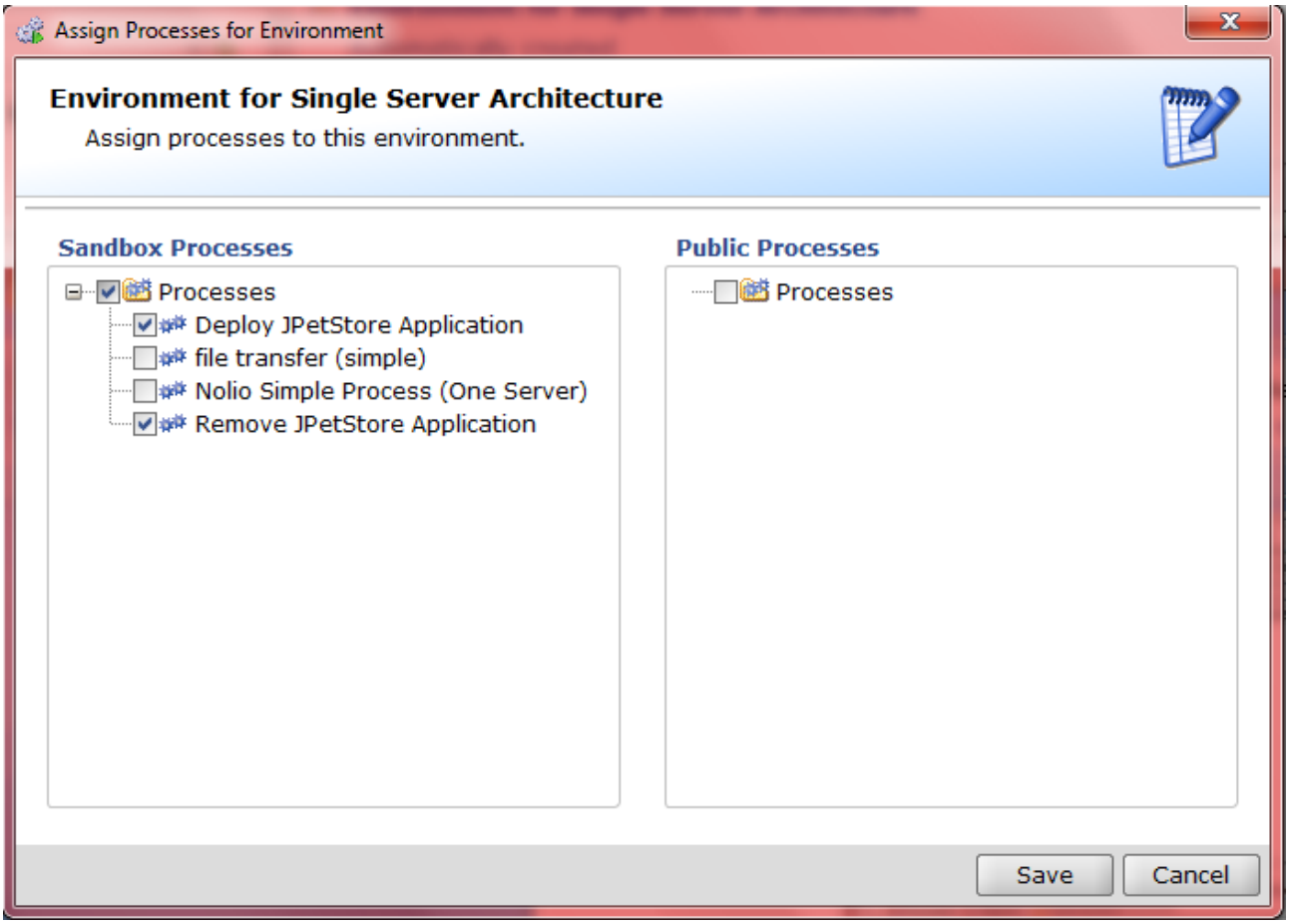
2 Expand the **Architecture** that holds the process you wish to publish and select the requested process.

3 Click Publish and approve the operation. The process is now published.

4 In the **Processes** tab of the **Navigation Panel**, click the **Environment** tab. The **Environment** page opens.



5 Right-click the **Environment** you wish to assign the published process to, and click **Assign Processes**. The **Assign Process to Environment** window opens.



6 You can now mark the process you've just published and promote it to **sandbox** or **public.**

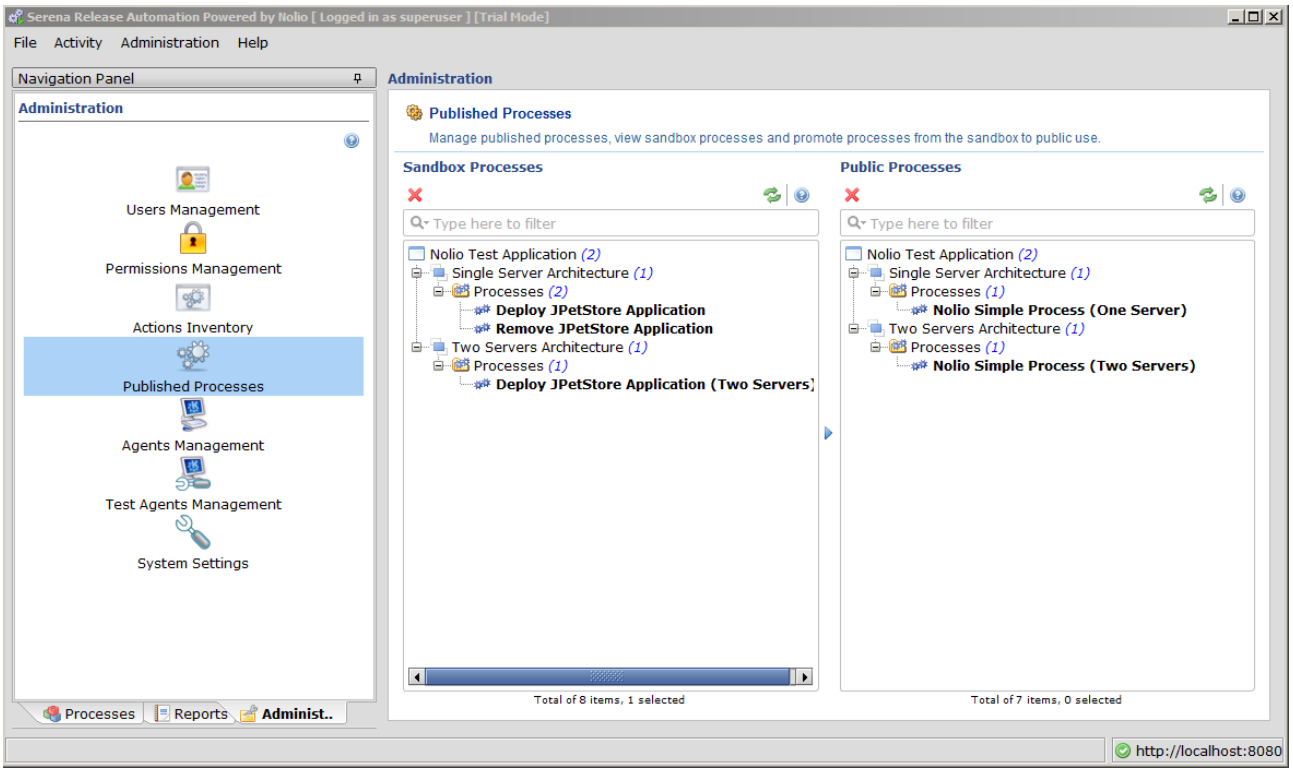
7 Click **Save.**

Viewing Published Processes

You can view processes that are still in the sandbox, and those promoted for public use.

