SERENA®

RELEASE MANAGER 1.1

Installation and Configuration Guide

Serena Proprietary and Confidential Information

The content of this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Serena. Serena assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

Trademarks

Serena, StarTool, PVCS, Comparex, Dimensions, Mashup Composer, Prototype Composer, and ChangeMan are registered trademarks of Serena Software, Inc. The Serena logo and Meritage are trademarks of Serena Software, Inc. All other products or company names are used for identification purposes only, and may be trademarks of their respective owners.

U.S. Government Rights

Any Software product acquired by Licensee under this Agreement for or on behalf of the U.S. Government, its agencies and instrumentalities is "commercial software" as defined by the FAR. Use, duplication, and disclosure by the U.S. Government is subject to the restrictions set forth in the license under which the Software was acquired. The manufacturer is Serena Software, Inc., 1900 Seaport Boulevard, 2nd Floor, Redwood City, California 94063-5587.

Publication date: November 28, 2011

Table of Contents

Welcome to Serena Release Manager

Thank you for choosing Serena Release Manager, the orchestrated solution for application release management that enables you to plan, manage, and automate the deployment of applications into test, pre-production, and production environments with one-click deploy, start-to-finish traceability, and end-to-end visibility.

Audience and Scope

This manual is intended for personnel who are responsible for installing and configuring Serena Release Manager.

Each product in the Serena Release Manager integrated suite has full documentation. This book is not meant to replace that documentation, but will serve as a master document to guide you through the process of installing and configuring the entire suite.

Before You Begin

See the Readme for the latest updates and corrections for this manual.

Guide to Serena Release Manager Documentation

The Serena Release Manager documentation set includes the following manuals and tutorials.

Manual or Tutorial	**Description**
Serena Release Control User's Guide	Describes how to use Serena Release Control to manage application releases. (PDF manual)
Serena Release Manager Installation and Configuration Guide	Describes how to install and configure the Serena Release Manager suite of products. (PDF manual)
Getting Started with Serena Release Manager	A web-based tutorial that shows you how to get started with Serena Release Manager.
Serena Release Manager Web Services Reference	Provides a reference for the Serena Release Manager Web service syntax. (HTML)

The following installation and configuration manuals are referenced in this document.

Manual or Tutorial	**Description**
Serena Business Manager Installation and Configuration Guide	Provides information on installing SBM and creating a database. Database and Web server configuration information is also provided. (PDF manual)
Serena Business Manager Composer Guide	Provides details on using SBM Composer to create the tables, fields, workflows, orchestrations, and other design elements comprised in process apps. Information about saving, versioning, importing, and exporting process apps is also provided. This document is intended for individuals who want to design and maintain process apps. (PDF manual)

Manual or Tutorial	Description
Serena Business Manager Application Administrator Guide	Provides information on using Application Administrator to deploy process apps to runtime environments and to promote configured applications from one environment to another. (PDF manual)
Serena Business Manager System Administrator Guide	Provides information on configuring and administering the SBM Application Engine. Instructions for managing projects, user accounts, system settings, and authentication are included. (PDF manual)
Serena Business Manager Licensing Guide	Explains how to manage licenses for Serena Business Manager. License types are discussed, along with instructions for installing and using the Serena License Manager. This guide is intended for administrators who will install and implement Serena Business Manager. (PDF manual)
Serena Business Manager User's Guide	Provides information about the SBM User Workspace and is intended for end users. Instructions on using the SBM User Workspace, including the robust reporting feature in SBM, are included. To ease the process of providing a copy for every user in your system, the Serena Business Manager User's Guide is provided in PDF and can be accessed from the Product Information tab of the About page in the SBM User Workspace. (PDF manual)
Nolio Automation Center Installation and Administration Guide	Describes how install and configure Nolio. (PDF manual)
Serena Dimensions CM Installation Guides for Windows and *for UNIX*	Describes how install and configure Dimensions CM for the respective platforms. (PDF manuals)
Serena Dimensions CM Administrator’s Guide	Describes how to administer the Dimensions CM tool. (PDF manual)
Serena Dimensions CM and RM Installing the Serena License Manager	Describes installing the Windows version of SLM. (PDF manual)
Serena Dimensions CM User's Guide	A user guide to the Dimensions CM Client Tools. (PDF manual)
Serena Dimensions CM Process Modeling Guide	Describes how to configure the process model using the Administration Console. (PDF manual)

Accessing the Documentation

You can access all documentation and tutorials for the current release of the Serena Release Manager suite through the Serena Release Control Help system.

When you click the **Help** link in Serena Release Control, the User’s Guide in HTML help format appears. You can search, use the index, and view the glossary for information on using the product. Click **Demos** on the Help toolbar to access online tutorials on using Serena Release Control. Click **All Documentation** on the Help toolbar to access the full set of documentation and tutorials for the Serena Release Manager suite.

To download all documentation and tutorials for any release of the Serena Release Manager suite, see the Serena Customer Support website My Downloads page.

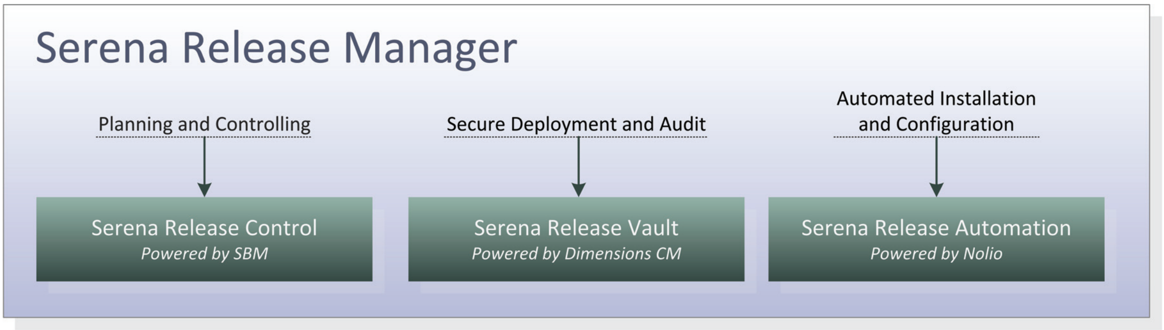
Chapter 1

Installation Prerequisites and Planning

This chapter tells you what you need to do before installing Serena Release Manager. It provides you with information to help guide you through the Serena Release Manager installation.

What is Serena Release Manager?

Serena Release Manager is an integrated suite that helps you with your release management by enabling you to control, secure, and automate your processes.

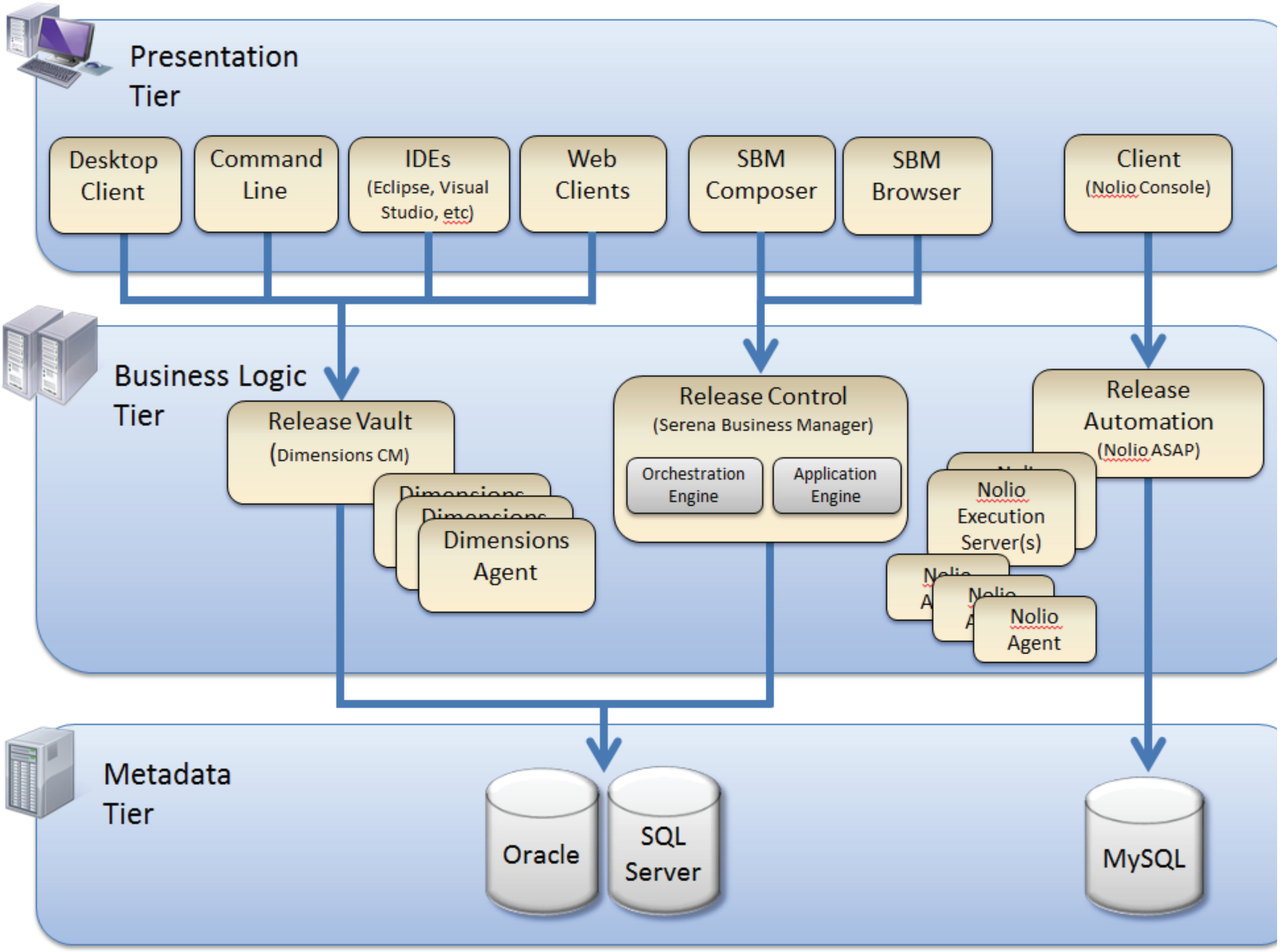


- **Serena Release Control**, powered by Serena Business Manager, helps you plan and control your application release processes across your enterprise, from definition to deployment.
- **Serena Release Vault**, powered by Dimensions CM, ensures a secure and auditable path to production.
  - Secures source for multiple releases
  - Prevents unauthorized changes
  - Enables full traceability and audit trail
- **Serena Release Automation**, powered by Nolio, automates application installation and configuration tasks.
  - Handles a high volume of tasks
  - Cuts deployment time and cost
  - Reduces deployment errors
  - Handles the complexity of multiple release destinations and configurations

For a full description of the use of Serena Release Control, see the Serena Release Control User's Guide.

Serena Release Manager Architecture

The architecture of the Serena Release Manager suite integrates Serena Release Control, Serena Release Vault, and Release Automation as shown in the following figure.



The integrated power of Release Control, Release Vault, and Release Automation are brought together through the Serena Release Control user interface and its underlying SBM infrastructure.

Preparing for the Installation

For a smooth and successful installation and configuration of the Serena Release Manager suite, prepare by gathering the information specified in the following sections.

- "Software Compatibility Requirements" on page 12
- "Installation Prerequisites" on page 12
- "Database Requirements" on page 12
- "System Requirements" on page 13
- "Installation Planning Checklists and Worksheets" on page 13

Software Compatibility Requirements

For details of supported versions of the products in the Serena Release Manager suite, supported platforms, and third party integrations, see the Serena Release Plan for Serena Release Manager at:

http://roadmap.serena.com

Select **Release Manager** and then select your version.

Installation Prerequisites

Before installing any of the products in the Serena Release Manager suite, please see their respective installation and configuration guides.

Preparing for Serena Product Installation

Before you install the Serena Release Manager suite of products, make sure you have completed the prerequisites as follows:

- **Serena Business Manager**

  Before you install Serena Business Manager, please see the "Pre-installation Checklist" section of the *Serena Business Manager Installation and Configuration Guide*.

- **Serena Release Control**

  You must complete the installation for Serena Business Manager before beginning the installation for Serena Release Control.

- **Dimensions CM**

  If upgrading from an existing version of Dimensions CM, this release supports an automatic upgrade to Serena Dimensions CM 12.1.1 from Serena Dimensions CM 10.1.3, 2009 R1.01, and 2009 R2. If you are upgrading from an older version please contact Serena Support.

- **Serena License Manager**

  If you are evaluating Serena Release Manager, Serena License Manager is not required.

- **Common Supporting Files**

  The installer for Serena Release Control automatically installs additional supporting software, such as a common Tomcat Web server.

Database Requirements

Before beginning the installation, you must have database systems installed and ready as follows:

- **Serena Business Manager**

  Serena Business Manager requires one of the following databases: Oracle or SQL Server.

- **Serena Release Vault: Powered by Dimensions CM**

Dimensions CM requires one of the following databases: Serena-Supplied Runtime, Oracle, or SQL Server.

- **Serena Release Automation: Powered by Nolio**

  Nolio requires a database to be pre-installed. Supported database systems include MySQL, Microsoft SQL Server, and Oracle.

System Requirements

This section includes references to the system requirements for each of the products installed as part of the Serena Release Manager suite.

You should determine your organization's specific needs using the detailed documentation for SBM, Dimensions CM, and Nolio as follows:

Product	System Requirements Location
Serena Business Manager	"Hardware Requirements" in *Serena Business Manager Installation and Configuration Guide.*
Serena Release Control	Your SBM requirements address the requirements for Serena Release Control.
Serena Dimensions CM	■ The Serena Dimensions CM Supported Platforms at: http://support.serena.com/Roadmap/Product.aspx?sel=PVDIMENSIONS For details of supported platforms and third party integrations, select the link for your release. ■ The Dimensions CM Readme.
Nolio	"System Requirements" in *Nolio Automation Center Installation and Administration Guide.*

Server Requirements

SBM and Dimensions CM must be installed on separate servers and use separate copies of Serena Common Web services. This ensures optimal performance and is the supported configuration.

Installation Planning Checklists and Worksheets

Checklists and worksheets that you can use to collect and document required information for the installation and configuration are provided in the Serena Business Manager and Dimensions CM documentation. These include port numbers, IP addresses and hostnames of various servers, and DB names and information.

Chapter 2

Serena Release Manager Installation

This chapter leads you through an installation of all of the components of Serena Release Manager.

Installation Overview

Make sure you have completed your planning and performed the installation prerequisites documented in the preceding chapter. Use the installation checklist and worksheets to help guide you through the installation processes.

You must install each of the systems included in the Serena Release Manager suite that you plan to use and that you do not already have installed. SBM, Serena Release Control, and Serena Release Vault are required components of the default implementation of Serena Release Manager.

For optimal performance, Serena recommends that SBM and Serena Release Vault be installed on separate servers and make use of separate instances of Serena Common Web services.

You should install or verify existing installation settings as follows:

1. Ensure that Serena License Manager is installed as required. Serena License Manager manages the licenses for Serena Release Control and Serena Release Vault. See "Installing Serena License Manager" on page 16.
2. Ensure that Serena Business Manager is installed as required. Serena Business Manager is the platform on which Serena Release Control runs. See "Installing Serena Business Manager" on page 17.
3. Ensure that Serena Release Control is installed as required. Serena Release Control enables you to plan and control your releases. See "Installing Serena Release Control" on page 18.
4. Ensure that Serena Release Vault is installed as required. Serena Release Vault enables you to securely deploy and audit your releases. See "Installing Serena Release Vault" on page 19.
5. Ensure that Serena Release Automation is installed as required. Serena Release Automation enables you to automate the installation and configuration of your deployed production files. See "Installing Serena Release Automation" on page 20.

New installations should proceed in the given order given unless you have the experience and necessity to do otherwise.

After you have completed installing each of the preceding systems, you must complete the system configuration to make Serena Release Control and Serena Release Manager ready to use.

Installing Serena License Manager

You must install Serena License Manager to license and run Serena Business Manager and Dimensions CM. You apply the licenses for all of the products installed for the Serena Release Manager suite as part of the post-installation system configuration. See "Applying Licenses" on page 26.

Existing Serena License Manager Systems

If you already have an installation of Serena License Manager, you do not need to install a new system for Serena Release Manager. You do need to do the following:

- Ensure that you are running a supported version of Serena License Manager as indicated in the Serena Release Manager Readme.
- After installing the rest of the products in the suite, follow the post-installation system configuration instructions for applying licenses.

Please continue to "Installing Serena Business Manager" on page 17.

New Serena License Manager Systems

Follow the instructions on installing Serena License Manager in either the Dimensions CM or Serena Business Manager documentation.

Documentation References

Complete documentation on installing Serena License Manager is in the following documents:

- *Serena Business Manager Licensing Guide*
- *Serena Dimensions CM and RM Installing the Serena License Manager*

Installing Serena Business Manager

Serena Business Manager must be installed before you can install Serena Release Control. After you install Serena Business Manager, you must install and configure Serena Release Control, which is a Serena Business Manager application.

Existing Serena Business Manager Systems

If you already have an installation of Serena Business Manager to which you plan to add Serena Release Control, you do not need to install a new system for Serena Release Manager. You do need to do the following:

- Ensure that you are running a supported version of SBM as indicated in the Serena Release Manager Readme.
- After installing the rest of the products in the suite, follow the post-installation system configuration instructions for SBM.

Please continue to "Installing Serena Release Control" on page 18.

New Serena Business Manager Systems

Before you install Serena Business Manager, please see the “Pre-installation Checklist” section of the *Serena Business Manager Installation and Configuration Guide*.

Documentation References

Complete documentation on installing Serena Business Manager is in the *Serena Business Manager Installation and Configuration Guide*.

Installing Serena Release Control

Existing Serena Release Control Systems

If you already have an installation of Serena Release Control and are upgrading, see "Serena Release Control Upgrade" on page 65.