To view a published process:

1 In the **Administration** tab of the **Navigation Panel**, click **Published Processes**. The **Published Processes** page opens.



2 In either the **Sandbox** Processes or **Public** Processes pane, select the process you want to view.

3 At the bottom of the page, click ⊞ to expand the **Preview** pane to display the selected process.

4 To resize the process diagram, use the **PageUp** and **PageDown** buttons.

Promoting Published Processes

The suggested working process states that a process should **not** be implemented by general audience until it is promoted from **sandbox** to **public**.

To promote a published process from Sandbox to Public:

1 In the **Administration** tab of the **Navigation Panel**, click **Published Processes**. The **Published Processes** page opens.

2 In the **Sandbox Processes** pane, select the process you want to promote and click ▶. The selected process is promoted to public.

Deleting Published Processes

You can **delete** processes that are still in the **sandbox**, as well as those promoted for **public** use.

To delete a published process:

1 In the **Administration** tab of the **Navigation Panel**, click **Published Processes**. The **Published Processes** page opens.

2 In either the **Sandbox Processes** or **Public Processes** pane, select the process you want to delete and click  . The process is removed from the **Published Processes** list.

Note: No confirmation message appears on delete.

Chapter 12

Managing Application Services

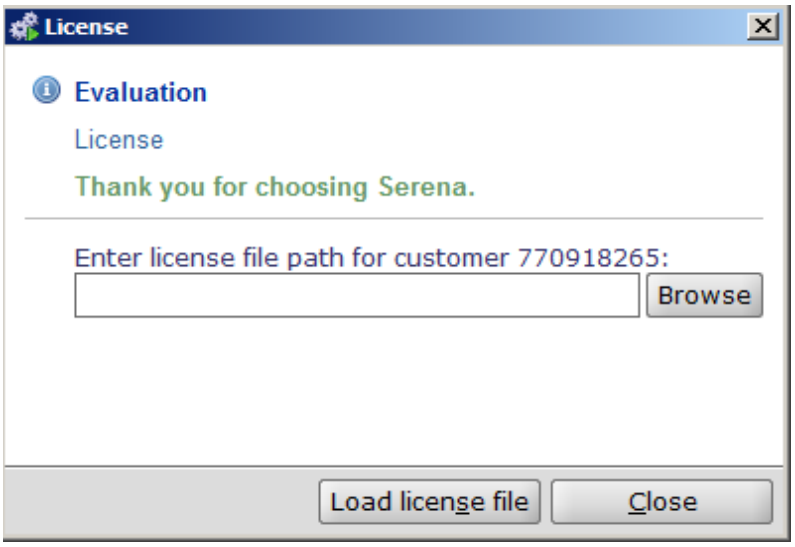
In This Chapter

This section covers application services relevant to the installation.

Updating Serena Release Automation License

To update your Serena Release Automation license from Evaluation key to Full or to extend your license time:

1 In the Serena Release Automation toolbar, click **Help > Enter License**. The **License** window opens.

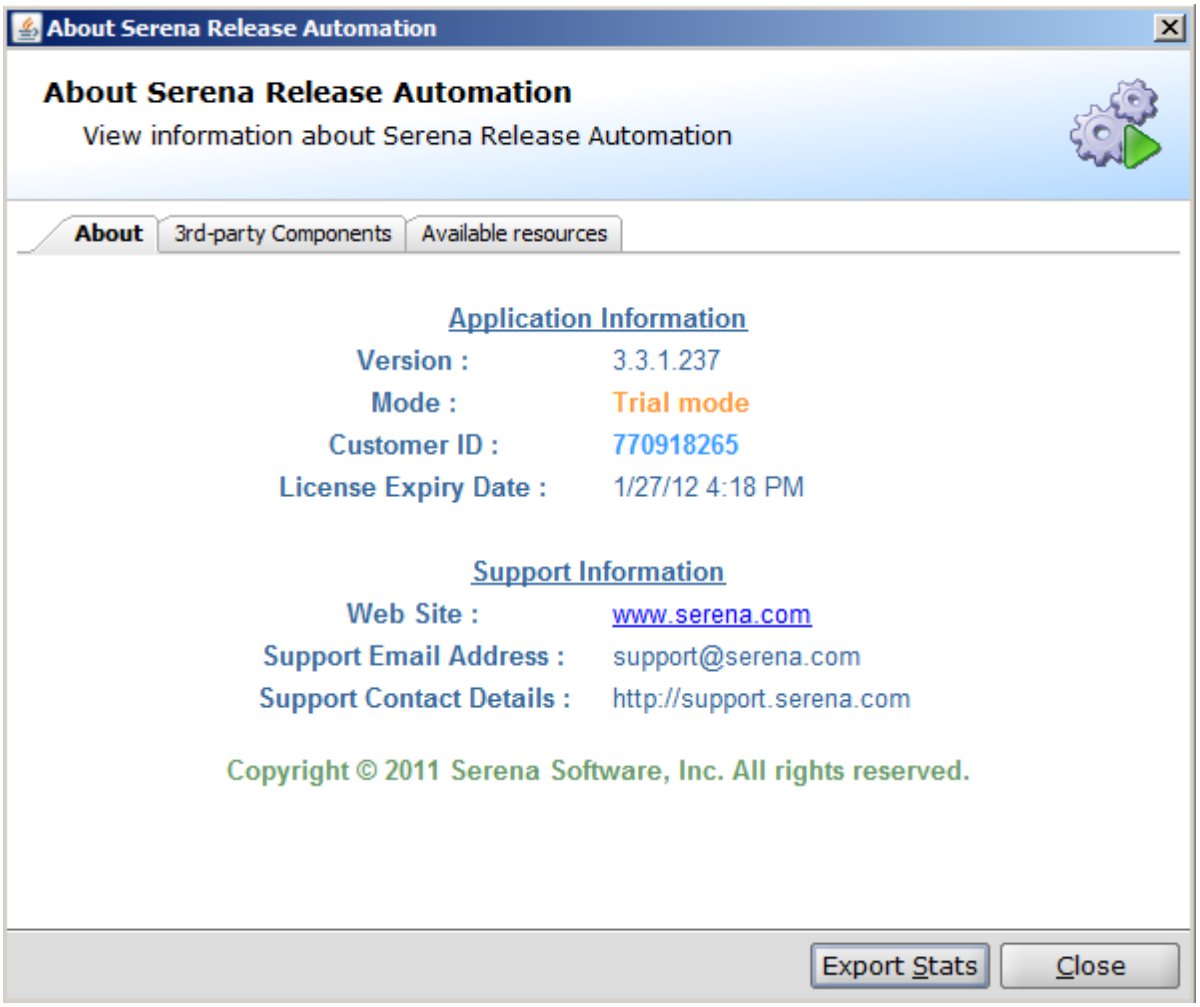


2 Browse to the location of the license file or supply a path to that file and click **Load license file**.

Displaying 3rd-Party License Agreements

To display 3rd-party licenses for libraries integrated in Serena Release Automation:

1 In the Serena Release Automation toolbar, click **Help** > **About**. The **About Serena Application Services** window opens.



2 Click the **3rd-party Components** tab. The **3rd-party Components tab** lists the 3rd-party licenses for the libraries used by Serena Release Automation.

3 To access detailed information about a specific license, click the relevant hyperlink.

Changing Default Local Font

Serena Release Automation Client UI provides the ability to support special characters and local fonts, such as European languages.

To enable local fonts:

1 From the Release Automation Client UI **File** menu, select **Use Local OS Font**.

2 Restart the Release Automation UI.

Note: For the font change to take effect, Release Automation UI must be restarted.

Enabling Design Auditing

Design Auditing captures and allows reporting on changes made to:

- Any design components
- Processes since last publication
- Changes at the administrative level
- Execution logging

The Design Auditing feature is not set to active at installation. The user needs to manually change its setting from **false** to **true**.

To enable Design Auditing:

1 From the Release Automation UI, open the **Administration** tab.

2 Select **System Settings**.

3 For Parameter name **Audit Design Changes**, double-click in the Parameter Value column.

The Edit System Settings dialog displays.

4 In the **New value** box, type **true** and click **Save**.

5 Close Release Automation UI.

6 Restart Serena Server service on the Data Management machine for the change to take effect.

Appendix A

Using MS SQL Server as Database for Serena

In This Appendix

As described above, when selecting MS SQL Server as the database for Serena, the user can select to create the required database components as part of the installation or to use a pre-defined database and existing login.

The following information is mainly relevant for users who:

1 Select Custom installation with the 'Skip Database Configuration' option

or

2 Want to install against a database that was previously created by a DBA for the sole use of Serena

Following are instructions on how to manually create the database and login.

Before the installation can proceed, and prior to the Serena Server service start-up, it is necessary to perform the following pre-installation tasks:

1 Create a MS SQL Server database for Serena use.

2 Verify TCP/IP protocol and properties are enabled on the Server.

3 Enable SQL Server Browser Service.

Note: Supported MS SQL Server versions are SQL 2005 and higher.

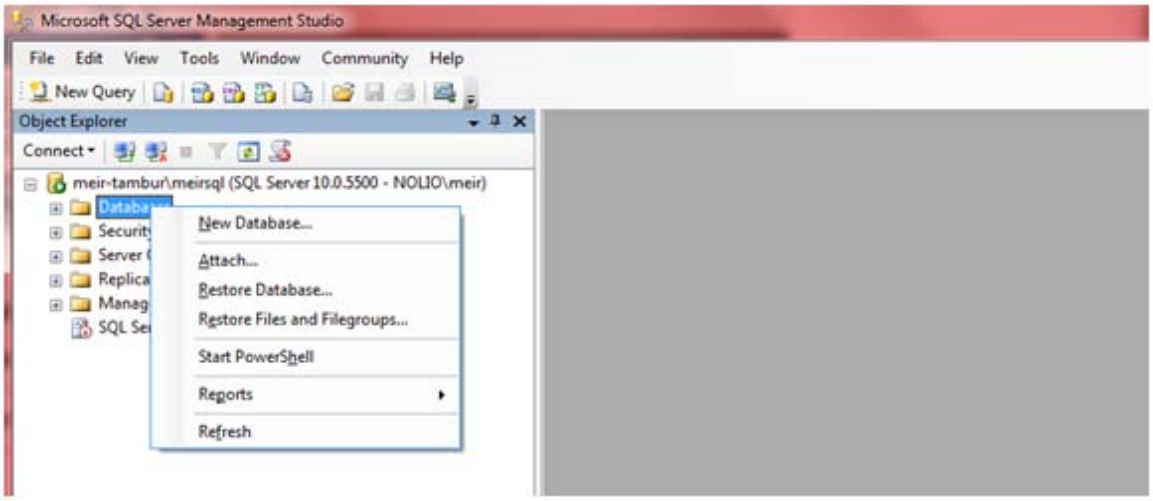
Creating MS SQL Server Database

On the MS SQL Server instance, create a new database to be used by Serena Server.

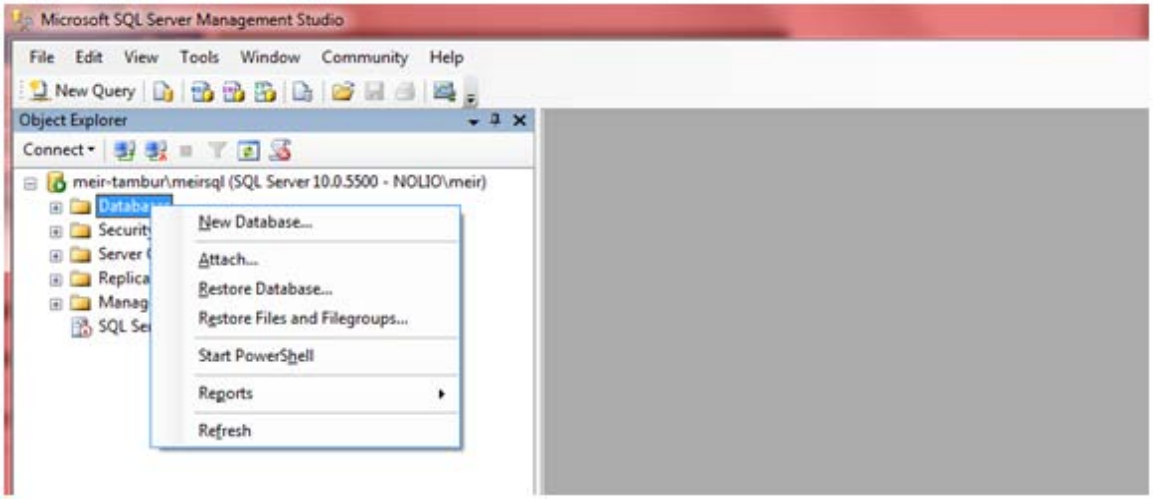
To create an MSSQL Server database:

1 Log in to SQL Server Management Studio.

2 Right-click on the Databases folder and select **New Database**.



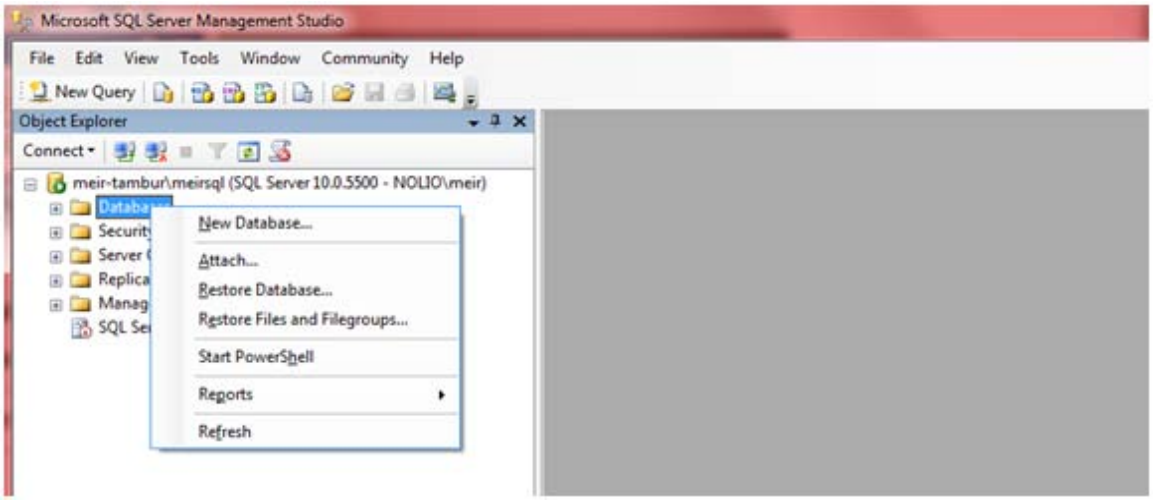
3 Enter the new database name and click **OK**.