New Serena Release Control Systems

Before you install Serena Release Control, make sure you have the required installation of SBM. See "Installing Serena Business Manager" on page 17.

The Serena Release Control installer copies necessary files to the SBM installation directory path. The files copied include the Java war files, the SBM solution file, and user interface shell files. After the installation, you must import and configure the SBM solution to complete the Serena Release Control installation as documented in Chapter 3, "System Activation and Configuration" on page 23.

Serena Release Control runs using the Serena Common Web server, which is an Apache Tomcat Web server. The Serena Release Control installer detects whether the Serena Common Web server is already installed, and if so, will use the existing occurrence.

The Serena Release Control installer automatically installs and configures the Serena Release Control Apache Tomcat Web server to run on the default port of 9095. If this port is already in use by another application on your server, or if you already have an instance of the Serena Common Web server running on a different port on this server, please see Chapter 6, "Addressing Web Server Port Conflicts" on page 97 for port customization options.

To install Serena Release Control:

1. If you downloaded electronic media, navigate to the folder where you downloaded the installation files and run the installer executable, `Release Control.exe`.

   OR

   If you received Serena Release Manager installation files on physical media, run `index.htm` and from the Serena Release Manager installer **Installation Components** page, click **Serena Release Control: Powered by SBM**.

   The **Welcome** page appears.

2. Click **Next**.

   The **License** page appears.

3. Confirm and click **Next**.

   The **Destination Folder** page appears.

4. Click **Next**.

The **Setup Type** page appears.

5 Click **Next**.

The **Configuration Details** page appears.

After a short pause, the installer displays whether a valid version of SBM and the Serena Release Control common Apache Tomcat Web server are installed. It installs the common Tomcat Web server if needed.

6 Click **Next**.

The **Install** page appears.

7 Click **Install**.

A page appears informing you files are being copied. After a short time, a **Completed** page appears.

Optionally select **Show the Windows Installer log**.

The installation log file appears. You can navigate to this file later to view installation details, for example for troubleshooting purposes. The default location for the file is:

```
%Temp%\Install_rc_comp.log
```

8 Click **Finish**.

The installer creates the following file and folders under the Serena Release Control installation directory path. For example:

```
C:\Program Files (x86)\Serena\Solutions\Release Control\
```

`com.serena.rlm.sbm.shell.zip`

`\solution` (contains the solution file)

`\war` (contains the `rlm.war` file)

The installer does the following automatically:

- Copies the `rlm.war` file to the Tomcat Web server and restarts Tomcat.
- Puts the UI shell files in the appropriate SBM folder.

Installing Serena Release Vault

Serena Release Vault, which is powered by Serena Dimensions CM, must be installed to use the deployment capabilities of Serena Release Manager.

Existing Dimensions CM Systems

If you already have an installation of Dimensions CM that you plan to use with Serena Release Manager, you do not need to install a new system. However, you do need to do the following:

- Ensure that you are running a supported version of Dimensions CM as indicated in the Serena Release Manager Readme.

- Ensure that Single Sign On (SSO) is enabled for Dimensions CM. For configuring SSO if not already enabled, see Dimensions CM Support for SSO in the Dimensions CM Administrator's Guide.
- After installing the rest of the products in the suite, follow the post-installation system configuration instructions for Dimensions CM.

Please continue to "Installing Serena Release Automation" on page 20.

New Dimensions CM Systems

Before you install Dimensions CM, please see the “Fresh Installation Checklist” section of the *Dimensions CM Installation Guide for Windows* or the *Dimensions CM Installation Guide for Unix*.

You must enable Single Sign On (SSO) for the communication between Dimensions CM and SBM to work successfully in Serena Release Manager. You should install SBM and the SSO server and then enable SSO during the installation of Dimensions CM to point Dimensions CM to the SSO server.

TIP If you install Serena Release Control and then install Dimensions CM 12.1.1 on the same server, make sure you enter the Serena Common Web server port number you used with Serena Release Control when prompted, so that the existing Serena Common Web server will be used.

Documentation References

- Complete documentation on installing Dimensions CM is in the *Dimensions CM Installation Guide for Windows* and the *Dimensions CM Installation Guide for Unix*.
- Complete documentation on configuring SSO is in the *Dimensions CM Administrator's Guide* in "Dimensions CM Support for SSO".

Installing Serena Release Automation

Serena Release Automation, powered by Nolio, may be installed to provide the release automation capabilities of Serena Release Manager.

Existing Serena Release Automation Systems

If you already have an installation of Serena Release Automation that you plan to use with Serena Release Manager, you do not need to install a new system. However, you do need to do the following:

- Ensure that you are running a supported version of Serena Release Automation as indicated in the Serena Release Manager Readme.
- Ensure that the supported database you are using for your Serena Release Automation system is open for remote access according to the database system documentation.

  Supported database systems include MySQL, Microsoft SQL Server, and Oracle.

- After installing the rest of the products in the suite, follow the post-installation system configuration instructions for Serena Release Automation.

Please continue to Chapter 3, "System Activation and Configuration" on page 23.

New Serena Release Automation Systems

Before you install Serena Release Automation, please see the *Serena Release Automation* documentation. You must do the following:

- Ensure that you are installing a supported version of Serena Release Automation as indicated in the Serena Release Manager Readme.
- Ensure that the supported database you are using for your Serena Release Automation system is open for remote access according to the database system documentation.

  Supported database systems include MySQL, Microsoft SQL Server, and Oracle.
- After installing the rest of the products in the suite, follow the post-installation system configuration instructions for Serena Release Automation.

NOTE The Oracle database JDBC driver is included with and licensed as part of the Serena Release Automation installation.

If you want to use MySQL or SQL Server with Serena Release Automation, you must obtain and install the JDBC drivers for those separately from the Serena Release Manager installation. You should obtain licenses for those products as required by the providers.

More details are included in the post-installation system configuration instructions for Serena Release Automation.

Documentation References

Complete documentation on installing Serena Release Automation is in the *Nolio Automation Center Installation and Administration Guide*.

Chapter 3

System Activation and Configuration

This chapter tells you how to configure your system after you have completed installing all of the products in the Serena Release Manager suite. The system configuration is required, and completes the installation of Serena Release Control. System configuration activates Serena Release Control and enables the communication between the products within Serena Release Manager.

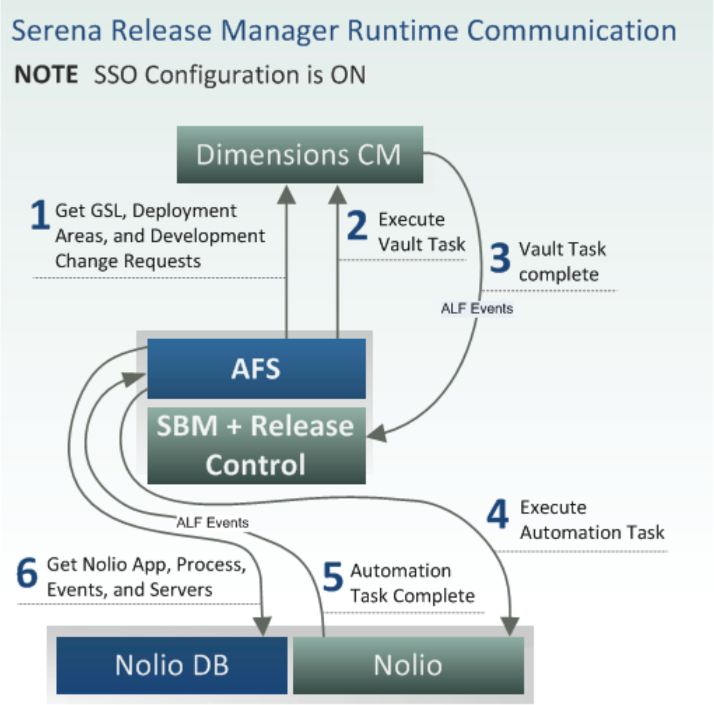
See the following sections for details.

System Configuration Overview

A high-level understanding of the flow of communication between the products in the Serena Release Manager suite will help you better understand the system information that you are required to configure as part of the installation process.

Serena Release Manager Runtime Communication

The flow of communication between products in the Serena Release Manager suite is shown in the following figure.


Serena Release Manager Runtime Communication
NOTE SSO Configuration is ON
Dimensions CM
1 Get GSL, Deployment Areas, and Development Change Requests
2 Execute Vault Task
3 Vault Task complete
ALF Events
AFS
SBM + Release Control
4 Execute Automation Task
ALF Events
6 Get Nolio App, Process, Events, and Servers
5 Automation Task Complete
Nolio DB
Nolio

All communication for Serena Release Manager goes through the Serena Release Manager Web services, which are collectively referred to as ALM Foundation Services (AFS), and SBM, with support from Application Lifecycle Framework (ALF) events for Dimensions CM and Nolio responses.

AFS and ALF are Serena ALM frameworks that provide the communication layer between integrating products.

All outgoing communication for Serena Release Manager goes from the SBM server through AFS. All incoming communication goes through AFS and Application Lifecycle Framework (ALF) to the SBM server.

Execution steps that correspond to the preceding figure are as follows:

1. ALM Foundation Services (AFS) gets the Global Stage Lifecycle, Deployment Areas, and Development Change Requests from Dimensions CM.
2. AFS executes a vault deployment task to initiate a command in Dimensions CM.
3. Dimensions CM sends a message back to SBM that the vault deployment task is complete. This is done using ALF Event Notifications.

4 AFS executes an automation deployment task to initiate a command in Nolio.

5 Nolio sends a message back to AFS that the automation deployment task is complete. This is done using ALF Event Notifications.

6 SBM gets Nolio applications, processes, events, and server lists from the Nolio database.

NOTE SSO configuration is ON to allow the communication to flow between SBM and Dimensions CM without prompting for additional signon information. This is required for the communication between SBM and Dimensions CM to work correctly.

In the system configuration, you are enabling the communication between the products within Serena Release Manager.

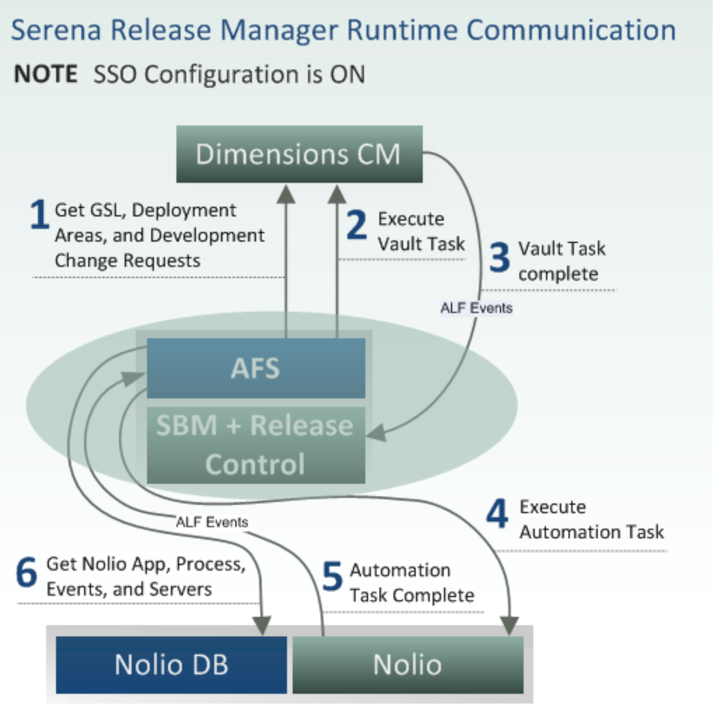
The system configuration is covered in the following sections:

- "Activating Serena Release Control" on page 25
- "Configuring Communication with Dimensions CM" on page 33
- "Configuring Communication with Serena Release Automation" on page 38

Activating Serena Release Control

Serena Release Control is a bundle of process apps that run in SBM. Before you can use Serena Release Control, you must activate it as documented in this section.

The Release Control components and communication mechanisms that you are configuring are highlighted in the following figure.


Serena Release Manager Runtime Communication
NOTE SSO Configuration is ON
Dimensions CM
1 Get GSL, Deployment Areas, and Development Change Requests
2 Execute Vault Task
3 Vault Task complete
ALF Events
AFS
SBM + Release Control
4 Execute Automation Task
ALF Events
6 Get Nolio App, Process, Events, and Servers
5 Automation Task Complete
Nolio DB
Nolio

The configuration activities included are as follows:

- "Applying Licenses" on page 26
- "Promoting and Configuring Serena Release Control in SBM" on page 26

Applying Licenses

Before you can use the products in a production environment, you must apply licensing. If you are evaluating the products, temporary licensing is available.

Use the following methods to apply licenses for the Serena Release Manager suite of products.

- **Serena Release Control**: Use Serena License Manager to apply licenses for Serena Release Control.
- **Serena Release Vault**: Use Serena License Manager to apply licenses for Serena Release Vault.
- **Serena Release Automation**: Enter license keys for Serena Release Automation from the Help menu of Nolio.

Documentation References

- Complete documentation on applying licenses for Serena Release Control, which runs in SBM, is in the *Serena Business Manager Licensing Guide* in "Licensing Serena Business Manager".
- Complete documentation on applying licenses for Serena Release Vault is in the *Serena Dimensions CM Administrator's Guide* in "Administering Your Licenses and the License Server".
- Complete documentation on applying licenses for Serena Release Automation is in the *Nolio Automation Center Installation and Administration Guide* in "Updating Nolio Automation Center License".

Promoting and Configuring Serena Release Control in SBM

Serena Business Manager sends requests and receives them from the Serena Release Manager AFS server through Web services and Java.

There is nothing unique to configure in SBM for Serena Release Manager other than what you configure for Serena Release Control.

NOTE It is a prerequisite for SBM to be installed with SSO. See "Installing Serena Business Manager" on page 17. For the communication between SBM and Dimensions CM to work correctly through SSO, the products must share the same SSO server and you must have same user ID for both SBM and Dimensions CM.

You import the Serena Release Control solution, which is a bundle of Process Apps. You then promote the snapshots of each of the Process Apps (Release Train, Application Release, Release Packages, Deployment Task, and Rlm Aux). Finally, you set the privileges, default ownership, and project roles within SBM for the Serena Release Manager administrative user.

Before using Serena Release Control, you need to configure it as follows:

- "Importing the Serena Release Control Solution" on page 27
- "Creating an Environment for Serena Release Control" on page 28
- "Promoting the Snapshots" on page 29
- "Configuring the Administrative User Privileges" on page 30
- "Configuring Default Owners" on page 31
- "Enabling Serena Release Control Project Roles" on page 33

Importing the Serena Release Control Solution

The Serena Release Control application is packaged as an SBM solution. Included in the solution are the following:

- Process apps for Serena Release Control
- Runtime configuration information
- Reports
- Notifications
- Auxiliary table information

The following procedures are documented here to help you through the Serena Release Manager-specific configuration process. For detailed documentation on promoting process apps, see the *Serena Business Manager Application Administrator Guide.*

To use the application, you must import the solution into SBM as follows:

1. Select **Start | Serena Business Manager | SBM Application Administrator**.
2. Enter your login details.
3. In the navigation pane, click **Solutions**.

   The Serena Release Control solution pack is listed. For example, `RLM_Solution_Pack 1.1_RCx`.
4. Click **Import** to import the solution file and extract the solution's snapshots and process apps.

   The **Import Solution** dialog box appears, explaining which snapshots and process apps will be imported with the solution.
5. Click **OK**.

   The solution is now listed under **Imported Solutions**.
6. Click **Open Snapshots** to verify that the following process app snapshots are listed:
   - RLM_AUX
   - Release Train
   - Application Release
   - Release Package
   - Deployment
   - Release Template

Creating an Environment for Serena Release Control

You must create an environment for your Serena Release Control process application unless you are promoting it into an existing environment.

After you create the environment, you must set up a BPEL engine and SBM event manager for that environment.

To create the environment:

1 From the SBM Application Administrator Solutions content pane, click **Environments**.

2 Click **New**.

The **New Environment** dialog box appears.

3 Fill out the form for the new environment as follows:

a Enter a name and description.

b In the **Composer** field, select **Enable Deployment**.

c Under **Application Engine Server**, enter a name and description.

d In the URL, change the server to localhost and specify the port number for the server. For example:

```
http://localhost:80/gsoap/gsoap_ssl.dll?sbminternalservices72
```

4 Click **Test Connection** to test the connection.

5 Click **OK**.

To set up the target servers, the BPEL engine and the event manager:

1 Select the environment you just created and then select the **Target Servers** tab.

2 Click **New**.

The **New Target Server** dialog box appears.

3 Fill out the form to create the BPEL server as follows:

a In the **Type** field, select **BPEL Server (JBPM)**.

b Name the BPEL server. For example: SBM Orchestration.

c Enter the URL using localhost as the hostname.

Port 8085 is the default, unless you specified a different port in the Configurator for your JBOSS server. For example:

```
http://localhost:8085/jbpm-bpel/services/DeployService
```

TIP Above the URL field, click **View Examples** and select from the examples. In the URL field, overtype the hostname and port.

d Click **Test Connection** to test the connection.

e Click **OK**.

4 Click **New**.

The **New Target Server** dialog box appears.

5 Fill out the form to create the event manager server as follows:

- **a** In the **Type** field, select **System Event Manager**.
- **b** Name the event manager server. For example: SBM Event Manager.
- **c** Enter the URL using localhost as the hostname. For example:

```
http://localhost:8085/eventmanager/services/ALFAdmin
```

- **d** Click **Test Connection** to test the connection.
- **e** Click **OK**.

Promoting the Snapshots

You must promote each of the Serena Release Control snapshots and define the destination endpoints for each.

To promote a snapshot:

1 In the navigation pane, click **Solutions** and then select the **Solutions** tab.

2 Select the solution and click **Open Snapshots**.

The list of snapshots appears.

3 Select a snapshot and click **Promote**.

The **Summary** page appears.

4 Click the **Destination** field.

The **Destination** page appears.

5 Select the environment that you created for Serena Release Control.

6 Click **Next**.

The **Entities** page appears.

7 Click **Next**.

The **Mapping** page appears.