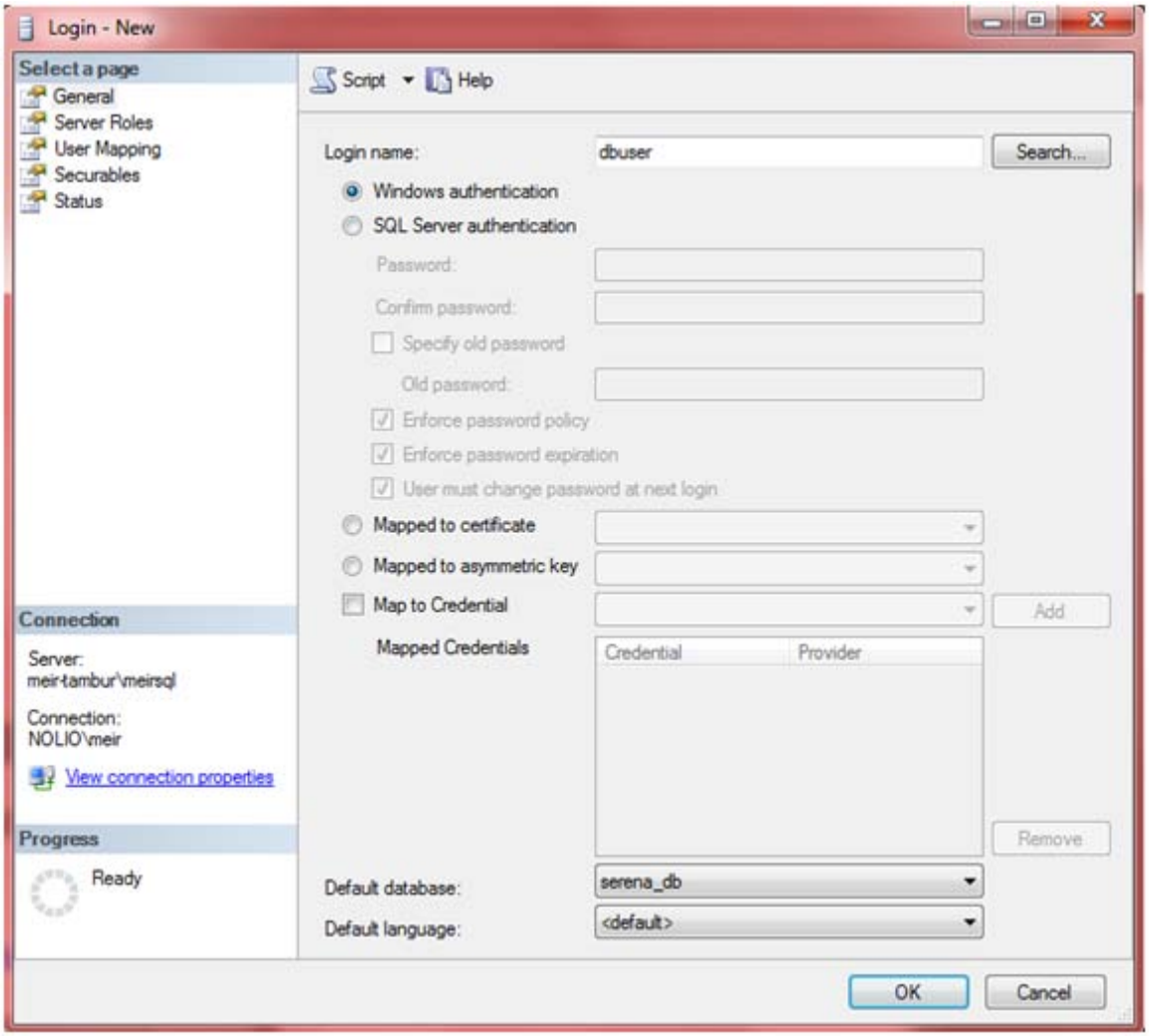
To add a new login:

1 Open the server Security folder.

2 Right-click on **Logins** and select **New Login**.



The general page for Logins- New displays.



3 In the **Login name** box, enter the name of the database user.

4 Select **Windows authentication**.

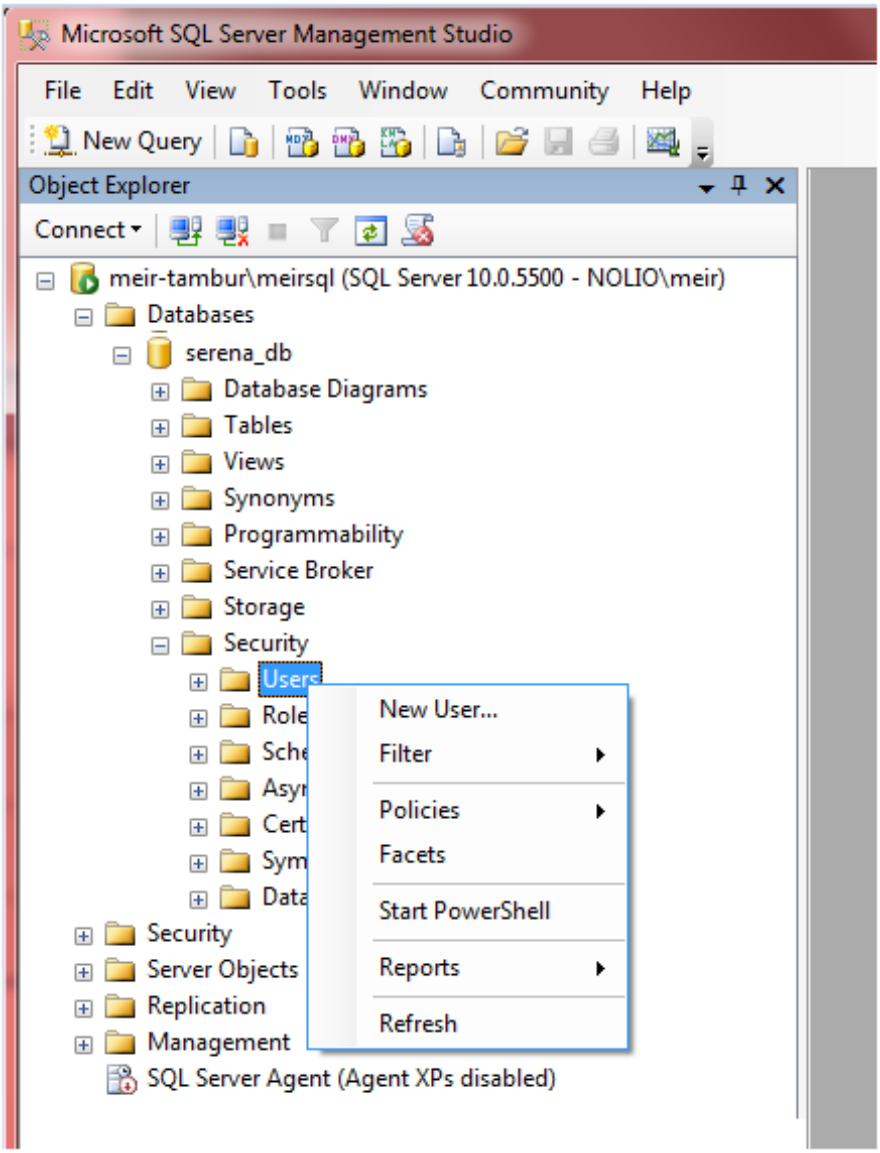
5 In the **Default database** box, enter the name of the new Serena database. The default is serena_db.

6 Click **OK** to add the user to the Serena database.

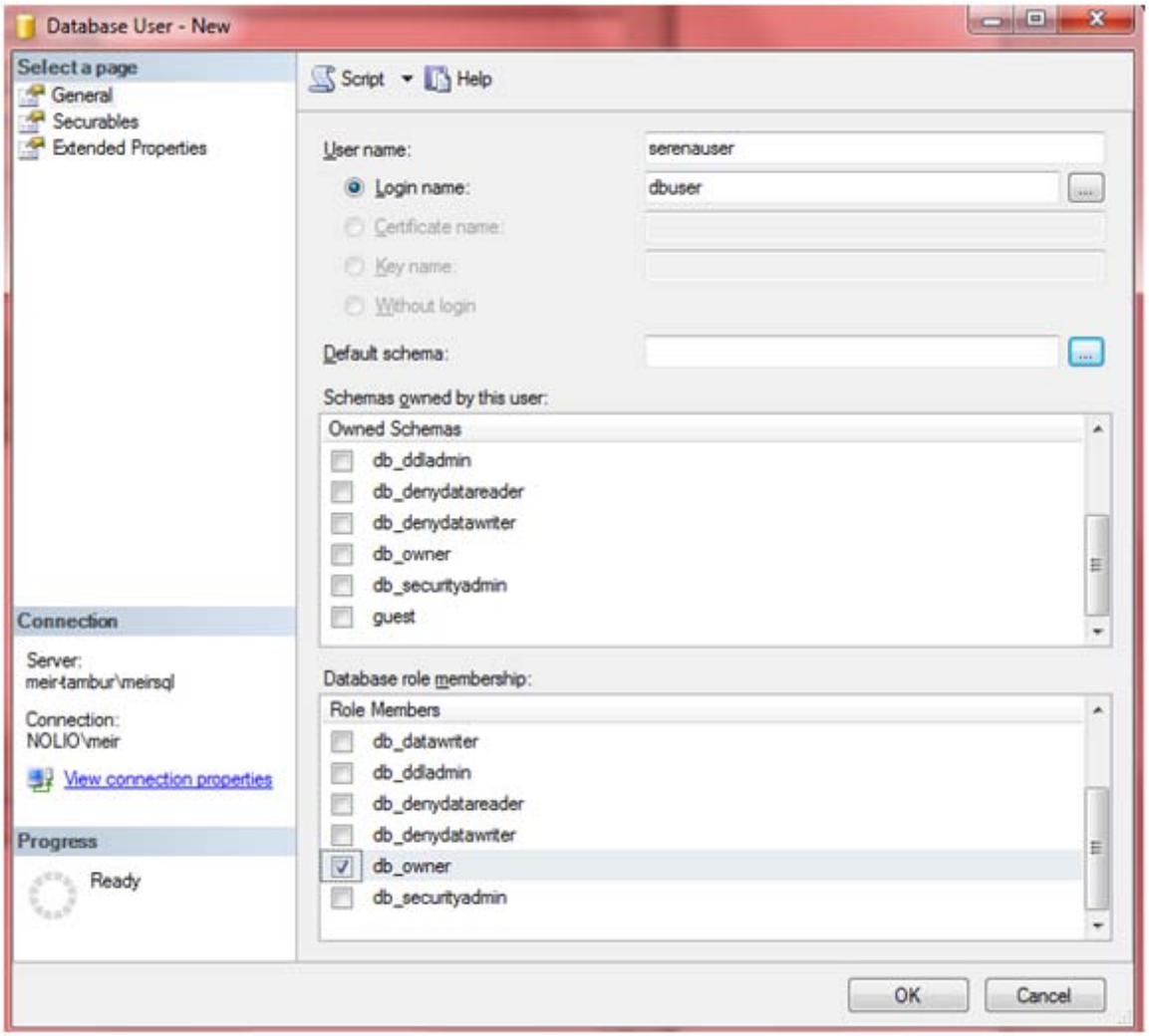
To add a new user to the database:

1 In the Databases tree, open the **serena_db** folder.

2 In the serena_db **Security** folder, right-click on **Users** and select **New User**.



The Database User – New window displays.



3 In the Database User – New window, enter the following:

a. In the User name box, enter the name of the newly created user.

b. Select **Login name**, click the browse icon, and select the login name of the new user.

c. In the Data role membership pane, select the **db_owner** check box.

d. Click **OK**.

Enabling TCP/IP Protocol and Restarting SQL Service

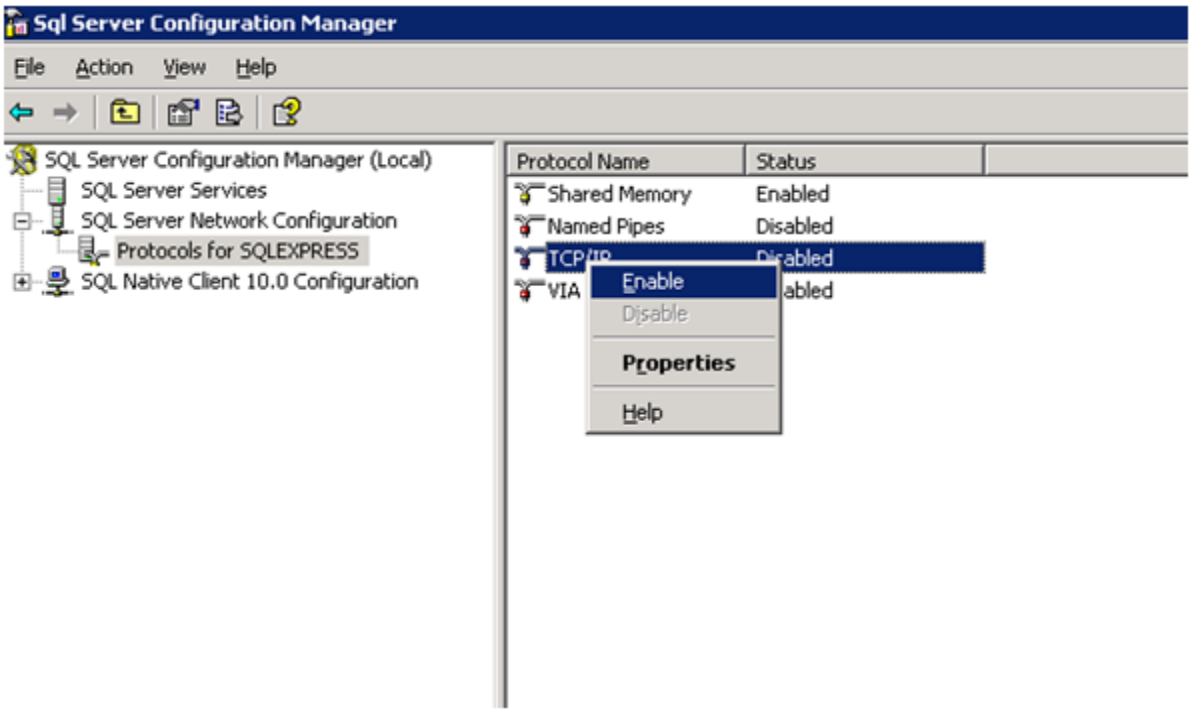
Verify that TCP/IP protocol and properties are enabled and restart the SQL Server service.

To verify TCP/IP protocol and properties enabled on MS SQL Server:

1 Login in to the Sql Server Configuration Manager.

2 In the SQL Server Configuration Manager tree, under **SQL Server Network Configuration**, select **Protocols for <INSTANCENAME>**.

In the following example, the instance name is SQLEXPRES.



3 In the right panel, verify that the status for TCP/IP protocol is enabled. If not, right-click on **TCP/IP** and select **Enable**.