8 Select any Source that does not have a Destination Endpoint and click the **Choose Destination Endpoint** button.

9 Select the endpoint from the list.

10 If the Destination Endpoint has not yet been defined for the selected Source, click **Create a new endpoint**.

To create and select an endpoint:

- **a** Name the endpoint.

> **TIP** Name the new endpoint the same as the Source Endpoint for which you are creating it for ease of identification when selecting it as the Destination Endpoint later.

- **b** In the URL field, enter the following:

```
http://localhost:<tomcat port>/rlm/services/<service name>
```

This specifies the path to the Web services, which are deployed under the `rlm/services` directory of the common Tomcat Web server.

Example endpoints are shown in the following table.

Source	Destination Endpoint
RLMUtilService	http://localhost:9095/rlm/services/RLMUtilService
ReleaseRequestService	http://localhost:9095/rlm/services/ReleaseRequestService

c In the **Authentication** field, select **Single Sign On**.

d Click **Test Connection** to test the connection.

e Click **OK**.

f Select the Source Endpoint again and click **Choose Destination Endpoint**. Select the endpoint you just created.

11 Click **Done**.

The **Summary** page appears.

12 Click **Promote**.

The **Promotion Started** page appears.

13 Click **View Log** or **Show Activities** to see results.

14 Repeat the procedure for each snapshot.

PRIVILEGES Privileges for deleting or modifying the Serena Release Control process applications must be set in the SBM Application Administrator. If you need to do these activities, click **Privileges** in the navigation pane and set the privileges according to the SBM documentation.

Configuring the Administrative User

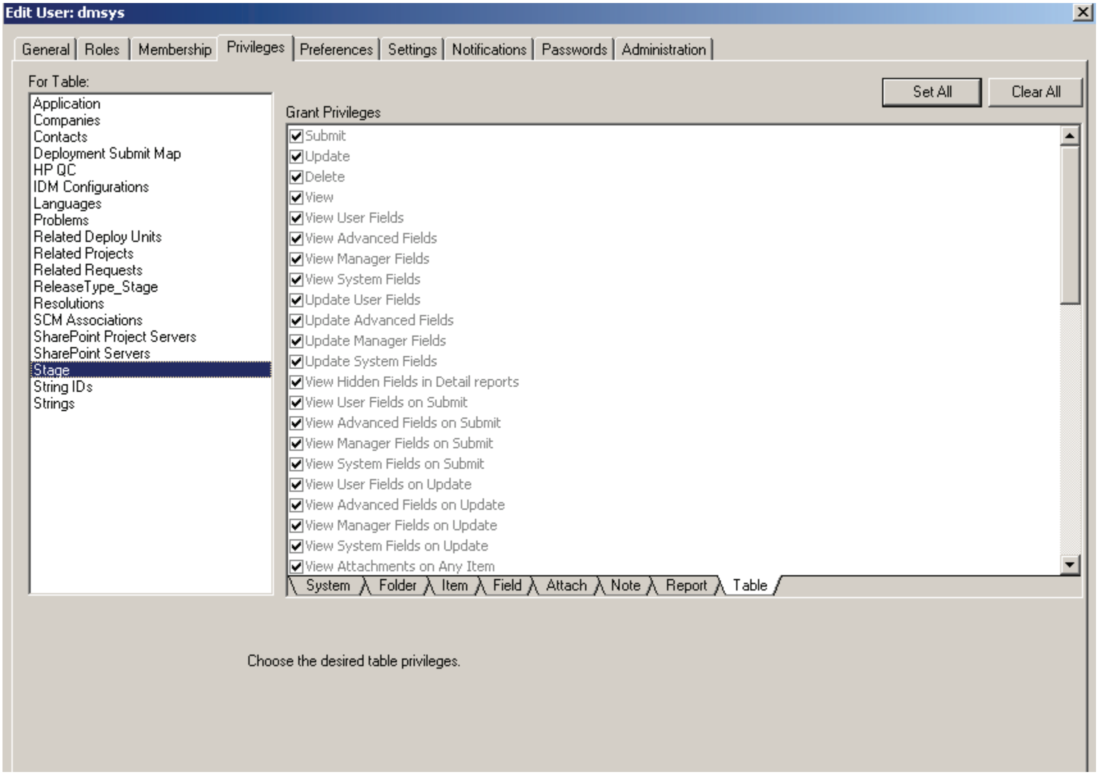
You will use the Serena Release Manager administrative user for single sign-on with Dimensions CM, and the user needs all privileges and project roles for the Dimension CM signon and communication with Serena Release Control through SBM.

Configuring the Administrative User Privileges

In the SBM System Administrator, you must give the administrative user all privileges to the Serena Release Control objects, such as projects, reports, and tables.

Example

The privilege settings for the Stage table are shown in the following figure.



For Table:

- Application
- Companies
- Contacts
- Deployment Submit Map
- HP QC
- IDM Configurations
- Languages
- Problems
- Related Deploy Units
- Related Projects
- Related Requests
- ReleaseType_Stage
- Resolutions
- SCM Associations
- SharePoint Project Servers
- SharePoint Servers
- Stage
- String IDs
- Strings

Grant Privileges

☑ Submit
☑ Update
☑ Delete
☑ View
☑ View User Fields
☑ View Advanced Fields
☑ View Manager Fields
☑ View System Fields
☑ Update User Fields
☑ Update Advanced Fields
☑ Update Manager Fields
☑ Update System Fields
☑ View Hidden Fields in Detail reports
☑ View User Fields on Submit
☑ View Advanced Fields on Submit
☑ View Manager Fields on Submit
☑ View System Fields on Submit
☑ View User Fields on Update
☑ View Advanced Fields on Update
☑ View Manager Fields on Update
☑ View System Fields on Update
☑ View Attachments on Any Item

Choose the desired table privileges.

Documentation References

Complete documentation on managing user privileges in SBM is in the *Serena Business Manager System Administrator Guide* in "Privileges".

Configuring Default Owners

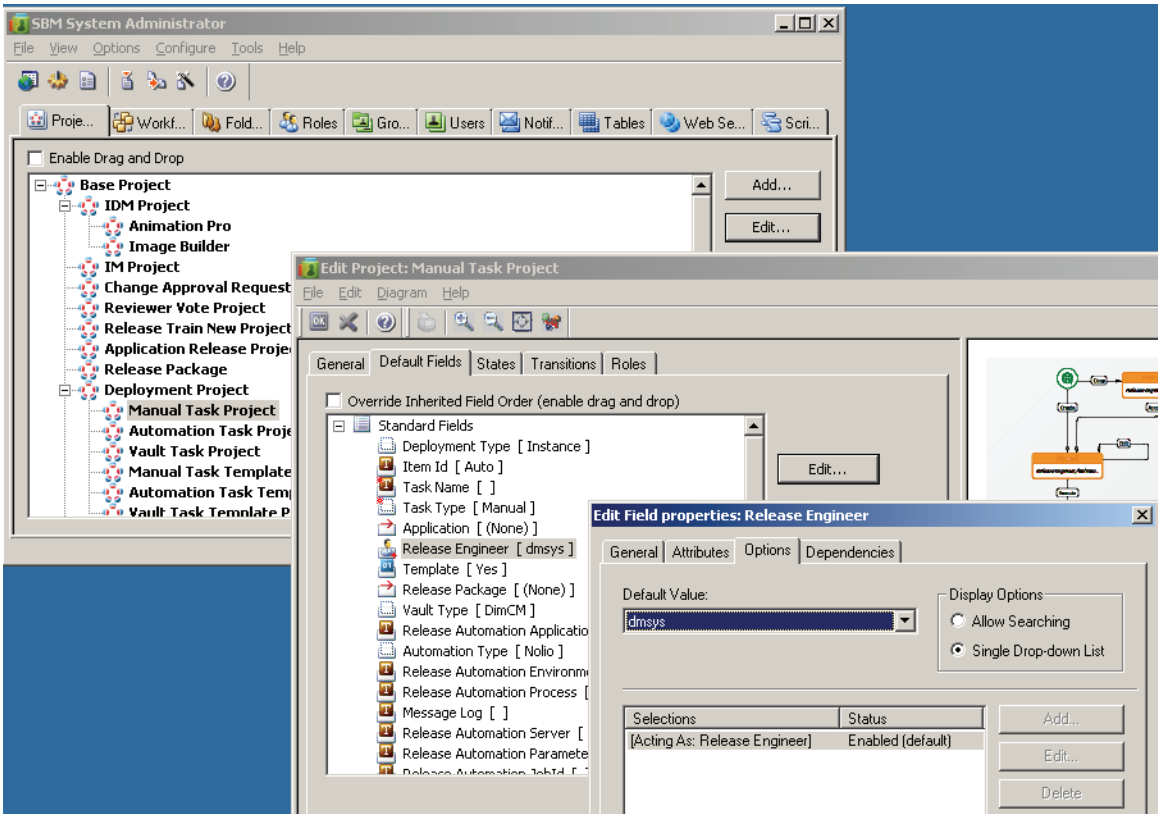
You must select a specific user as owner for the Release Package and Deployment Task projects by selecting a user from the owning role.

This user must have full administrative privileges in both SBM and Dimensions CM.

Example

In the default implementation, the owner value must be set in any sub-projects for a specific user with the ownership role.

The steps for setting the owner for a Deployment Project sub-project, Manual Task Project, are shown in the following figure.



☐ Enable Drag and Drop

☐ Override Inherited Field Order (enable drag and drop)

- Deployment Type [Instance]
- Item Id [Auto]
- Task Name [ ]
- Task Type [Manual]
- Application [(None)]
- Release Engineer [dmsys]
- Template [Yes]
- Release Package [(None)]
- Vault Type [DimCM]
- Release Automation Applicatio
- Automation Type [Nolio]
- Release Automation Environm
- Release Automation Process [
- Message Log [ ]
- Release Automation Server [
- Release Automation Paramete

Default Value: dmsys

○ Allow Searching
● Single Drop-down List

Selections	Status
[Acting As: Release Engineer]	Enabled (default)

Setting Project Owner Default Values

1. From the SBM System Administrator, select the **Projects** tab and select one of the Serena Release Control sub-projects. For example, Manual Task Project under Deployment Project.

2. Click **Edit**.

   The **Edit Project** dialog box appears.

3. Select the **Default Fields** tab.

4. Select the owner field, such as **Release Engineer**, and then click **Edit**.

5. Select the **Options** tab.

6. In **Default Value**, select the administrative user that you will use for SBM and Dimensions CM through SSO. For example, `dmsys`.

7. Click **OK**.

   The user name should appear as the field value following the owner field.

8. Click **OK**.

9. Repeat this procedure for each of the Release Package and Deployment Task sub-projects.

Documentation References

- Complete documentation on promoting process apps in SBM is in the *Serena Business Manager Application Administrator Guide* in "Promoting Process Apps".
- Complete documentation on managing roles in SBM is in the *Serena Business Manager System Administrator Guide* in "Managing User Accounts".

Enabling Serena Release Control Project Roles

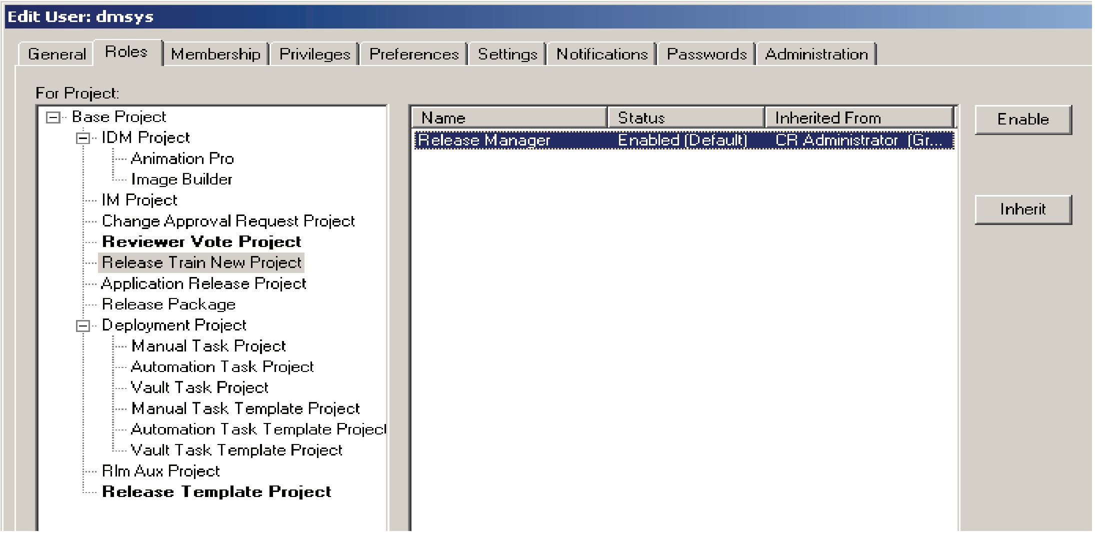
The administrative user must be enabled for all the roles that may have ownership within each of the Serena Release Control projects.

To enable the role for each Serena Release Control project:

1. In the SBM System Administrator, edit the administrative user. For example, `dmsys`.
2. Select the **Roles** tab.
3. Select a Serena Release Control project in the **For Project** tree.
4. For each role that is not enabled, click **Enable**.
5. Repeat for each Serena Release Control project and role.

Example

An example of project role assignments for the Release Manager is shown in the following figure.



For Project:

- Base Project
  - IDM Project
    - Animation Pro
    - Image Builder
  - IM Project
  - Change Approval Request Project
  - **Reviewer Vote Project**
  - Release Train New Project
  - Application Release Project
  - Release Package
  - Deployment Project
    - Manual Task Project
    - Automation Task Project
    - Vault Task Project
    - Manual Task Template Project
    - Automation Task Template Project
    - Vault Task Template Project
  - Rlm Aux Project
  - **Release Template Project**

Name	Status	Inherited From
Release Manager	Enabled (Default)	CR Administrator (Gr...

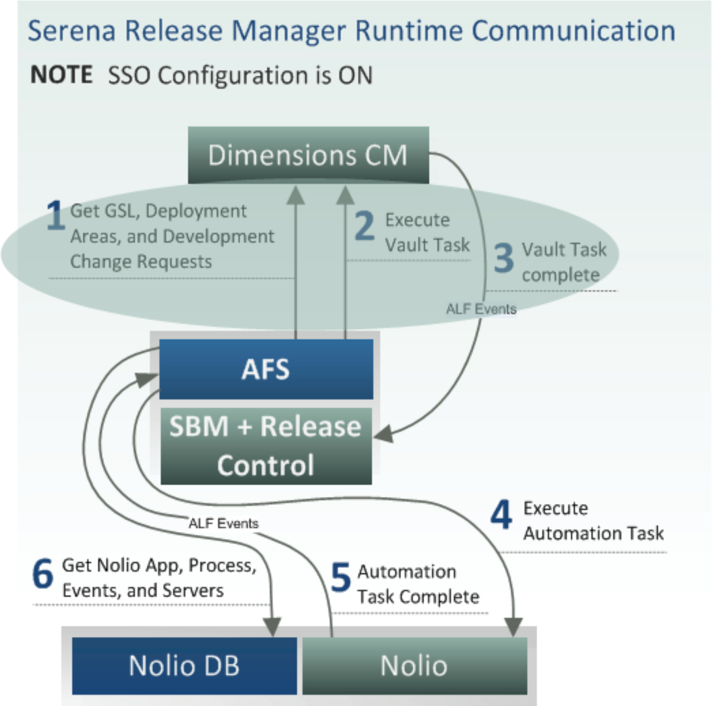
Documentation References

Complete documentation on managing role assignments in SBM is in the *Serena Business Manager System Administrator Guide* in "Managing Roles".

Configuring Communication with Dimensions CM

So that Serena Release Control can get the information from Dimensions CM about the Global Stage Lifecycle, Deployment Areas, and Development Change Requests, you must configure the specific information that AFS and ALF need for your specific environment.

The Release Vault components and communication mechanisms that you are configuring are highlighted in the following figure.


Serena Release Manager Runtime Communication
NOTE SSO Configuration is ON
Dimensions CM
1 Get GSL, Deployment Areas, and Development Change Requests
2 Execute Vault Task
3 Vault Task complete
ALF Events
AFS
SBM + Release Control
4 Execute Automation Task
ALF Events
6 Get Nolio App, Process, Events, and Servers
5 Automation Task Complete
Nolio DB
Nolio

The configuration for Release Vault is included in this section. The configuration activities included in this section are as follows:

- "Connecting Serena Release Control to Dimensions CM" on page 34
- "Connecting Dimensions CM to SBM" on page 35
- "Configuring Dimensions CM to Provide Requests and Baselines" on page 36

Connecting Serena Release Control to Dimensions CM

So that Serena Release Control can connect and send information to Dimensions CM, you must update the Dimensions CM Web service properties file with the correct connection information.

To update the Dimensions CM connection information:

1. On your SBM server, navigate to the Serena Release Manager common Tomcat Web server `classes` folder. For example:

   ```
   C:\Program Files\Serena\common\tomcat\6.0\webapps\rlm\WEB-
   INF\classes
   ```

2. Open the `dimensions.properties` file.

3. Make sure the variables are set to the correct connection information for your installation of Dimensions CM.

Example

`dimensions.properties`

```
DIM_WS_URL = http://sys-dev-srv1:8080/dmwebservices2/services/
    dmwebservices/

DIM_DBNAME = cm_typical
DIM_DBCONN = Dim12
DIM_SERVER = sys-dev-srv1

JOB_STATE_SUCCESS = Succeeded
JOB_STATE_FAILED = Failed

# for testing
DIM_USERID = dmsys
DIM_PASSWORD = dmsys_pswd
```

Connecting Dimensions CM to SBM

So that Serena Release Control can receive information from Dimensions CM, you must update the Dimensions CM configuration settings with the endpoints and signon credentials SBM uses for the connection.

The connection of SBM with Dimensions CM is implemented using ALF events. Serena Release Control will get the information through SBM.