4 Click the **IP Addresses** tab.

5 Verify the following for each interface:

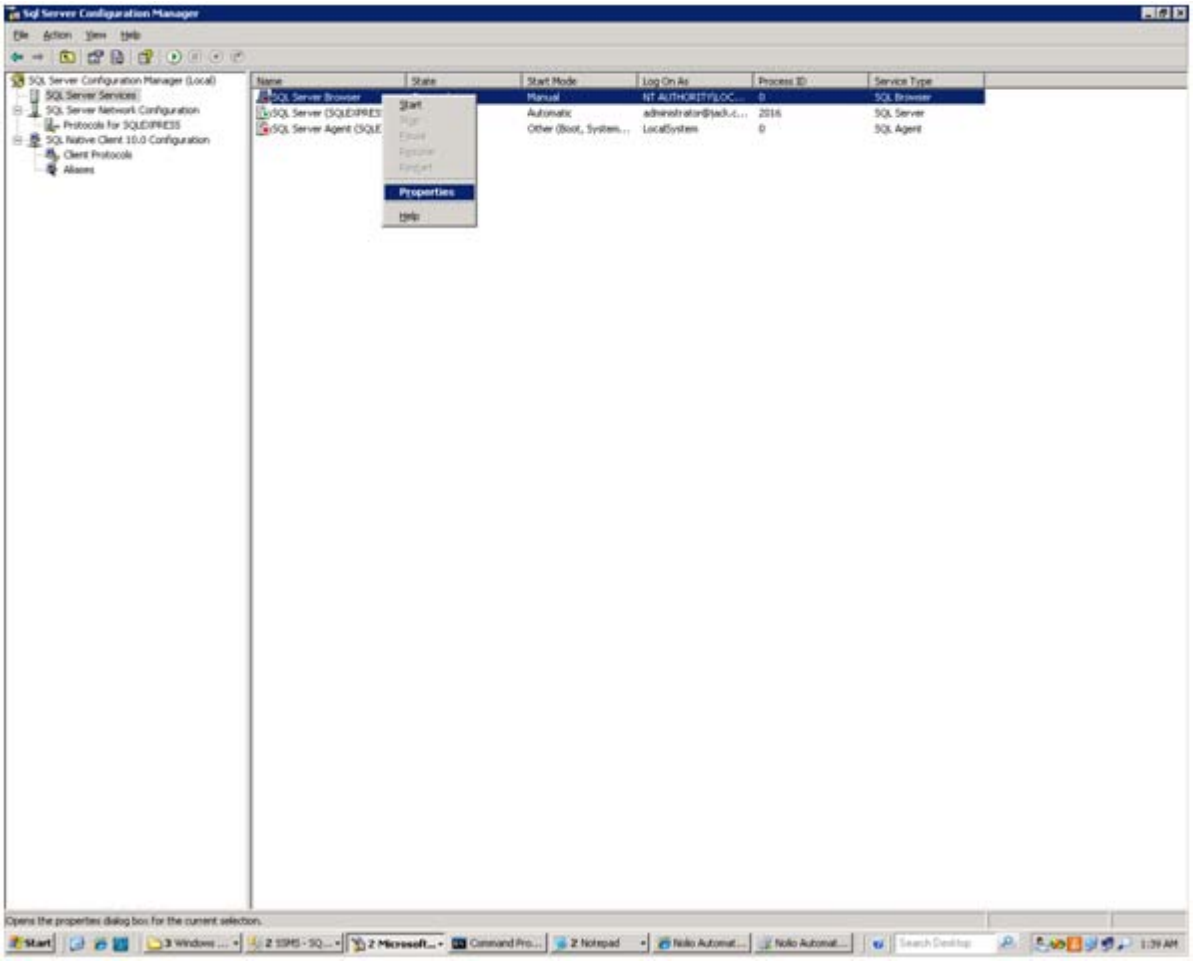
a. TCP Dynamic Ports is set to the TCP port you want MS SQL to use. The default is 1433.

b. Enable binding is enabled on relevant interfaces.

Enabling SQL Server Browser Service

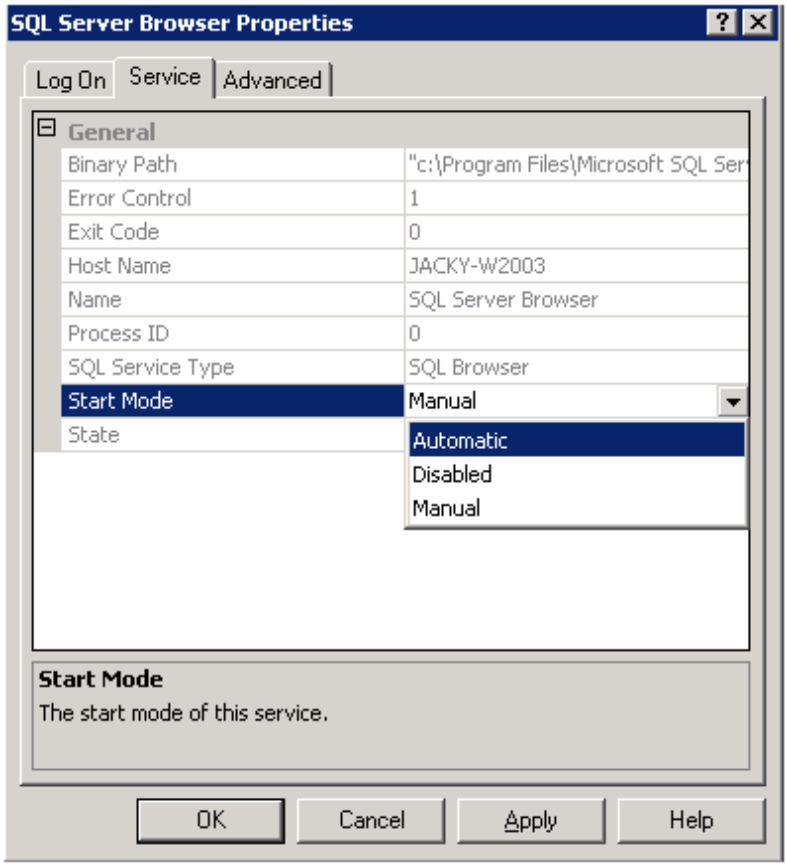
To enable the SQL Server Browser Service:

1 Log in to the SQL Server Configuration Manager and select **SQL Server Services**.



2 Right-click on **SQL Server Browser** and select **Properties**.

The SQL Server Browser Properties dialog displays.



3 Click the **Service** tab.

4 In the **Start Mode** list, select **Automatic** and click **OK**.

5 Right-click on SQL Server Browser again and select **Start**.

Appendix B

Using Oracle as Database for Serena

In This Appendix

As described above, when selecting to use Oracle as the database for Serena, the user can select to create the required schema owner and its tablespace as part of the installation or to use a pre-defined database user and tablespaces.

The following information is mainly relevant for users who:

1. Select Custom installation with the 'Skip Database Configuration' option

   or

2. Want to install against a database that was previously created by a DBA for the sole use of Serena

Creating Oracle Database

The information below includes instructions how to manually create the database user, its tablespaces and provide the required grants:

The following instructions should be performed by a database user with DBA privileges:

1. create user <USERNAME> identified by <PASSWORD>;
2. create tablespace <TABLESPACENAME> datafile '<FILENAME.DBF>.dbf' size 500m autoextend on;
3. alter user < USERNAME > default tablespace < TABLESPACENAME >;
4. grant unlimited tablespace to < USERNAME >;
5. grant connect to < USERNAME >;
6. grant create table to < USERNAME >;
7. grant create sequence to < USERNAME >;
8. grant create trigger to < USERNAME >;

Checking for Oracle Listener Name

To check for the Oracle Listener Name:

1 From the command line, or prompt, on the database machine, run the following command , replacing LISTENER NAME with the actual listener name:

```
lsnrctl services <LISTENER NAME>
```

A list of SERVICE_NAMES that Oracle Listener is expecting will display.

2 Identify the SERVICE NAME representing the Oracle SID to which you are referring, and use the name as the value for SID or Service Name as provided during installation.

Example 1:

```
C:\Users\Administrator>lsnrctl services listener
LSNRCTL for 32-bit Windows: Version 10.2.0.3.0 - Production on 14-MAR-2011
10:27:13
Copyright (c) 1991, 2006, Oracle.  All rights reserved.
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1)))
Services Summary...
Service "ora10g.serena.com" has 1 instance(s).
  Instance "ora10g", status READY, has 1 handler(s) for this service...
    Handler(s):
      "DEDICATED" established:2 refused:0 state:ready
         LOCAL SERVER
Service "ora10gXDB.serena.com" has 1 instance(s).
  Instance "ora10g", status READY, has 1 handler(s) for this service...
    Handler(s):
      "D000" established:0 refused:0 current:0 max:1002 state:ready
         DISPATCHER <machine: serena-ORA10G-SRV, pid: 1216>
         (ADDRESS=(PROTOCOL=tcp)(HOST= serena-ORA10G-SRV)(PORT=49159))
The command completed successfully
```

In Example 1, the Serena schema is to be created in the "ora10g" SID. Therefore, according to the SERVICE_NAME, the value for "data.management.database.name" should be "ora10g.serena.com".

Example 2:

```
C:\Users\Administrator>lsnrctl services listener
LSNRCTL for 32-bit Windows: Version 11.2.0.1.0 - Production on 14-MAR-2011
19:30:37
Copyright (c) 1991, 2010, Oracle.  All rights reserved.

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
Services Summary...
Service "ora11gXDB.serena.com" has 1 instance(s).
  Instance "ora11g", status READY, has 1 handler(s) for this service...
```

```
    Handler(s):
      "D000" established:0 refused:0 current:0 max:1022 state:ready
         DISPATCHER <machine: serena-ORA11G-SRV, pid: 2268>
         (ADDRESS=(PROTOCOL=tcp)(HOST=2k8-x64-0)(PORT=49171))
Service "orcl" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this service...
    Handler(s):
      "DEDICATED" established:0 refused:0 state:ready
         LOCAL SERVER
The command completed successfully
```

In Example 2, the Serena schema is to be created in the "orcl" SID. As such, according to the SERVICE_NAME, the value for the "data.management.database.name" should be "orcl".

Appendix C

Installing Server using varfile

In This Appendix

Copy the following varfile template to a new file named varfile response.varfile.

Note: After copying the template, it is necessary to edit the new file to remove all comments including angle brackets ('<text>') before adding site-specific values.

Server varfile Template

```
## MAIN Information

#Installation Path (if windows, path should include ‘\\’ replacing single ‘\’. For example: C:\\Program Files (x86)\\Serena\\Serena Release Automation)

sys.installationDir=<MyInstallationPath>

#Supernode IP Address (should include value only if server installation includes an Execution Server)

nolio.agents.supernode=<127.0.0.1>

#Execution Server Name (the name of the Execution Server as known to the Agent machines)

nolio.execution.name=<MyExecutionServerHostName>

#Execution Server Node Name (a unique node id)

nolio.nimi.node.id=<es_ MyExecutionServerHostName>

#Program Group Name

sys.programGroupName=Serena

#Nolio Serice as Local System (true if installed with LocalSystem account. otherwise, false)

install.service.lsa$Boolean=<true>

#Nolio Service Password (blank if using LocalSystem account. otherwise, service password)

nolio.service.pw=

#Nolio Service User (blank if using LocalSystem. otherwise, service owner. If the installation will use mssql windows authentication, the service user must be defined administrator in the mssql database. When using the format domain\user, the format should include ‘\\’ replacing single ‘\’. For example: mydomain\\myusername.)

nolio.service.user=

#Add shortcut of Serena to Desktop (true of false)

createDesktopLinkAction$Boolean=<true>
```

```
## DB Variables

#DB TYPE 0-MYSQL,  1-MSSQL, 2-ORACLE

nolio.db.type$Integer=<2>

#DB Hostname or IP Address

nolio.db.host.name=<DBHostName>

#DB Username (when installing with mssql windows authentication, leave blank)

nolio.db.user.name=<myusername>

#DB Password  (when installing with mssql windows authentication, leave blank)

nolio.db.password=<mypassword>

#DB Schema Name (for more details what is expected as database name see
instructions in the installation and administration guide)

nolio.db.database.name=<mydatabasename>

#DB Create Schema (true or false. If set to false, database will not be created
and installation supports only DM and ES installation)

nolio.db.create$Boolean=<true>

#DB Port

nolio.db.port=<1521>

#DB Demo create true or false (can be ignored)

nolio.db.isempty$Boolean=true

#MSSQL DBA User name (mssql instance username that can create database. blank
if not using mssql or using mssql windows authentication)

nolio.db.mssql.dba.user=

#MSSQL DBA Password (blank if not using mssql or using mssql windows authentication)

nolio.db.mssql.dba.password=

#MSSQL Windows Authentucation (true or false. blank if not using mssql)

nolio.db.mssql.winauth=

#MSSQL DBA Windows Authentucation (true or false. blank if not using mssql)

nolio.db.mssql.dba.winauth=
```

```
#ORACLE DBA Username (oracle user that can create other users and database
objects. typically ‘sys’ or ‘system’ user. blank if not using oracle)

nolio.db.oracle.dba.user=

#ORACLE DBA Password (blank if not using oracle)

nolio.db.oracle.dba.password=

#ORACLE Tablespace Name (blank if not using oracle)

nolio.db.oracle.tablespace=

#ORACLE DATAFILE Name (blank if not using oracle)

nolio.db.oracle.tablespace.file=


## Ports Variables

#Agent NiMi PORT (default is 6600)

nolio.nimi.port=<6600>

#Execution Server NiMi Port (default is 6900)

nolio.execution.port=<6900>

#TOMCAT HTTP Secured Port (default is 8443)

tomcat.port.ssl=<8443>

#TOMCAT AJP PORT (default is 8009)

tomcat.port.ajp=<8009>

#TOMCAT HTTP Port (default 8080)

tomcat.port.http=<8080>

#TOMCAT Shutdown Port (default is 8005)

tomcat.port.shutdown=<8005>

#JMX FLAG (false if using default setting. True if the JMX port is to be
changed)

nolio.hiddenport$Boolean=false

#JMX Port (provide value only if nolio.hiddenport$Boolean is set to true.
default is 20203)

port.hidden=20203
```

```
# GENERAL parameters
#Installation Type 0 - Complete, 1 - Custom
nolio.install.type$Integer=0
#Mark to install Data Management (true or false)
nolio.install.dm$Boolean=<true>
#Mark to install Execution Server (true or false)
nolio.install.es$Boolean=<true>
#Mark to install Agent (true or false)
nolio.install.agent$Boolean=<true>
#Mark to Demo Processes (not at use)
nolio.install.flows$Boolean=false
#BRANDING TYPE (nolio or serena. If left blank will use nolio)
nolio.branding.name=<serena>
#Just for Windows - STARTUP Menu
sys.programGroupDisabled$Boolean=true
#Additional Execution Flag
sys.component.12751$Boolean=true
#Server Infrastructure Flag
sys.component.336$Boolean=true
#Additional DM Flag
sys.component.12750$Boolean=true

# Other parameters
#Product Activation Key
nolio.product.key=youractivationkey
# Language - not in use
sys.languageId=en
```

```
#Nimi Supernode

nolio.nimi.supernode=default

#Nimi Secured flag (true of false. will be update for both Execution Server
and Agent. must be aligned between these components)

nolio.nimi.secured$Boolean=<true>

#Agent Node ID (unique agents node id)

nolio.agent.jxta.node.name=<MyAgentHostName>
```