To update the Dimensions CM endpoints and user IDs for SBM:

1. On the Dimensions CM server, navigate to the Dimensions CM installation directory. For example:

   ```
   C:\Program Files\Dimensions 12.1\CM
   ```

2. Update the `dm.cfg` file to add or update ALF variables as follows:

   ```
   #######################################
   # ALF Events Configuration
   #######################################
   DM_ALF_ENDPOINT http://<dim_server>:<port#>/eventmanager/services/ALFEventManager
   DM_ALF_USER <user>
   DM_ALF_PASSWORD <password>
   DM_ALF_EVENT_CONFIG %DM_DFS%alf_events_config.xml
   DM_ALF_PRODUCT_INSTANCE DimensionsUnderReleaseManagementControl
   ```

   where

   - `DM_ALF_ENDPOINT` is pointing to the Dimensions CM server and port number and the ALF Event Manager for that Dimensions CM server
   - `DM_ALF_USER` is a valid SBM and Dimension CM user with administrative privileges

Example

`dm.cfg`

```
#######################################
# ALF Events Configuration
#######################################
DM_ALF_ENDPOINT http://dmhost:8085/eventmanager/services/ALFEventManager
DM_ALF_USER dmsys
DM_ALF_PASSWORD dmsys_pswd
DM_ALF_EVENT_CONFIG %DM_DFS%alf_events_config.xml
DM_ALF_PRODUCT_INSTANCE DimensionsUnderReleaseManagementControl
```

Configuring Dimensions CM to Provide Requests and Baselines

If you plan to use Dimensions CM to provide requests for your Development Change Requests or baselines as your Deployment Units, you must configure the ALF events to enable this information to be sent to Serena Release Control.

Configuring Dimensions CM to provide requests and baselines includes the following activities:

- "Specifying Selection Criteria for Projects and Baselines" on page 36

Specifying Selection Criteria for Projects and Baselines

So that Serena Release Control knows which project and baseline information to get from Dimensions CM, you must update the ALF configuration settings with the necessary information.

To update the Dimensions CM configuration settings:

1. Navigate to the Dimensions CM installation directory. For example:

   `C:\Program Files\Dimensions\12.1\CM\dfs`

2. Update the `ALF_EVENTS_CONFIG.XML` file to specify your Dimensions CM database name, project name, baseline type, and deploy event.

Example

ALF_EVENTS_CONFIG.XML

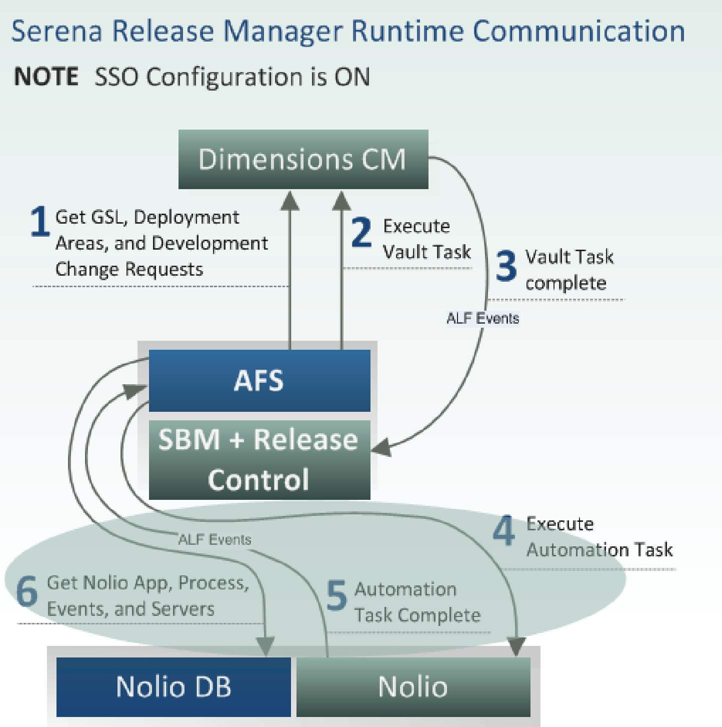
```
  <!--       Specify a specific database-->
<Database>
    <!--      Example database name specification
    <base_database_id>@<server>-<db_connection>
   (using the wildcard character "*")-->
             <Name>CM_TYPICAL@*-Dim12</Name>
    <!--Specify one or more projects-->
    <Projects>
     <!--Specify a specific  project-->
     <Project
        <!-- Example project name specification
        <project-spec> - <product-id>:<project-id>
       (using the wildcard character "*")-->
             <Name>*</Name>
      <!--Specify one or more object classes-->
      <Objects>
      <!-- Specify a specific object class -->
       <Object>
             <Type>Baseline</Type>
        <!-- Example events to support for  above object class-->
        </Events>
             <Event>Deploy</Event>
        </Events>
       </Object>
      </Objects>
     </Project>
    </Projects>
</Database>
```

Configuring Communication with Serena Release Automation

So that Serena Release Control can get the information from Serena Release Automation about Nolio applications, processes, events, and server lists, you must configure the specific information the ALM Foundation Services (AFS) server and Application (ALF) need for your specific environment.

You should have already defined typical Nolio server properties, database name, location, and so on during the installation. After the installation, you need to configure additional properties files so that Nolio can communicate through AFS and ALF with SBM.

The Release Automation components and communication mechanisms that you are configuring are highlighted in the following figure.


Serena Release Manager Runtime Communication
NOTE SSO Configuration is ON
Dimensions CM
1 Get GSL, Deployment Areas, and Development Change Requests
2 Execute Vault Task
3 Vault Task complete
ALF Events
AFS
SBM + Release Control
ALF Events
4 Execute Automation Task
5 Automation Task Complete
6 Get Nolio App, Process, Events, and Servers
Nolio DB
Nolio

The configuration activities included in this section are as follows:

- "Configuring Nolio Event Notification" on page 38
- "Specifying Nolio Database Connection Information" on page 43

Configuring Nolio Event Notification

When an automation task runs, a Nolio event is invoked. That event is sent to the AFS server. To request the Nolio event, AFS needs the Nolio server information. Then, to return the response for the event, Nolio needs to know your AFS server information.

Configuring the Nolio event notification includes the following activities:

- "Specifying the Nolio Server to Notify" on page 39
- "Telling Nolio Which Event Notifications to Send" on page 39

- Connecting SBM to Nolio

Specifying the Nolio Server to Notify

You must update the `rest.integration.properties` file to tell Nolio what server to notify when an event occurs.

To specify the Nolio server:

1 On the Nolio server, navigate to the Nolio installation directory. For example:

```
C:\Program Files\Nolio\NolioAutomationCenter\conf
```

2 Open the Nolio `rest.integration.properties` file.

If the file does not exist, create it.

3 Set the `target.url` variable to point to the AFS `NolioNotification` servlet as follows:

```
target.url=http://<AFS_server>:<tomcat_port>/rlm/servlet/
NolioNotification
```

For example:

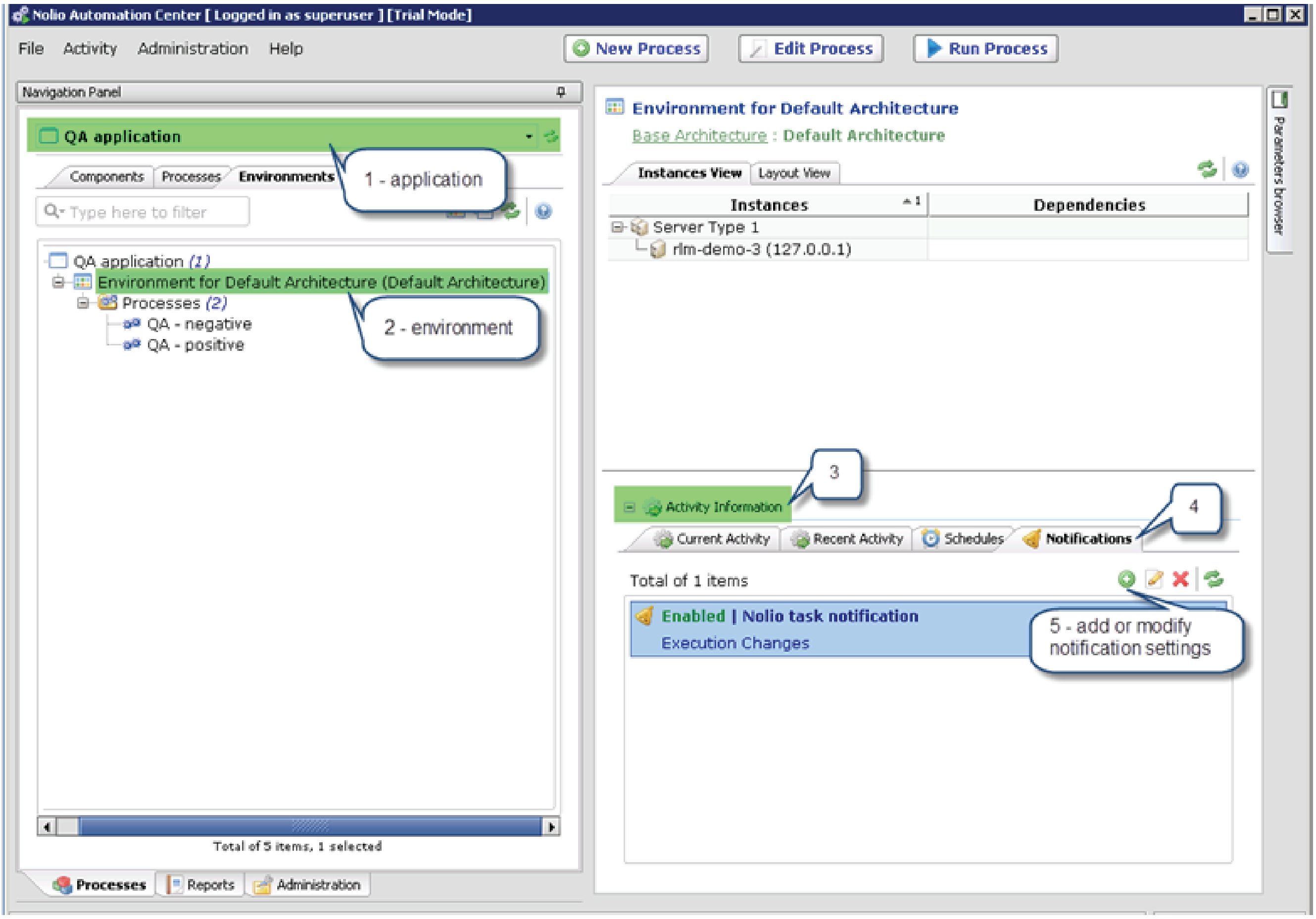
```
target.url=http://rlmhost:8080/rlm/servlet/NolioNotification
```

4 Restart the Nolio server and agents.

Telling Nolio Which Event Notifications to Send

You must update the Nolio environment notifications for each application to tell Nolio which events to notify the AFS server about.

Nolio Automation Center is shown in the following figure.



Instances	Dependencies
Server Type 1	
rlm-demo-3 (127.0.0.1)	

To configure Nolio environment notification:

1 From Nolio Automation Center, for each application, select the application.

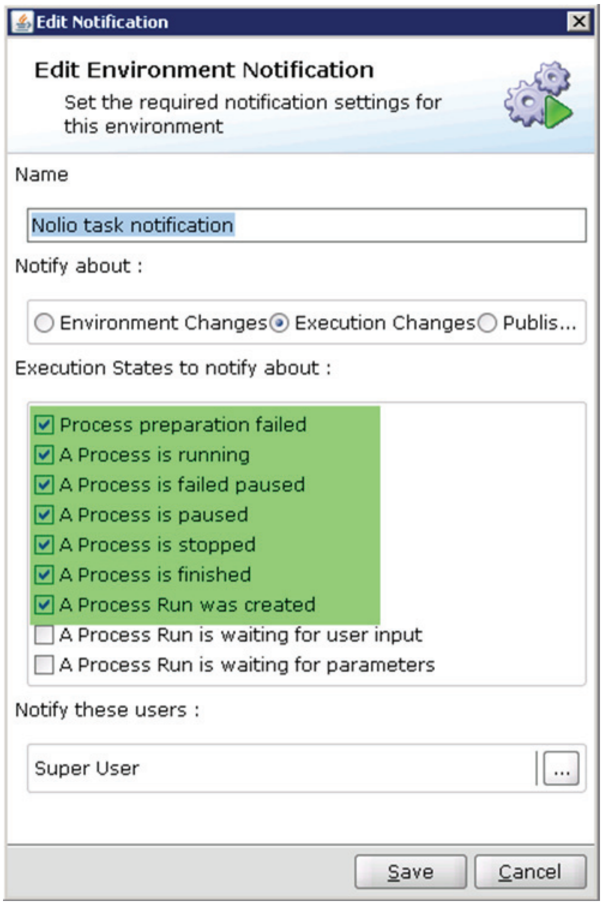
2 For each environment, select the environment.

3 Expand the **Activity Information** section.

4 Select the **Notifications** tab.

5 Click the **Add or modify notification settings** button.

The Edit Environment Notification dialog box appears.



Edit Environment Notification

Set the required notification settings for this environment

Name

Nolio task notification

Notify about :

○ Environment Changes ● Execution Changes ○ Publis...

Execution States to notify about :

☑ Process preparation failed
☑ A Process is running
☑ A Process is failed paused
☑ A Process is paused
☑ A Process is stopped
☑ A Process is finished
☑ A Process Run was created
□ A Process Run is waiting for user input
□ A Process Run is waiting for parameters

Notify these users :

Super User

6 Select the **Execution States to notify about** as shown in the preceding figure.

7 Click **Save**.

Connecting Nolio to SBM

Connecting SBM to Nolio includes the following:

- "Specifying ALF Event Credentials" on page 41
- "Installing the JDBC driver for your Database Connection" on page 42
- "Specifying Nolio Database Connection Information" on page 43

Specifying ALF Event Credentials

So that Serena Release Control can receive information from Serena Release Automation, you must update the ALF event properties file used by AFS with signon credentials SBM uses for the connection. AFS sends ALF events to SBM after it is notified from Nolio that requested events have been processed. Serena Release Control gets the information through SBM.

To update the Nolio ALF event signon credentials needed by SBM:

1 Navigate to the Serena Release Manager common Tomcat Web server `classes` folder. For example:

```
C:\Program Files\Serena\common\tomcat\6.0\webapps\rlm\WEB-INF\classes
```

2 Open the `alf.properties` file.

3 Set the `AE_USERID` and `AE_PASSWORD` variables with the user ID and password of your Serena Release Manager administrative user.

Example

`alf.properties`

```
ALF_EVENTMANAGERURL = http://localhost:8085/eventmanager/services/ALFEventManagerDocLit
AE_USERID = dmsys
AE_PASSWORD = dmsys_test
```

Installing the JDBC driver for your Database Connection

If you are using Serena Release Automation with Serena Release Control and you are using a database management system other than Oracle, you must download the JDBC driver needed to connect to your database.

To install the JDBC driver for your database connection:

1 Download the JDBC drivers.

For MySQL, you can download the JDBC driver from here:

http://www.mysql.com/downloads/connector/j/

For SQL Server, you can download the JDBC driver from here:

http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=2505

2 Extract and place the `.jar` file from the root location into the Serena Common Web server `rlm\WEB-INF\lib` folder on the Serena Release Manager server. For example, copy

```
mysql-connector-java-5.1.18-bin.jar
```

to

```
C:\Program Files\Serena\common\tomcat\6.0\\webapps\rlm\WEB-INF\lib
```

3 Restart the Serena Common Tomcat service.

IMPORTANT! The Oracle database JDBC driver is included with and licensed as part of the Serena Release Automation installation.

If you want to use MySQL or SQL Server with Serena Release Automation, you must obtain and install the JDBC drivers for those separately from the Serena Release Manager installation. You should obtain licenses for those products as required by the providers.

Specifying Nolio Database Connection Information

After you configure the Nolio event notification, you must specify the connection information for Serena Release Control to access it through the AFS server.

NOTE The database used by Nolio must be open for remote connection so that Serena Release Control can access it through the JDBC driver and the AFS server. See your Database Administrator to ensure this is done.

To specify the Nolio database connection information:

1. Navigate to the Serena Release Manager common Tomcat Web server `classes` folder. For example:

   ```
   C:\Program Files\Serena\common\tomcat\6.0\webapps\rlm\WEB-INF\classes
   ```

2. Open the `nolio.properties` file.

3. Set the properties for the database connection as follows:

Parameter	Value
nolio.db.driver.class.name	The Web service class name for the Nolio database JDBC driver. For example: `com.mysql.jdbc.Driver`

nolio.db.jdbc.url	The URL that points to the Nolio database, in the form shown as follows for each of the supported database systems.
	MySQL
	`jdbc:mysql://<NolioHostName>:<DBPort>/nolio_db?useUnicode=true&characterEncoding=utf-8`
	SQL Server
	`jdbc:sqlserver://server_name:port;user=login;password=****;database=name;...`
	Oracle
	`jdbc:oracle:thin:[user/password]@[host][:port]:SID`

nolio.db.user	The Nolio database user name.
nolio.db.pwd	The Nolio database password.
nolio.ws.uri	The URL that points to the Nolio execution server, in the form of: `http://<NolioHostName>:<Execution ServerPort>/datamanagement/ws/ExecutionRelayWS`
nolio.job.state.success	List of Nolio job states that indicate success, delimited by commas.
nolio.job.states.failure	List of Nolio job states that indicate failure, delimited by commas.
nolio.username	The Nolio execution server user name.
nolio.password	The Nolio execution server password.

Example

MySQL DB

This example sets the values needed to retrieve successful and failed job notifications from Nolio using a MySQL database. Login credentials are specified for the database and for the Nolio execution server.

MySQL configuration in `nolio.properties`

```
nolio.db.driver.class.name = com.mysql.jdbc.Driver
nolio.db.jdbc.url = jdbc:mysql://sys-qa-test4:3306/
     nolio_db?useUnicode=true&characterEncoding=utf-8
nolio.db.user = root
nolio.db.pwd = password
nolio.ws.uri = http://sys-qa-test4:7080/datamanagement/ws/ExecutionRelayWS

nolio.job.state.success = FLOW_FINISHED
nolio.job.states.failure =
     BLOCKED,CREATION_FAILED,FILES_DISTRIBUTION_FAILED,FILES_PROPAGATION_FAILED,FLOW_FA
     ILED_PAUSED,PRE_FAILED

nolio.username = superuser
nolio.password = suser
```

Configuring Access to Development Change Requests and Deployment Units

A Development Change Request (DCR) Provider is an integration between Serena Release Control and a change request system. The default implementation includes provider connections for SBM and Dimensions CM so that you can also associate DCRs with SBM issues and Dimensions CM requests.

A Deployment Unit (DU) Provider is an integration between Serena Release Control and a change management system. The default implementation includes a provider connection for Dimensions CM, so that you can also associate deployment units with Dimensions CM baselines.

This section tells how to configure DCR and DU access from SBM and Dimensions CM. For information on customizing your Serena Release Manager implementation to use providers for DCRs and DUs other than SBM and Dimensions CM, see "Adding Development Change Request and Deployment Unit Providers" on page 92.

Designating Providers

The designation of DCR and DU provider information is divided into two steps as follows:

1. "Designating the Details for Each Provider" on page 44
2. "Selecting the Providers" on page 47

Designating the Details for Each Provider

You should give the connection details for each DCR and DU provider in separate properties file for each instance. This keeps the details separate from the selection, and helps with maintenance and security.

To designate details for each DCR and DU provider:

1 Navigate to the Serena Release Manager common Tomcat Web server `classes` folder. For example:

```
C:\Program Files\Serena\common\tomcat\6.0\webapps\rlm\WEB-
INF\classes
```

2 Open one of your provider's properties files. For example:

- `dm_qlarius.properties`
- `sbm_issues.properties`
- `sbm_incidents.properties`

3 The details are unique for each properties file, and variables and parameters are defined in the implementation for the provider. See Chapter 6, "Adding Development Change Request and Deployment Unit Providers" on page 92 for details if you plan to use a provider other than Dimensions CM or SBM for DCRs and DUs.

For example settings for the default Dimensions CM and SBM properties files, see the following examples.

Examples

Dimensions CM

Set the Dimensions CM provider details for DCRs and DUs. This example sets the values needed to retrieve requests and deployment units (baselines) from the Dimensions CM sample database, which has sample data based on a fictitious company called Qlarius.

`dm_qlarius.properties`

```
# requests provider definitions
requests.provider.name = DIM_QLARIUS
requests.provider.description = Dimensions Requests Provider for QLARIUS product

# deploy units provider definitions
deploy.units.provider.name = DIM_QLARIUS
deploy.units.provider.description = Dimensions Deployment Unit Provider for QLARIUS
     product

# filter requests by statuses
FILTER_REQUEST_BY_STATUSES = IN QA,IN PROGRESS,UNDER WORK,IN TEST

# filter deploy units by statuses
FILTER_DEPLOY_UNITS_BY_STATUSES = VERIFIED,DEPLOYED,CAPTURED
```

The text following the keys, `requests.provider.name` and `deploy.units.provider.name`, is documentary. In this example, `DIM_QLARIUS` simply describes the database from which the requests are being retrieved. The actual connection to the Dimensions CM database is defined in the `dimensions.properties` file. See "Connecting Serena Release Control to Dimensions CM" on page 34.

This example tells Dimensions CM to return only requests and deployment units in a specified list of statuses.

SBM Issues

Set the SBM provider details for DCRs. This example sets the values needed to retrieve requests from the SBM Issues process app.

`sbm_issues.properties`

```
# requests provider definitions
requests.provider.name  = Issues
requests.provider.description = SBM Requests Provider based on Issues solution

REQUESTS_TABLE_DBNAME = UBG_ISSUES
REQUESTS_FIELD_STATUS = STATE
REQUESTS_FIELD_LINK = URL
REQUESTS_FIELD_OWNER = OWNER
REQUESTS_FIELD_PROJECTNAME= PROJECTID

REQUESTS_QUERY_WHERE_CLAUSE =
REQUESTS_ORDER_BY_CLAUSE    =

# possible values
# SUBMIT_PROJECTS
# REPORT_PROJECTS
REQUESTS_PROJECTS_TYPE= SUBMIT_PROJECTS

# valid only for REQUESTS_PROJECTS_TYPE = REPORT_PROJECTS
PROJECTS_REPORT_NAME     =
PROJECTS_FIELD_TITLE=
PROJECTS_FIELD_STATUS =
PROJECTS_FIELD_OWNER =
PROJECTS_FIELD_TYPE =
# end of properties specific for REQUESTS_PROJECTS_TYPE = REPORT_PROJECTS
```

The text following the keys, `requests.provider.name` and `deploy.units.provider.name`, is documentary. In this example, `Issues` simply describes the SBM primary table from which the issues, or requests, are being retrieved. The REQUESTS_TABLE_DBNAME key specifies the primary table from which to retrieve the issues.

This example shows a number of filters that can be used to restrict the list of issues to associate with DCRs in a release package.

SBM Incidents

Set the SBM provider details for DCRs. This example retrieves incidents from the Incidents process app. This process app must be on the same SBM application server as your Serena Release Control process app.

`sbm_incidents.properties`

```
# requests provider definitions
requests.provider.name  = Incidents
requests.provider.description = SBM Requests Provider based on Incidents solution

REQUESTS_TABLE_DBNAME = UIM_INCIDENTS
REQUESTS_FIELD_STATUS = STATE
REQUESTS_FIELD_LINK = URL
REQUESTS_FIELD_OWNER = OWNER
REQUESTS_FIELD_PROJECTNAME= PROJECT_FOR_INCIDENT

REQUESTS_QUERY_WHERE_CLAUSE =
REQUESTS_ORDER_BY_CLAUSE    =

# possible values
# SUBMIT_PROJECTS
# REPORT_PROJECTS
REQUESTS_PROJECTS_TYPE= REPORT_PROJECTS

# all properties below are valid only for REQUESTS_PROJECTS_TYPE = REPORT_PROJECTS
PROJECTS_REPORT_NAME     = Projects for Incidents
PROJECTS_FIELD_TITLE= TITLE
PROJECTS_FIELD_STATUS = STATE
PROJECTS_FIELD_OWNER = OWNER
PROJECTS_FIELD_TYPE = ISSUETYPE
# end of properties specific for REQUESTS_PROJECTS_TYPE = REPORT_PROJECTS
```

The text following the keys, `requests.provider.name` and `deploy.units.provider.name`, is documentary. In this example, `Incidents` simply describes the SBM primary table from which the incidents, or requests, are being retrieved. The REQUESTS_TABLE_DBNAME key specifies the primary table from which to retrieve the incidents.

This example shows a number of filters that can be used to restrict the list of issues to associate with DCRs in a release package .

Selecting the Providers

Once you have defined the login information for the providers in separate properties files, you tell Serena Release Manager which providers you want to use by specifying those properties file names in the providers properties file.

You can select one or more providers for DCRs and DUs.

To designate the DCR and DU providers:

1. Navigate to the Serena Release Manager common Tomcat Web server `classes` folder. For example:

   `C:\Program Files\Serena\common\tomcat\6.0\webapps\rlm\WEB-INF\classes`

2. Open the `providers.properties` file.

3. Set `requests.providers.keys` to the DCR provider value or values that you want Serena Release Manager to use. This is the first node of the file name you used for the properties file you updated in "Designating the Details for Each Provider" on page 44. For example,

   `requests.providers.keys = sbm_issues`

tells Serena Release Manager to use the `sbm_issues.properties` file.

4 Set `deploy.units.providers.keys` to the DU provider value or values that you want Serena Release Manager to use. This is the first node of the file name you used for the properties file you updated in "Designating the Details for Each Provider" on page 44. For example,

```
deploy.units.providers.keys = dm_qlarius
```

tells Serena Release Manager to use the `dm_qlarius.properties` file.

Example

`providers.properties`

```
# requests provider keys
requests.providers.keys = sbm_issues

# deploy units provider keys
deploy.units.providers.keys = dm_qlarius
```

Chapter 4

Configuration and Administration of the Integrating Entities

This chapter tells you how to configure and administer entities in the integrating products to enable and support your organization's release management activities. These are ongoing administrative tasks, such as adding application names for your organization and managing users, reports, and notifications in Serena Release Control, and managing projects and streams in Dimensions CM.

You need to complete the following configuration before the people who participate in release management in your organization begin using Serena Release Control.

Configuring Entities in Serena Release Control

Before you begin using Serena Release Control or Serena Release Manager, you must configure the entities needed to use its features.

The following sections explain the configuration and administration needed for Serena Release Control:

- "Accessing the Standard SBM User Interface" on page 50
- "Adding Your Application Names in Serena Release Control" on page 50
- "Managing Release Control Users" on page 51
- "Configuring Reports" on page 53
- "Configuring Notifications" on page 55

Accessing the Standard SBM User Interface

Serena Release Control runs in a shell UI on top of the standard SBM user interface of the SBM Web client. However, you will do most of your custom configurations using the standard SBM user interface.

To display the Serena Release Control standard SBM user interface:

1 Login to the Serena Release Control Web client.

2 Remove the shell parameter from the URL.

For example, if your Serena Release Manager URL is:

```
http://rlmhost/tmtrack/tmtrack.dll?shell=rlm
```

your Serena Release Control standard SBM user interface URL would be:

```
http://rlmhost/tmtrack/tmtrack.dll?
```

Adding Your Application Names in Serena Release Control

Before you start releasing applications using Serena Release Control, you must pre-populate Serena Release Control with the names of your applications that you plan to release over time using Serena Release Manager. This enables you to associate those applications with the release information that you enter in Serena Release Control.

What Can You Change?

You can change the following application information:

- Add application name and description.

What is the Impact?

The application names appear in the **Application Release** dialog box in the **Associate to application** selection field. If you change the names or add names, this impacts the list of names the users see when they select applications for an application release.

How Do You Change It?

You add and change application information in the Application auxiliary table.

To change the Application table entries:

1. From the Serena Release Control standard SBM user interface navigation pane, click the **Search** link, and then click **Manage Data**.
2. Click **Create New Item** and enter the application name and description.
3. Click **OK** to save.

Documentation References

Complete documentation on configuring tables in SBM is in the *Serena Business Manager System Administrator Guide* in "Table Configuration".

Managing Release Control Users

Before your Serena Release Manager users begin logging into Serena Release Control, you must configure the user information in SBM, which is used to:

- Login and access Serena Release Control functionality through the Serena Release Control user interface
- Assign ownership in the process workflow, to the SBM workflow states
- Access SBM reports and notifications used by Serena Release Control

Adding Users and Groups

You add users and groups as you would normally do in SBM.

What Can You Change?

You can change the following user information:

- Add users and groups
- Add new roles
- Modify roles
- Change ownership in existing states
- Change privileges of roles that are assigned to states
- Assign roles in projects

What is the Impact?

If you change users, you will also need to:

- If an individual user is selected as a primary owner of a release item, and that user is removed, you will need to select another user as owner before the item can be progressed in the workflow.

You may need to add the roles that are needed in Dimensions CM for performing the necessary actions, as they won't be in the default process model. To define roles, see the Users and Roles chapter in the Process Modeling User's Guide.

If you change or add roles, you will also need to:

- Add roles in projects for any workflows in which this role may be assigned ownership.
- Change privileges of roles that are assigned to states.

How Do You Change It?

You manage users according to the SBM documentation.

- You can create the roles with privileges in SBM Composer.
- You can add users and groups to roles in SBM System Administrator.
- You can grant privileges not related to roles, such as administrative privileges, in SBM System Administrator.

Documentation References

- Complete documentation on managing roles in SBM is in the *Serena Business Manager SBM Composer Guide* in "Managing Roles".
- Complete documentation on managing users in SBM is in the *Serena Business Manager System Administrator Guide* in "Managing User Accounts".

Example

The default roles defined in Serena Release Control are Release Manager and Release Engineer. Typical release management roles are shown in the following table for your reference:

Role Name	Description
Application Owner	The business owner of an application. An application owner is responsible for approving an application release during the planning stage and receives notifications of application release status.
Build Manager	The role that performs builds for a release. This role may also be the Installation Manager in some organizations.
Change Manager	The role that is responsible for the IT Operations for the systems where the pre-production and production release environments reside. A change manager approves deployment into pre-production and production environments.
Development Manager	The role that is responsible for and approves development activities for a release. A development manager would typically be consulted during the release management process and provide approval on the content of a release package.
Installation Manager	The role that ensures that the deployment, or installation, of a release is done correctly and completely. An installation manager is assigned manual deployment tasks in Serena Release Control and is responsible for deployment of request packages into environments. This role may also be the Build Manager in some organizations.

Role Name	Description
QA Manager	The role that is responsible for and approves testing activities for a release.
Release Engineer	The role that is responsible for the automating the release deployment and resolving any deployment failures. A release engineer creates deployment tasks for release packages using Nolio. This role may be the Installation Manager or Build Manager in some organizations.
Release Manager	The role that is responsible for releases within a particular organization. A release manager manages and monitors releases, plans releases in collaboration with development managers, and responds to successes, failures, and other statuses.

Managing Release Control Reports and Notifications

Reports and notifications help keep release management stakeholders informed of status, history, and other release information. Serena Release Control provides a default set of reports and notifications, and you can configure these as needed to support your organization.

Configuring Reports

Some of the Serena Business Manager reports are used to populate the Serena Release Control user interface views. You can configure those as documented in Chapter 6, "Customizing the User Interface" on page 72.

In addition to reports used for the UI views, you can configure general reports to provide the information you need for your organization’s release control.

The general reports configured by default for Serena Release Control are shown in the following table.

Entities	Reports	Description
Release Trains	Dashboard	A multi-view report for the **Dashboard** page. See Chapter 6, "Configuring the Dashboard" on page 72.
	Inactive RTrains	All inactive release trains.
	Release train by type	Release trains by type.
	Release Trains in Gantt	Release trains in a Gantt-style format, used for the **Calendars** page. See Chapter 6, "Configuring the Calendar" on page 73.
	Train All	All release trains.
	train Assigned To Current User	All release trains assigned to the current user, used for the **My Inbox** page. See Chapter 6, "Configuring the Inbox" on page 74.

Entities	Reports	Description
Application Releases	Application releases by application and type	Application releases by application and type.
	Application All	All application releases.
	application Assigned To Current User	Application releases assigned to the current user, used for the **My Inbox** page. See Chapter 6, "Configuring the Inbox" on page 74.
	Application Releases In Release Train	Application releases in the selected release train.
	Application Releases Unassigned	Application releases that are not yet assigned to release trains.
Release Packages	Unassigned Release Packages	Release packages that are not yet assigned to application releases.
	Package All	All release packages.
	package Assigned To Current User	Release packages assigned to the current user, used for the **My Inbox** page. See Chapter 6, "Configuring the Inbox" on page 74.
	Release Packages in Application Release	Release packages in the selected application release.
	Release packages list by owner	Release packages list by owner.
Deployment Tasks	all DT	All deployment tasks.
	Assigned Tasks	Deployment tasks assigned to the selected release package stage.
	Deployment task I own	Deployment tasks owned by the current user.
	task All	All deployment tasks.
	task Assigned To Current User	Deployment tasks assigned the current user, used for the **My Inbox** page. See Chapter 6, "Configuring the Inbox" on page 74.
	Template Tasks	Tasks in the selected deployment process template.
Other	Assigned Deployment Units	Deployment units in the selected release package.
	Assigned Development Change Requests	Development change requests in the selected release package.
	All Release Type Stages	All stages sorted by release type and sequence.
	Stages in Release Type	All stages for the selected release type.
	All Stages	All stages.

What Can You Change?

You can change the following report information:

- Add reports
- Modify reports that you created
- Manage access to reports

What is the Impact?

- If you add a report, you must give users access to the report.
- If you add reports, the new reports will appear in the Serena Release Control Reports page for those users with access to them.

How Do You Change It?

You can manage reports from the Serena Release Control standard SBM user interface according to the SBM documentation.

NOTE End users with the proper privileges can add reports from the Reports view in the Serena Release Control UI shell by editing a report and using the Save As option to save it to another report name. However, only Serena Release Control administrators should be given the privileges to modify reports provided with Serena Release Control.

Documentation References

Complete documentation on configuring SBM reports is in the *Serena Business Manager User's Guide* in "Working with Reports".

Configuring Notifications

E-mail notifications are sent to Serena Release Control users to alert them of actions requiring their attention, provide important status information on releases, and so on.

The e-mail notifications configured by default for Serena Release Control are shown in the following table.

Entity	Notifications
Release Train	RTN - Any Release Train changes owner
	RTN - Any Release Train changes state
	RTN - Any Release Train changes to inactive
	RTN - Any Release Train I submitted changed state
	RTN - Any Release Train I submitted changed to inactive
	RTN - Any Release Train is submitted
	RTN - I become the owner of any Release Train
Application Release	AR - Any Application Release changes owner
	AR - Any Application Release changes state
	AR - Any Application Release changes to inactive
	AR - Any Application Release I submitted changed state
	AR - Any Application Release I submitted changed to inactive
	AR - Any Application Release is submitted
	AR - I become the owner of any Application Release

Entity	Notifications
Release Package	RP - Any Release Package changes owner
	RP - Any Release Package changes state
	RP - Any Release Package changes to inactive
	RP - Any Release Package I submitted changed state
	RP - Any Release Package I submitted changed to inactive
	RP - Any Release Package is submitted
	RP - I become the owner of any Release Package
Deployment Task	D - Any Deployment changes owner
	D - Any Deployment changes state
	D - Any Deployment changes to inactive
	D - Any Deployment I submitted changed state
	D - Any Deployment I submitted changed to inactive
	D - Any Deployment is submitted
	D - I become the owner of any Deployment
Deployment Process Template	RT - Any Release Template changes owner
	RT - Any Release Template changes state
	RT - Any Release Template changes to inactive
	RT - Any Release Template I submitted changed state
	RT - Any Release Template I submitted changed to inactive
	RT - Any Release Template is submitted
	RT - I become the owner of any Release Template
Other (Auxiliary)	RA - Any Rlm Aux changes owner
	RA - Any Rlm Aux changes state
	RA - Any Rlm Aux changes to inactive
	RA - Any Rlm Aux I submitted changed state
	RA - Any Rlm Aux I submitted changed to inactive
	RA - Any Rlm Aux is submitted
	RA - I become the owner of any Rlm Aux

What Can You Change?