Appendix D

Installing Server without using varfile

In This Appendix

Server Response varfile Template

Following is an example of an installation command line with input parameters marked with '-V' prefix:

```
./nolio_server_Linux_4_1_0_b13.sh –q
-Vsys.programGroupDisabled$Boolean=true Vsys.programGroupName=SERENA
-Vsys.programGroupAllUsers$Boolean=true -Vsys.languageId=en
-Vsys.installationDir=/opt/Serena/SerenaReleaseAutomation
-Vsys.component.12750$Boolean=true -Vsys.component.336$Boolean=true
-Vsys.component.28932$Boolean=true -Vsys.component.12751$Boolean=true

-Vnolio.db.isempty=true -Vnolio.db.database.name=serena_db
-Vnolio.db.create$Boolean=true -Vnolio.db.password=mypassword
-Vnolio.db.port=3306 -Vnolio.db.user.name=root
-Vnolio.db.host.name=serena-server -Vnolio.hiddenport$Boolean=false
-Vnolio.comm.conf.type$Integer=0 -Vnolio.os.arch=x86
-Vnolio.branding$Boolean=true -Vnolio.product.key=0123456789
-Vnolio.install.dm$Boolean=true -Vnolio.install.es$Boolean=true
-Vnolio.install.agent$Boolean=true -Vnolio.install.flows$Boolean=false
-Vnolio.install.type$Integer=0 -Vnolio.skip.install.mysql$Boolean=false
-Vnolio.nimi.port=6900 -Vnolio.nimi.secured$Boolean=false
-Vnolio.nimi.supernode=default -Vnolio.nimi.node.id=es_servername
-Vnolio.server.address=192.168.10.10 -Vnolio.server.name=serena-srvname
-Vnolio.agents.supernode=192.168.10.10 -Vnolio.agent.jxta.node.name=
-Vnolio.execution.port=6600 -Vnolio.execution.name=192.168.10.10
```

Appendix E

Installing Agent using varfile

In This Appendix

Copy the agent varfile template to new file named `deployer.silent.varfile`.

Note: There is no need to update anything in the Agent file.

Agent varfile Template

```
#Installation Path (if windows, path should include '\\' replacing single '\'. For example: C:\\Program Files (x86)\\Serena\\Serena Release Automation)

sys.installationDir=<MyInstallationPath>

#application name to which the agent should be mapped to. empty if mapping is done from the UI

nolio.agent.mapping.application=

#environment  name to which the agent should be mapped to. empty if mapping is done from the UI

nolio.agent.mapping.environment=

#server type  name to which the agent should be mapped to. empty if mapping is done from the UI

nolio.agent.mapping.servertype=

#unique node id of the agent. typically, hostname or ip address of the machine

nolio.nimi.node.id=<MyAgentHostName>

#agent port number (default 6600)

nolio.nimi.port=<6600>

#owner of the agents service true if installed with LocalSystem account. otherwise, false)

install.service.lsa$Boolean=true

#agent service user (blank if using LocalSystem. otherwise, service owner)

nolio.service.user=

#agent service password (blank if using LocalSystem account. otherwise, service password)

nolio.service.pw=

#nimi secured flag (true of false. must be aligned between the agent and its execution server)

nolio.nimi.secured$Boolean=true

#execution server name or ip address

nolio.execution.name=127.0.0.1
```

```
#execution server port
nolio.execution.port=6900


#Not in use and can be ignored
nolio.nimi.supernode=127.0.0.1\:6900
nolio.hiddenport$Boolean=false
sys.programGroupDisabled$Boolean=false
sys.component.336$Boolean=true
sys.programGroupName=Serena
sys.programGroupAllUsers$Boolean=true
sys.languageId=en
```

Appendix F

Dynamic Agent Mapping

In This Appendix

To automatically map agents to their appropriate application and environment during a silent installation:

1. Create one additional input file, per agent, according to the template in Agent Mapping File Template (on page 139).
2. Place the appropriate agent mapping file in the `SerenaReleaseAutomation/conf` folder of the target agent machine.
3. When the agent connects to its Execution Server, the application and environments will be mapped to the agent according to the agent mapping file.

Agent Mapping File Template

Following is a template for contents of the `agent.mapping.xml` file:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<agent-mapping>
    <application name="My Test Application">
        <environment name="Environment for Test">
            <server-type name="Remote"/>
        </environment>
    </application>
</agent-mapping>
```

Appendix G

Enabling LDAP Integration

In This Appendix

To enable LDAP integration, administrators need to manually update the `distributed.properties` file, located on the Serena Server machine under webapps/datamanagement/WEB-INF folder.

The file includes a commented-out section for the required fields for enabling LDAP integration.

Procedure

To enable LDAP integration:

1. Close all Serena client UIs
2. Stop the Serena Server service.
3. In webapps/datamanagement/WEB-INF , update the `distributed.properties` file.
   a. Uncomment the required lines.

      See Required Lines in Properties File (on page 141).

   b. Provide the required inputs.
4. Start the Serena Server service.

Required Lines in Properties File

```
#Uncomment and edit following lines to be able to log in with your Active Directory domain user.

#NOTE: User will see nothing in ASAP, unless he is a member of some security group in

#the domain, which was previously imported to ASAP, and granted with permissions

#to some application
```

```
#NOTE: only one type of LDAP integration, General or Active Directory, can be
enabled at the same time.


#use.active.directory.authentication=true

#use.active.directory.domain=<domain name, e.g: mycompany.com>

#use.active.directory.url=<ldap url, e.g: ldap://server.domain.com>


#Uncomment and edit following lines to be able to log in with your a user defined
in your local LDAP.

#NOTE: User will see nothing in ASAP, unless he is a member of some security
group in

#the domain, which was previously imported to ASAP, and granted with permissions

#to some application

#NOTE: only one type of LDAP integration, General or Active Directory, can be
enabled at the same time.

#use.general.ldap.authentication=true

#use.general.ldap.url=<ldap url, e.g: ldap://localhost:10389>

#use.general.ldap.user.fqdn=<fully qualified DN of domain user that has
permissions to see other users, e.g:uid=admin,ou=system>

#use.general.ldap.user.password=<password of the user defined in
use.general.ldap.user.fqdn>
```

Glossary

Action

An **action** is a predefined operation that may be made available to and customized for application components. In the context of the *flows* created in Serena Release Automation, each such action becomes a *step* in the flow.

Application

In the context of Serena Release Automation, an **application** is a high-level design construct whose design and ultimate execution allow achieving a fully-automated implementation of a server-based system. In Serena Release Automation, applications are developed in the **Processes** screen's **Components** tab.

Component

A **component** is each one of the system-level software modules that is involved in the implementation of a server-based application designed in Serena Release Automation.

Execution Server

The Execution Server is the server that manages a specific Data Center, handles data propagation for a server group, and includes Serena Agent modules controlling the specific servers involved in the execution of a process. An Execution Server may communicate with other Execution Servers for the purpose of Agent to Agent communication.

Flow

A **flow** is the grouping together of *actions,* made available to an application's component. This grouping together of actions specifies the chronological sequence in which two or more actions will occur. In addition, it may specify the conditional nature of the transition made from the completion of one action to the start of the next action in the sequence.

Implementation

The fully automated performance of all operations related to the deployment, maintenance, or other activities, on the software components participating in a server-based application. An implementation is achieved by executing some or all of the processes created for an architecture.

LDAP

The Lightweight Directory Access Protocol (LDAP) is an application protocol for reading and editing directories over an IP network. LDAP directories often contain user authentication information.

Process

A **process** is a high-level sequence of planned activity involving one, multiple, or all components of an application. A process is assembled by instantiating the lower-level *flows* that have previously been created per component.

Serena Agent

Serena Agent is the Serena Release Automation module that receives the appropriate instructions passed to it by Serena Release Automation and implements them on the server to which it is linked.

Serena Center

Serena Center is the central control mechanism that coordinates the implementation of packaged processes designed in Serena Release Automation.

When a process is actually executed, the appropriate data and instructions are channeled from the **Serena Center** via one or more dedicated **Gateways** to the **Serena Execution Server** that manages a specific **Data Center**. Handling data propagation for a specific group of servers, each **Data Center** also includes any number of **Serena Agent** modules, each of which controls a specific physical server involved in the execution of a process.

Index

S

T

U

V

W