You can modify the provided notifications and configure additional notifications as needed to support your release management processes.

Notifications in SBM are e-mail messages sent to users when certain events or conditions occur in the system. Notifications can also be used to automatically add and remove items from folders and to execute scripts.

You can change the following notification information:

- Add notifications

- Modify notifications
- Subscribe users to notifications

What is the Impact?

If you add a notification, you must subscribe users to it so that they will be sent to them upon the event associated with the notification.

How Do You Change It?

You can manage notifications in SBM System Administrator according to the SBM documentation.

Documentation References

Complete documentation on configuring SBM notifications is in the *Serena Business Manager System Administrator Guide* in "Managing Notifications".

Configuring Entities in Dimensions CM

Before you begin using the Serena Release Control features that integrate with Dimensions CM, you must configure the Dimensions CM entities needed.

The following sections explain the configuration and administration needed for Serena Release Control:

- "Configuring the Dimensions CM Global Stage Lifecycle" on page 57
- "Managing Dimensions CM Users" on page 58
- "Configuring Dimensions CM Projects and Streams" on page 59
- "Considerations for Requests and Baselines" on page 60

Configuring the Dimensions CM Global Stage Lifecycle

To make the deployment stages in Serena Release Control easy for users to understand, it is recommended that you match the Global Stage Lifecycle (GSL) stages in the Dimensions CM process model to the Serena Release Control stages for major, minor, and emergency release types. You can also configure Serena Release Control stages to match the GSL stages if you have already established GSL stages that users are familiar with.

You must define the deployment areas and assign them to each project you are using for each stage in the GSL. See "Area Definitions" in the *Dimensions CM Process Modeling User’s Guide.*

Documentation References

- Complete documentation on configuring the GSL in Dimensions CM is in the *Dimensions CM Process Modeling User’s Guide* in the "Lifecycle Management" chapter.
- Complete documentation on configuring the deployment areas in Dimensions CM is in the *Dimensions CM Process Modeling User’s Guide* in the "Area Definitions" chapter.

Managing Dimensions CM Users

Serena Release Control uses the user information that you configure as part of the ongoing use of Dimensions CM.

Typically, Serena Release Control users do not need to log into the Dimensions CM client. The interaction most Serena Release Control users have with Dimensions CM is through system functions, and access to Dimensions CM information and actions requested through those functions are executed through the administrative user through which Serena Release Control communicates with Dimensions CM.

Information passed from Dimensions CM to Serena Release Control through the underlying administrative login includes:

- Development Change Request provider: Dimensions CM projects, streams, and requests (optional)
- Deployment Unit provider: Dimensions CM projects, streams, and baselines
- Vault Deployment Tasks: Dimensions CM projects, streams, baselines, and deployment areas

There are two main types of Dimensions CM users that interact with Serena Release Control:

- An administrative user that is specified in the system configuration files and that Serena Release Control uses to sign on to Dimensions CM through SBM and Web services. This user must be set up for single sign-on and have privileges for the following in Dimensions CM:
  - Access to all project and stream information for applications to be deployed from Serena Release Control through Dimensions CM.
  - Access to all request information for applications that will be tracked from Serena Release Control through Dimensions CM as the Development Change Request provider.
  - Access baseline information for applications that will be deployed from Serena Release Control through Dimensions CM as the Deployment Unit provider.
  - Deploy baselines for applications that will be deployed from Serena Release Control through Dimensions CM.
- Any users that have roles in both Dimensions CM and Serena Release Control, such as Serena Release Control power users or Serena Release Manager administrators.

What Can You Change?

- You can change user and role information in Dimensions CM as needed for the users' roles in Dimensions CM.
- Use caution when changing the administrative user that is used to sign on from Serena Release Control and execute the underlying integrative functions in Dimensions CM. You must use single sign-on for the communication between Serena Release Control and Dimensions CM to work.

What is the Impact?

- When a user is signed through single sign-on to either SBM or Dimensions CM and accesses the other client through the Web interface, that user is automatically logged into the other product.

- If you don’t use single sign-on for the administrative user used for the Serena Release Control communication to Dimensions CM, the Serena Release Manager integration and the access to Dimensions CM from Serena Release Control will not work as designed.

How Do You Change It?

Dimensions CM administrators should configure user and role information in Dimensions CM according to the Dimensions CM documentation.

Documentation References

- Complete documentation on configuring users in Dimensions CM is in the *Dimensions CM Process Modeling User’s Guide* in the "Users and Roles" chapter.

Configuring Dimensions CM Projects and Streams

Serena Release Control uses the projects and streams that you configure as part of the ongoing use of Dimensions CM.

Serena Release Control uses projects in integrating products to filter lists of requests or issues to associate with Development Change Requests and to filter lists of baselines or other deployment-ready components to associate with Deployment Units. The default implementation of Serena Release Control integrates with SBM and Dimensions CM projects.

Dimensions CM project association with Release Packages is used for the following purposes in Serena Release Control:

- To filter the development change requests available for creating the association between release packages and development change requests.
- To filter the deployment units, or baselines, available for creating the association between release packages and deployment units.

What Can You Change?

You can change the following project and stream information:

- Add projects and streams.
- Delete projects and streams.
- Modify projects and streams.
- Associate components to projects and streams.
- Associate projects and streams to Dimensions CM requests.
- Create baselines from projects and streams.

What is the Impact?

- The project and stream names appear in the project selection table in the Release Package dialog box. If you change the names or add names, this impacts the list of names the users see when they select projects or streams for a release package.
- The associations with the projects and streams affect the record of change requests and the set of components to be deployed, so changes to these affect release package DCRs and DUs.

How Do You Change It?

You can change project and stream information in the Dimensions CM user clients according to the Dimensions CM documentation.

Documentation References

Complete documentation on configuring Dimensions CM projects and streams is in the *Serena Dimensions CM User's Guide* in "Managing Projects".

Considerations for Requests and Baselines

If Dimensions CM is used as your development change request provider, requests are displayed for selection in Serena Release Control only if they are in one of the statuses defined by your Serena Release Manager administrator.

If Dimensions CM is used as your deployment unit provider, baselines are displayed for selection in Serena Release Control only if they are in release mode and are in one of the statuses defined by your Serena Release Manager administrator.

For more information, see Chapter 3, "Configuring Access to Development Change Requests and Deployment Units" on page 44.

Configuring Entities in Serena Release Automation

Before you begin using the Serena Release Control features that integrate with Serena Release Automation, you must configure the Serena Release Automation entities needed.

The following sections explain the configuration and administration needed for Serena Release Control:

- "Configuring Nolio Users" on page 60
- "Configuring Nolio Processes and Servers" on page 61

Configuring Nolio Users

Typically, Serena Release Control users do not need to log into the Nolio client. The interaction most Serena Release Control users have with Nolio is through system functions, and access to Nolio information and actions requested through those functions are executed through the administrative user through which Serena Release Control communicates with Nolio.

There are two main types of Nolio users that interact with Serena Release Control:

- An administrative user that is specified in the system configuration files and that Serena Release Control uses to sign on to Nolio through Web services and remote database access. This user must have privileges for the following in Nolio:
  - Accessing applications in Serena Release Automation
  - Accessing environments in Serena Release Automation
  - Accessing processes in Serena Release Automation

    - Accessing servers in Serena Release Automation

- Any users that have roles in both Nolio and Serena Release Control, such as Serena Release Control power users or Serena Release Manager administrators.

What Can You Change?

- You can change user and role information in Nolio as needed for the users' roles in Nolio.
- Use caution when changing the administrative user that is used to sign on from Serena Release Control and execute the underlying integrative functions in Serena Release Automation. The Nolio MySQL database must be open for remote access for the communication between Serena Release Control and Nolio to work.

What is the Impact?

- When you create an automation deployment task in Serena Release Control, the super user credentials specified in the configuration files is used to sign on to Serena Release Automation.
- If the super user used to sign on to Serena Release Automation does not have proper privileges or cannot access the MySQL database remotely, the automation deployment tasks will fail.

How Do You Change It?

Nolio administrators should configure users in Nolio according to the Nolio documentation.

Documentation References

- Complete documentation on configuring users in Serena Release Automation is in the *Nolio Automation Center Installation and Administration Guide*.

Configuring Nolio Processes and Servers

Serena Release Manager and Serena Release Control rely on the information that you configure as part of the ongoing use of Nolio.

If you use Serena Release Automation to install and configure deployment units on the servers, or environments, to which you deploy, you must configure the required information in Serena Release Automation. Information required from Nolio includes:

- Applications
- Environments
- Processes
- Servers

Serena Release Control uses the processes and servers that you configure as part of the ongoing use of Nolio. Serena Release Control filters the processes based on your selection of environment, application, and combination of process and server.

Nolio process and server association with Release Packages is used for the following purposes in Serena Release Control:

- To automate the initiation of processes defined in Nolio to complete the installation and configuration of files on designated servers after the deployment units are deployed by Dimensions CM.

What Can You Change?

Nolio administrators can change entities in Nolio according to the Nolio documentation.

What is the Impact?

- When you create an automation deployment task in Serena Release Control, you must select from the Serena Release Automation applications, environments, processes, and servers that you have predefined in Serena Release Automation.
- When a release package is deployed, the automation deployment task is initiated, and the processes you have selected for that task are executed according to the configuration in Serena Release Automation for that process and server.

How Do You Change It?

Nolio administrators should configure application, environment, process, and server information according to the Nolio documentation.

Documentation References

- Complete documentation on configuring entities in Serena Release Automation is in the *Nolio Automation Center Installation and Administration Guide*.

Configuring Serena Business Manager Projects

Serena Release Control uses the projects that your SBM administrators configure as part of the ongoing use of SBM. SBM project association with Release Packages is used for the following purpose in Serena Release Control:

- To filter the development change requests available for creating the association between release packages and development change requests.

What Can You Change?

You can change the following project information:

- Add projects.
- Delete projects, with caution.
- Modify projects.
- Submit items into projects, such as issues and incidents.

What is the Impact?

- The project names appear in the project selection table in the Release Package dialog box. If you change the names or add names, this impacts the list of names the users see when they select projects for a release package.
- The associations with the projects affect the record of SBM items, such as issues or incidents, so changes to these affect release package DCRs.

How Do You Change It?

SBM administrators should change project information in the SBM System Administrator and submit items into projects in the SBM user workspace according to the SBM documentation.

Documentation References

- Complete documentation on configuring SBM projects is in the *Serena Business Manager System Administrator Guide* in "Project Configuration".
- Complete documentation on submitting items into SBM projects is in the *Serena Business Manager System User's Guide* in "Working with Primary Items".

Chapter 5

Serena Release Control Upgrade

This chapter leads you through an upgrade of Serena Release Manager.

Upgrading from Serena Release Control v3.0.0.01 to v3.1

If you need to upgrade Serena Release Control from version 3.0 to version 3.1, please see the Readme for Serena Release Manager version 1.0.0.01 for recommended steps.

You can upgrade Serena Release Control from version 3.0.0.01 to version 3.1 using the following steps.

NOTE Path names in these instructions are examples. The names on your system may be slightly different, in particular for drive, operating system level and 32 or 64-bit choices, and build numbers.

Step	Description	Actions
1	Back up files in target directories.	(Optional) It is a good practice to snapshot your system or back up existing files and data before beginning the upgrade. At minimum, you should back up as follows: **1** Copy the Serena Common Web services `rlm` folder, for example `C:\Program Files\Serena\common\tomcat\6.0\webapps\rlm`, to a temporary folder. **2** Copy the `Serena\Solutions\Release Control` folder, for example `C:\Program Files\Serena\Solutions\Release Control`, to a temporary folder. **3** Copy the following files from the Serena Common Web services `rlm\WEB-INF\classes` folder, for example `C:\Program Files\Serena\common\tomcat\6.0\webapps\rlm`, to a temporary folder. ■ `alf.properties` ■ `dimensions.properties` ■ `nolio.properties`
2	Copy the upgrade package to the Release Control directory.	**1** Delete the contents of the `Serena\Solutions\Release Control` folder. For example: `C:\Program Files\Serena\Solutions\Release Control` **2** Copy the upgrade package, for example `RC-3.1.0-bbb-Win32-Upgrade.zip`, to a temporary directory. **3** Unzip the upgrade package to the `Serena\Solutions\Release Control` folder. **4** The following files should now appear under the `Release Control` folder. ■ `rlm_solution_pack_1_1_0_bbb.sln`, where `bbb` is build number ■ `rlm.war` ■ `com_serena_rlm_sbm_shell.zip`

Step	Description	Actions
3	Extract the Web services files (war file) into the Serena Common Tomcat Web server.	**1** Select **Start | Administrative Tools | Services** and stop the Serena Common Tomcat service. **2** Navigate to the Serena Common Tomcat `webapps` folder. For example: `C:\Program Files\Serena\common\tomcat\6.0\webapps` **3** Delete the `rlm` folder. **4** From the `Release Control` folder where you copied it in a preceding step, copy the `rlm.war` file to the Serena Common Tomcat `webapps` directory. For example: `C:\Program Files\Serena\common\tomcat\6.0\webapps` **5** Restart the Serena Common Tomcat service. This war file contents are extracted to a new `rlm` directory in that location.
4	Install the shell user interface and report templates.	**1** Extract the `com_serena_rlm_sbm_shell.zip` file directly to the SBM Application Engine folder. For example: `C:\Program Files\Serena\SBM\Application Engine\` The files in the `zip` file should extract to the appropriate directory structure. Verify the extraction by looking at the dates of the files in the `template\shell\rlm folder`, for example `C:\Program Files\Serena\SBM\Application Engine\template\shell\rlm`. The files should have the date close to that of the upgrade package you used. **2** From the SBM System Administrator, select **File | Put Files Into Database**. Confirm when prompted. This puts any necessary metadata into the SBM database.
5	Install the solution file that contains the Serena Release Control process apps and all related orchestrations, reports, and tables.	**1** From the `Release Control` folder, copy the solution pack `.sln` file, such as `rlm_solution_pack_1_1.sln`, to the `SBM WEB-INF\solutions` folder. For example: `C:\Program Files\Serena\SBM\Common\jboss405\server\default\deploy\mashupmgr.war\WEB-INF\solutions`
6	Import the Release Control solution.	**1** Log into the SBM Application Administrator as the SBM administrative user. **2** Navigate to the **Solutions** tab and import the Serena Release Control solution. For example: `RLM_Solution_Pack_1.1.m.bbb`, where `m` is maintenance release number and `bbb` is build number. **3** See Chapter 3, "Promoting and Configuring Serena Release Control in SBM" on page 26.

Step	Description	Actions
7	Promote the snapshots.	**1** Navigate to **Environments** and select the **Endpoints** tab. **2** Edit the following endpoints for your Serena Release Manager environment and change the port number to 9095. ■ RLMUtilService ■ ReleaseRequestService ■ DeployUnitService If you are using a port number other than the default of 9095, see Chapter 6, "Addressing Web Server Port Conflicts" on page 97. **3** Navigate to the **Process App Snapshots** tab. **4** Promote the snapshots in the following recommended order: **a** **RLM_AUX** **b** **ReleaseTemplate** **c** **Application Release** **d** **Release Package** **e** **Deployment** **f** **Release Train** When promoting the snapshots, make sure to select the endpoints as needed, and make sure the endpoints are authenticated with Single Sign-on. **IMPORTANT!** Make sure to check for any warnings after you have promoted the snapshots. If you received warnings for a snapshot, promote that snapshot again to ensure that dependencies between the applications are set up properly. See Chapter 3, "Promoting and Configuring Serena Release Control in SBM" on page 26.

Step	Description	Actions

8	Update the Serena Common Web Server to use the new port number, 9095, instead of 8080.	**1** Select **Start | Administrative Tools | Services** and stop the Serena Common Tomcat service.
		2 Navigate to the Serena Common Tomcat `conf` folder. For example:
		`C:\Program Files\Serena\common\tomcat\6.0\conf`
		3 Edit the server.xml file and change the Connector port line as follows:
		`<Connector port="9095" protocol="HTTP/1.1" ...`
		4 Restart the Serena Common Tomcat service.
		The Tomcat Web server is now running on port 9095.
		5 Verify by trying the Web server URL in your Web browser. For example:
		`http://localhost:9095/`
		If the Web server is running correctly on that port, the The default main web page of the Serena Common Tools appears.
		IMPORTANT! If you are using a port number other than the default of 9095, see Chapter 6, "Addressing Web Server Port Conflicts" on page 97.

9	Download the JDBC driver for your Serena Release Automation database if needed	**1** If you are using Serena Release Automation with Serena Release Control and you are using a database management system other than Oracle, you must download the JDBC driver needed to connect to your database.
		For MySQL, you can download the JDBC driver from here:
		http://www.mysql.com/downloads/connector/j/
		For SQL Server, you can download the JDBC driver from here:
		http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=2505
		2 Extract and place the `.jar` file from the root location into the Serena Common Web server `rlm\WEB-INF\lib` folder on the Serena Release Manager server. For example, copy
		`mysql-connector-java-5.1.18-bin.jar`
		to
		`C:\Program Files\Serena\common\tomcat\6.0\\webapps\rlm\WEB-INF\lib`
		3 Restart the Serena Common Tomcat service.

10	Set privileges for the administrative user to the Serena Release Control objects, such as projects, reports, and tables.	**1** Set privileges for the new deployment process template objects and verify privileges are set for all other Serena Release Control objects. See Chapter 3, "Configuring the Administrative User Privileges" on page 30.

Step	Description	Actions
11	Select a specific user as owner for the Deployment Task sub-projects by selecting a user from the owning role.	**1** Select a specific owner for the new deployment template sub-projects and verify that the owner is selected for all other deployment task sub-projects. See Chapter 3, "Configuring Default Owners" on page 31.
12	Enable roles for Serena Release Control projects and verify that Serena Release Control is activated.	**1** Enable the roles in the new Release Template project and verify that roles are enabled for all other Serena Release Control projects. See Chapter 3, "Enabling Serena Release Control Project Roles" on page 33. **2** Verify that Serena Release Control is activated by entering the URL in your Web browser. For example: `http://rlmhost/tmtrack/tmtrack.dll?shell=rlm`
13	Configure the dashboard if needed.	If your Dashboard page does not appear with the multi-view dashboard report, configure the dashboard as follows. **1** In SBM System Administrator, select the **Users** tab. **2** Select your Login ID and click **Edit**. **3** Select the **Preferences** tab and deselect the **Show Launch Page** field. **4** Select the **Settings** tab. **a** In the **Application** list, select **Release Train** and then click **Set Preferred Application**. **b** In the Home Page Report field, select **Base Project: Dashboard**. **5** Click **OK**.
14	Update the registry to ensure the proper version of Serena Release Control is registered for future reference.	(Optional) **1** Automatically update the key by opening the provided `RC_3.1.0.reg` file. Confirm when prompted to update your registry key. OR Manually update the key as follows: **a** Edit the registry with a program such as `Regedit`. **b** View the following key: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{F711E0DD-CAE6-4ADE-8CAA-8F54BB92214F}` **c** Modify **DisplayVersion** by changing the **Value data** entry to 3.1.0. **NOTE** This registry location has all the information that you will see in **Add/Remove** programs and **Programs and Features**.

Chapter 6

Serena Release Control Customization

This chapter gives an overview of advanced configuration, or customization, in Serena Release Manager.

Serena Release Manager is designed as a flexible, extensible system that you can customize to match the best Release Control, Release Vault, and Release Automation for your organization’s release processes. You can modify key elements of the system to match the way your organization manages releases, such as the user interface, release control processes, and release types and stages.

You may also want to add support for systems other than Serena Release Vault and Serena Business Manager for your Development Change Request and Deployment Unit providers.

NOTE The actual deployment of Deployment Units to stage environments must be done by integrating with Serena Release Vault, because Serena Release Control relies on the process modeling provided by the GSL and Serena Release Vault deployment areas.

CAUTION! Modification of transitions and states in Serena Release Control must be done by personnel who have a strong understanding of SBM orchestrations and SBM Composer. Some of the transitions and states in Serena Release Control are used by the underlying Web services and are referenced by Javascript, and if these are modified without additional system changes, Serena Release Control will no longer function.

We strongly recommend that you contact Serena Services for assistance in customizing your Serena Release Control system.

This chapter includes the following topics.

Customizing the User Interface

Before the people who participate in the release management processes in your organization begin using Serena Release Control, you can configure the user interface to suit your needs.

User interface configuration topics are as follows:

- "Configuring the Dashboard" on page 72
- "Configuring the Calendar" on page 73
- "Configuring the Inbox" on page 74
- "Configuring Views and Dialog Boxes" on page 76

NOTE This section includes typical user interface configuration. This does not include customization of the UI shell to include new form elements. Some UI customization is required when you add or modify stages. This customization is documented in context in Chapter 6, "Serena Release Control Customization" on page 71.

Configuring the Dashboard

The SBM home page is shown on the Serena Release Control dashboard. To change the view that is shown on the dashboard, you can configure the home page report in SBM.

Upon login, the dashboard page is displayed by default. The report last selected for the dashboard is shown.

What Can You Change?

You can change the reports displayed on the dashboard as follows:

- Select a different report to display on the dashboard page.
- Select different graphical report styles to display for each report.

What is the Impact?

If you change the dashboard reports, the dashboard page changes accordingly.

How Do You Change It?

You can select from available reports to display the release information that is most important to you.

To configure and select the Dashboard multi-view report for the dashboard page:

1 From the Serena Release Control standard SBM user interface, click the **RTrain** tab.

2 Expand the **Reports** section of the navigation pane and click **Browse Multi-View Reports**.

3 Select the **Dashboard** report from the list of reports.

4 Click **Edit report** and configure the Dashboard report to meet your needs as documented in the SBM documentation.

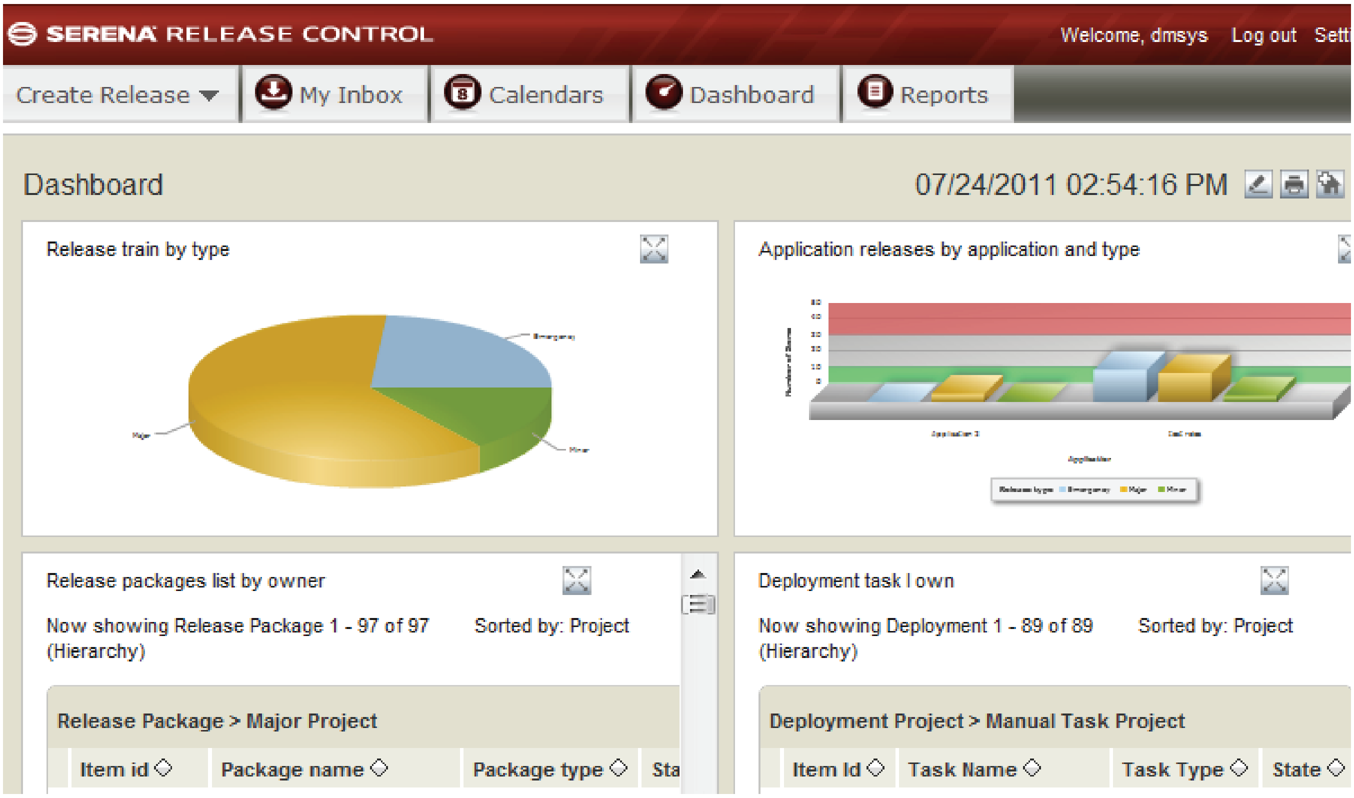
5 Click **Set as Home Page** if the dashboard report is not already set as your home page report.

Documentation References

Complete documentation on managing reports in SBM is in the *Serena Business Manager User's Guide* in "Working with Reports".

Example

The Serena Release Control dashboard with reports for **Release train by type**, **Application releases by application and type**, **Release packages list by owner**, and **Deployment tasks I own** is shown in the following figure.



Release packages list by owner

Now showing Release Package 1 - 97 of 97 (Hierarchy) Sorted by: Project

Release Package > Major Project

Item id ◇	Package name ◇	Package type ◇	Sta

Deployment task I own

Now showing Deployment 1 - 89 of 89 (Hierarchy) Sorted by: Project

Deployment Project > Manual Task Project

Item Id ◇	Task Name ◇	Task Type ◇	State ◇

Configuring the Calendar

Serena Release Control uses a Gantt report to display the information in your calendar view.

What Can You Change?

You can change the reports displayed on the calendar as follows:

- Select a different report to display for the calendar.
- Modify the report to display for the calendar.
- Select a different HTML template for the calendar.

What is the Impact?

If you change the calendar reports, the calendar page changes accordingly.

How Do You Change It?

You can change the associated report to change the information shown on the calendar.

To change the calendar report:

1 From the Serena Release Control standard SBM user interface, click the **RTrain** tab.

2 Select **Reports** from the navigation pane and search for reports with Gantt in the title.

3 Under **Release Train New Project (Release Train)**, select **Release Trains in Gantt** and then click **Modify Listing** for that report.

4 Expand **Additional Options** and look at the **Optional HTML Template** field.

This lists the HTML page for the selected report. The default report is:

```
ganttview.htm
```

5 Modify the report selections and the corresponding HTML page as needed.

Documentation References

Complete documentation on managing reports in SBM is in the *Serena Business Manager User's Guide* in "Working with Reports".

Configuring the Inbox

Serena Release Control uses reports to display the action item information in your inbox views.

What Can You Change?

You can change the reports displayed in the inbox as follows:

- Change the reports that populate the inbox.
- Save reports with a different name by saving in place and retaining the report ID.

CAUTION! You must change the report name by saving in place and retaining the report ID. Otherwise, the view that uses that report will no longer be displayed without additional changes to the underlying shell UI Javascript.

What is the Impact?

- If you change the inbox reports, the list of items in the **My Inbox** page changes according to the new report criteria.
- Most of these reports are used for multiple views. You need to ensure that the changes you make are appropriate for all the places the information is displayed.
- If you change the name of the report incorrectly, the view that uses that report will be displayed incorrectly.

How Do You Change It?

You can change the associated reports to change the information shown in the inbox.

To change reports for the inbox:

1. From the Serena Release Control standard SBM user interface, select the process app for which you want to change a report, such as **RTrain**.
2. From the **Reports** section of the navigation pane, search for all reports.
3. Select the report you want to change and then click **Save**.
4. Overtype the existing name with a new name, making sure that you do not change the report ID.
5. Repeat for all Serena Release Control process apps for which you want to change **My Inbox** page reports.

The default reports for each inbox selection are shown in the following table.

Section	Entity	Report Name
My Action Items	Release Trains	train Assigned To Current User
	Application Releases	application Assigned To Current User
	Release Packages	package Assigned To Current User
	Deployment Tasks	task Assigned To Current User
	Deployment Process Templates	Template Assigned To Current User
Manage All Items	Release Trains	Train All
	Application Releases	Application All
	Release Packages	Package All
	Deployment Tasks	All DT
	Deployment Process Templates	Templates All

TIP To find out the names of the reports used by your **My Inbox** pages, view each page using the Serena Release Control shell UI and view the source for the page. You'll see the name of the report used for the page in the Javascript. For example, for the **Manage All Items | Deployment Process Template**, the Javascript snippet is as follows:

```
var report = {"request":{"id":123,"title":"Templates All","rptkey":1319845733, ...
```

Documentation References

Complete documentation on managing reports in SBM is in the *Serena Business Manager User's Guide* in "Working with Reports".

Configuring Views and Dialog Boxes

Serena Release Control is designed so that most unique user interface needs can be met by configuring the underlying SBM entities. Reports and auxiliary table items populate much of the interface and the workflow transitions determine the action buttons used to progress the release items through their lifecycles. Flexible widgets also provide a way to quickly change UI information.

What Can You Change?

You can change the views and dialog boxes as follows:

- Change the values provided in selection lists and search fields.
- Edit the reports used to display information in tables.
- Show and hide columns in tables.
- Save reports with a different name by saving in place and retaining the report ID.

⚠ **CAUTION!** You must change the report name by saving in place and retaining the report ID. Otherwise, the view that uses that report will no longer be displayed without additional changes to the underlying shell UI Javascript.

What is the Impact?

Possible impacts for changing dialog boxes are as follows:

- If you change reports for the views and dialog boxes, you must make sure the columns display properly in the tables, which by default use the `flexigrid` table widget.
- Most of these reports are used for multiple views. You need to ensure that the changes you make are appropriate for all the places the information is displayed.
- If you change the name of the report incorrectly, the view that uses that report will be displayed incorrectly.

How Do You Change It?

To change information displayed in a selection field:

1. From the Serena Release Control standard SBM user interface navigation pane, click the **Search** link, and then click **Manage Data**.
2. Update the table for the selection field with the entries you need.

To change information displayed in a table:

1. From the Serena Release Control standard SBM user interface, view and change the report that populates the table information you want to change.

   The default implementation of Serena Release Control reports provided for views and dialog boxes are listed in the following table.

View	Report
Release Train	train Assigned To Current User
Application Release	application Assigned To Current User

View	Report
Release Package	package Assigned To Current User
	Release packages list by owner
Deployment Task	task Assigned To Current User
Deployment Process Template	Template Assigned To Current User

Table formatting is controlled by a grid widget. The table is populated by an SBM report.

2 Verify that the updated report information shows properly in the associated UI.

3 To show or hide columns of information returned from the report, click the down-arrow beside the last column heading and then select or deselect column names in the list.

Documentation References

- Complete documentation on configuring tables in SBM is in the *Serena Business Manager System Administrator Guide* in "Table Configuration".
- Complete documentation on managing reports in SBM is in the *Serena Business Manager User’s Guide* in "Working with Reports".

Customizing Release Control Workflows

Before the people who participate in the release management processes in your organization begin using Serena Release Control, you can customize the workflow states and transitions in SBM that support your organization's release control processes.

Serena Release Control workflow states are defined in the underlying Serena Business Manager system. If your organization uses different states within the release management workflow, you can change the Serena Release Control workflow accordingly.

What Can You Change?

- You can change the following workflows:
  - Release Train
  - Application Release
  - Manual Deployment Task
  - Vault Deployment Task
  - Automation Deployment Task
- You can change the following workflow state and transition information:
  - Modify state and transition names
  - Modify state ownership
  - Add states and transitions
  - Remove states and transitions

IMPORTANT! The Release Package workflow contains logic that is integrated with the Release Type and Stage implementation. Information on changing the Release Package workflow is included in context in "Modifying Release Types and Stages" on page 79.

What is the Impact?

When you change the workflow, the following impacts are made.

- If you add states, you must add owners and transitions.
- If you add transitions, you must add any functionality required to support the associated action and change the form if needed to support this.
- If you change state names, you may need to change the associated entries in the auxiliary table.
- If you change transition names, different actions appear on the related UI for progressing the items through their workflows. You may need to change the related Javascript for the UI shell to display the new name properly.
- Ownership relates to SBM projects, so you must add ownership for any new roles for each project and workflow.

How Do You Change It?

Change the workflow and related entities in SBM Composer according to the SBM documentation.

Change the auxiliary table entries in the SBM System Administrator according to the SBM documentation.

Documentation References

- Complete documentation on configuring workflows in SBM is in the *Serena Business Manager SBM Composer Guide* in "Managing Workflows".
- Complete documentation on configuring tables in SBM is in the *Serena Business Manager System Administrator Guide* in "Table Configuration".

Modifying Release Types and Stages

Serena Business Manager release types are used to determine the stages, or environments, that release packages move through on their path into production.

The default release types are major, minor, and emergency, and default stages include Integration Test, User Acceptance Testing (UAT), and Production Deployment.

Stages are implemented in the Release Train and the Release Package, so any changes to one must be made in the other.

What Can You Change?

You can change the following release type and stage information:

- Modify release types
- Add release types
- Modify stages
- Add stages
- Delete release types
- Delete stages

What is the Impact?

- The release types and stages are interrelated, and if you change one aspect, you need to change all related information in both release train and release package primary and auxiliary tables, forms, workflows, and UI Javascript.
- You must be careful when changing the Release Package system field definitions because they are used by Serena Release Control to deploy release packages. These include the following:
  - **Package type**: Controls the release package staging process sequence.
  - **Deploy state**: Controls the re-deployment process.
  - **Next Deploy Transition**: The update transition name to use when a release package is successfully deployed. The default Release package workflow value is set to Deployed.
  - **Failed Deploy Transition**: The update transition name to use when there is failure in the release package deployment process. The default Release package workflow value is set to Fail Deployment.

- You must be careful when changing the Release Package process app not to adversely affect the implementation of the Deployment Task execution. Deployment Task field dependencies are as follows:
    - Deployment Task Status Single Selection control field:

      The Task status (TASK_STATUS) single selection field controls when a deployment task can be executed for deployment, when to execute the task deployment process, and when to execute the fail deployment process.
    - Default settings:

      To indicate when a deployment task can be executed for deployment, the task status value must be set to Planned. This is currently set in the Create transition.
    - The automation transition actions associated to the Task Update transition are mapped to the value set in the Task Status. Currently the default actions are set as follows:

Transition Action	Task Status
Invoke Execute Deployment transition of Manual, Automation, and Vault Task	In Progress
Invoke Fail Deployment transition of Automation and Vault Task	Failed
Invoke Complete Deployment transition of Automation and Vault Task	Completed

How Do You Change It?

You add or change most stage information in SBM Composer.

The summary and example given here are for adding a stage. From this you should also gain the knowledge you need to modify or delete a release type or stage.

Before you add a stage, it is recommended that you look at the implementation of an existing stage, such as INT, as the new stage should be implemented in a very similar manner.

Summary of Adding a Stage

Adding a Stage for the Release Train Process App

1. Add new Start and End Date fields in the primary table Release Train.
2. Add the field controls in the related form, **createReleaseTrain**.
3. Edit the related Javascript for the UI shell to specify when to show or hide the dates in the following forms:
    - createReleaseTrain
    - viewReleaseTrain
4. Include the new Start and End Date fields in the report All Release Train.

Adding the Stage for the Release Package Process App

1. Add the new stage name value in the primary table Release package.
2. Create a new swim lane for the new stage process.
3. Create required states for the new stage process.
4. Add From and To transitions with all associated forms, mappings, and overrides for the new states.
5. Map a new deployment transition action.
6. Add a new re-deploy transition action.
7. Configure the Stage auxiliary table to relate the new stage to the Release Train Start and End Dates and the Release Package Deploy state.
8. Optionally add a new deploy decision rule to the Release Package stage process.
   - **a** Add the decision to the workflow.
   - **b** Add a new **Package type** single-selection field value.
   - **c** Add a rule for the selection, **Package type** in value.
   - **d** Add a new transition for the deploy decision.
   - **e** Add the new rule to the deploy decision box.

Example of Adding a Stage

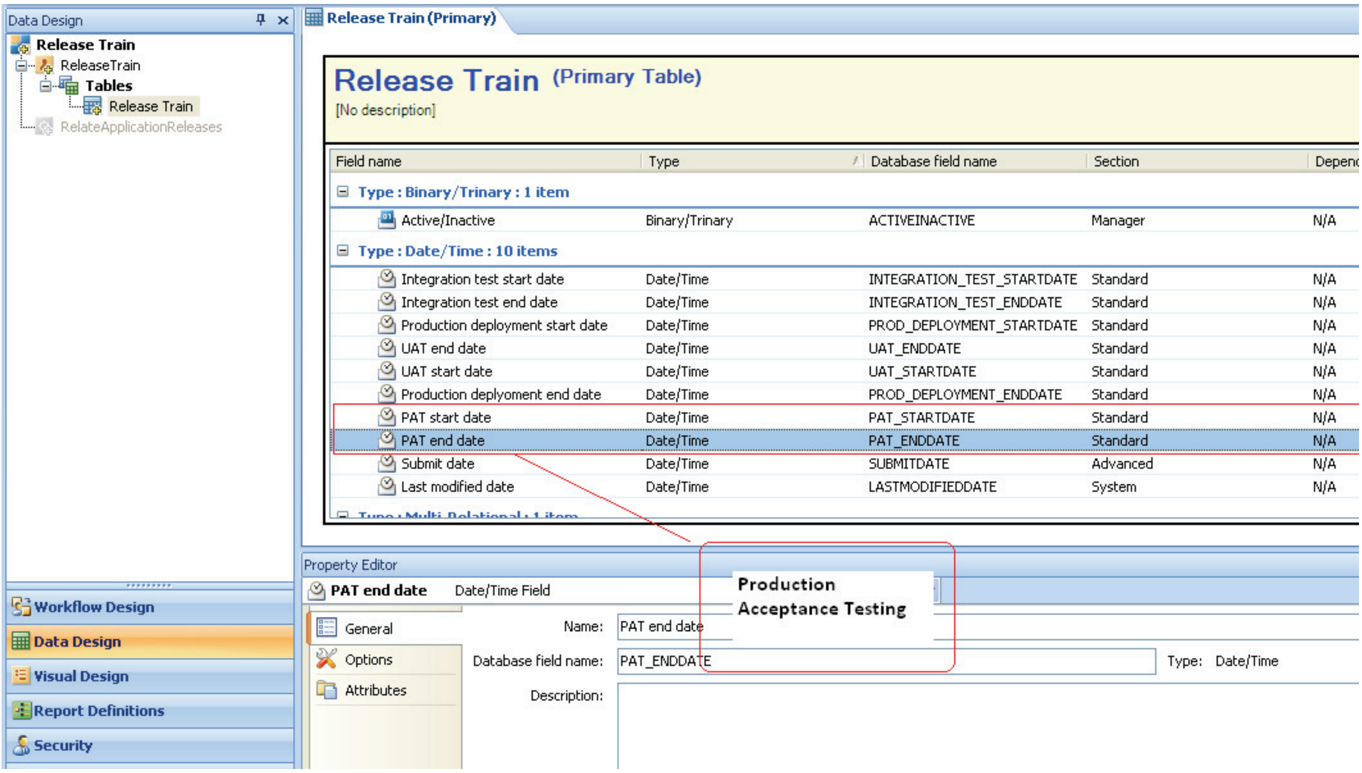
This example shows how to add a new stage that's associated with a new Release Train Start and End Date and a new Release Package stage process. This example adds a stage of PAT, PAtch Test. This stage is defined to be used only when the release type is Patch.

Adding a Stage for the Release Train Process App

To add a stage for the Release Train process app:

1. Add new Start and End Date fields in the primary table Release Train.

In the Release Train table add PAT start and end dates, as shown in the following figure.



Release Train (Primary Table)

[No description]

Field name	Type	Database field name	Section	Depend
Type : Binary/Trinary : 1 item				
Active/Inactive	Binary/Trinary	ACTIVEINACTIVE	Manager	N/A
Type : Date/Time : 10 items				
Integration test start date	Date/Time	INTEGRATION_TEST_STARTDATE	Standard	N/A
Integration test end date	Date/Time	INTEGRATION_TEST_ENDDATE	Standard	N/A
Production deployment start date	Date/Time	PROD_DEPLOYMENT_STARTDATE	Standard	N/A
UAT end date	Date/Time	UAT_ENDDATE	Standard	N/A
UAT start date	Date/Time	UAT_STARTDATE	Standard	N/A
Production deplyoment end date	Date/Time	PROD_DEPLOYMENT_ENDDATE	Standard	N/A
PAT start date	Date/Time	PAT_STARTDATE	Standard	N/A
PAT end date	Date/Time	PAT_ENDDATE	Standard	N/A
Submit date	Date/Time	SUBMITDATE	Advanced	N/A
Last modified date	Date/Time	LASTMODIFIEDDATE	System	N/A

Property Editor

PAT end date Date/Time Field

Name: PAT end date

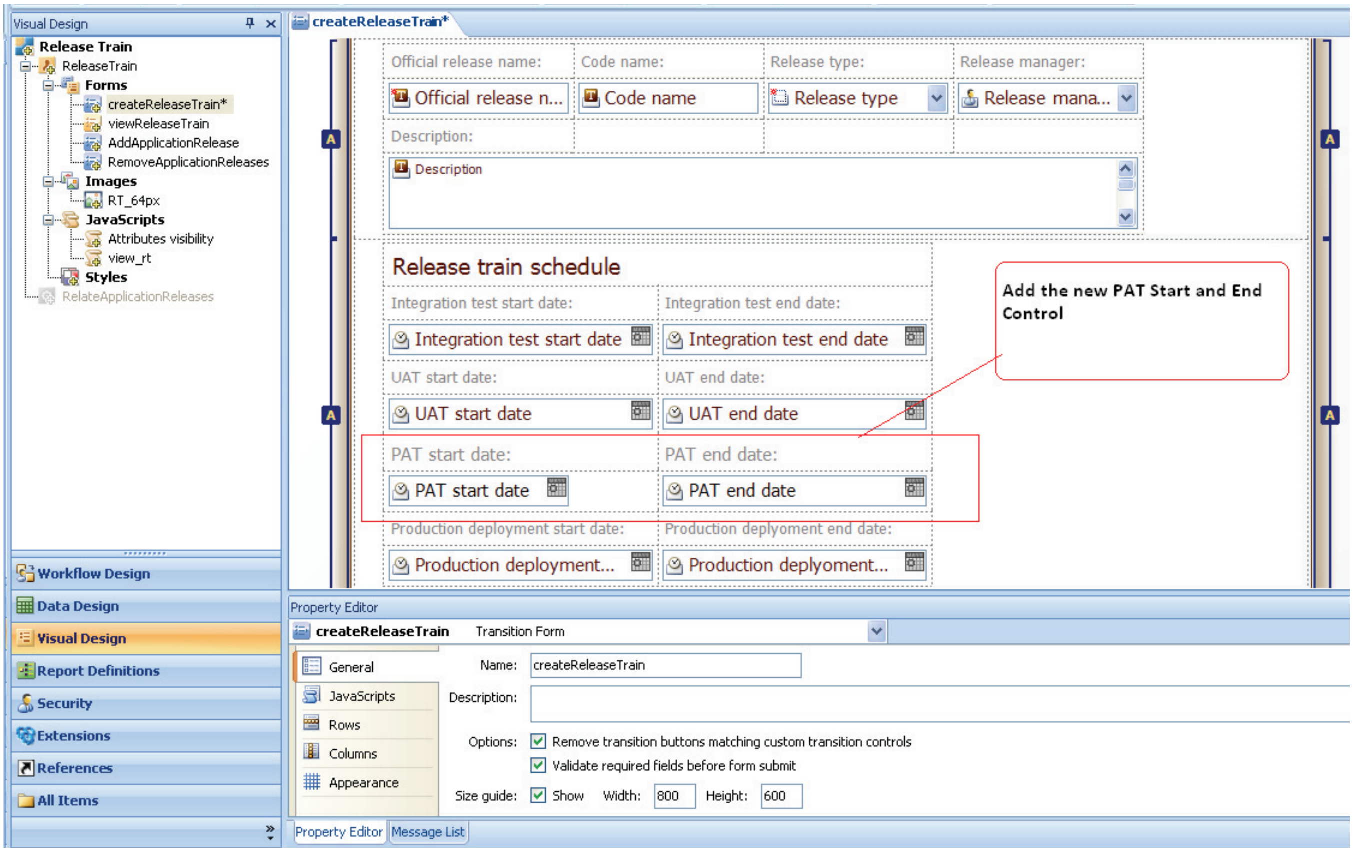
Database field name: PAT_ENDDATE

Type: Date/Time

Description:

Production Acceptance Testing

2 Add the Start and End field controls in the related form, **createReleaseTrain**, as shown in the following figure.



Add the new PAT Start and End Control

Property Editor

createReleaseTrain Transition Form

Name: createReleaseTrain

Description:

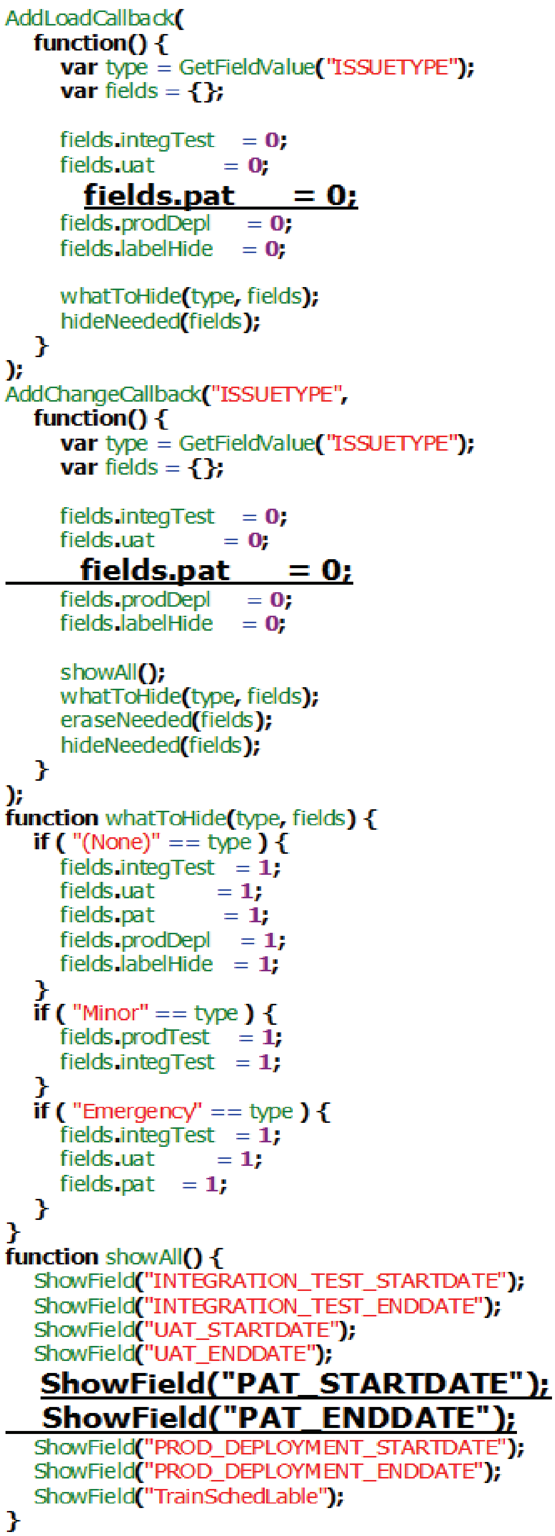
Options: ☑ Remove transition buttons matching custom transition controls
☑ Validate required fields before form submit

Size guide: ☑ Show Width: 800 Height: 600

3 Edit the related Javascript for the UI shell, Attributes visibility, to specify when to show or hide the dates in the following forms:

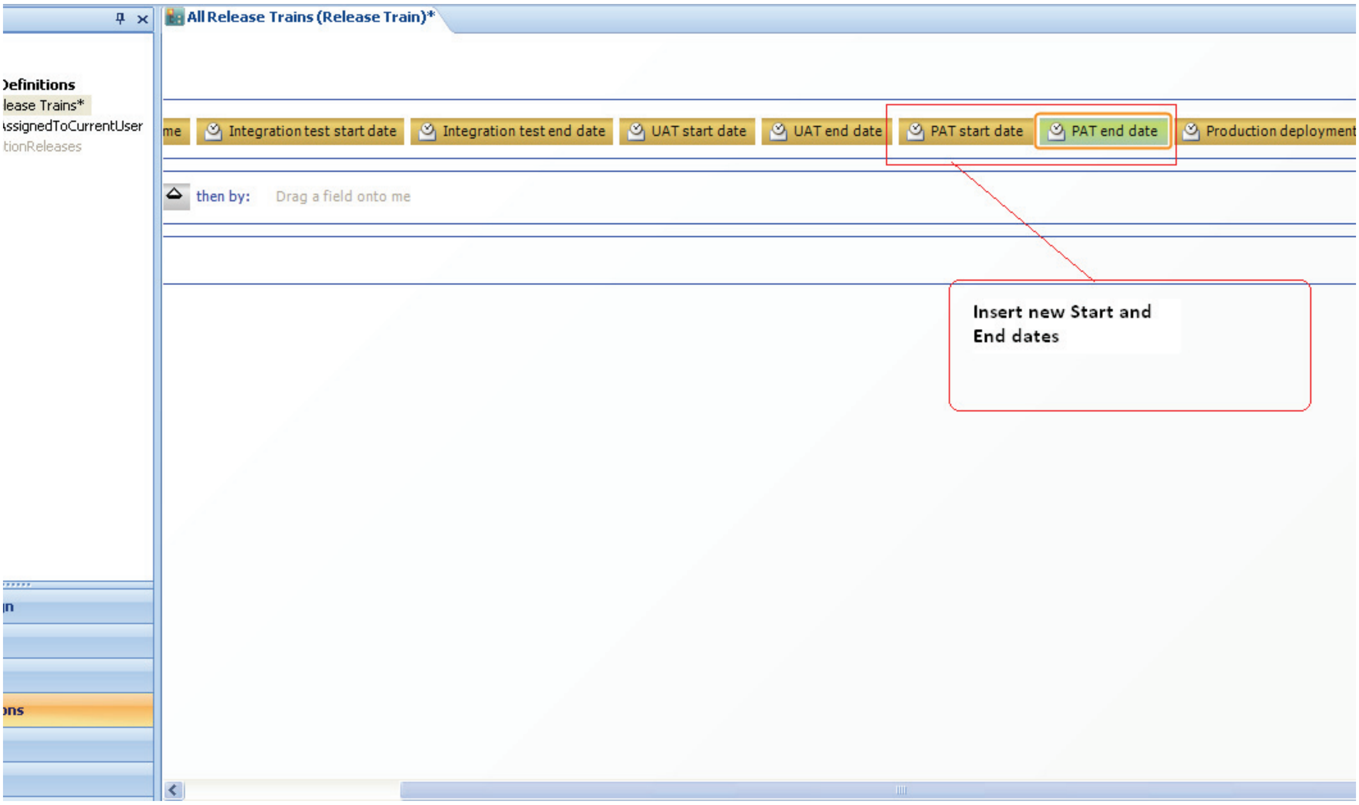
- createReleaseTrain
- viewReleaseTrain

The dates are shown or hidden based on release train type (**ISSUETYPE** table field). The lines to change in the Javascript file are shown in bold in the following figure.



```
AddLoadCallback(
   function() {
      var type = GetFieldValue("ISSUETYPE");
      var fields = {};

      fields.integTest    = 0;
      fields.uat          = 0;
        fields.pat      = 0;
      fields.prodDepl     = 0;
      fields.labelHide    = 0;

      whatToHide(type, fields);
      hideNeeded(fields);
   }
);
AddChangeCallback("ISSUETYPE",
   function() {
      var type = GetFieldValue("ISSUETYPE");
      var fields = {};

      fields.integTest    = 0;
      fields.uat          = 0;
        fields.pat      = 0;
      fields.prodDepl     = 0;
      fields.labelHide    = 0;

      showAll();
      whatToHide(type, fields);
      eraseNeeded(fields);
      hideNeeded(fields);
   }
);
function whatToHide(type, fields) {
   if ( "(None)" == type ) {
      fields.integTest   = 1;
      fields.uat         = 1;
      fields.pat          = 1;
      fields.prodDepl     = 1;
      fields.labelHide   = 1;
   }
   if ( "Minor" == type ) {
      fields.prodTest    = 1;
      fields.integTest   = 1;
   }
   if ( "Emergency" == type ) {
      fields.integTest   = 1;
      fields.uat         = 1;
      fields.pat    = 1;
   }
}
function showAll() {
   ShowField("INTEGRATION_TEST_STARTDATE");
   ShowField("INTEGRATION_TEST_ENDDATE");
   ShowField("UAT_STARTDATE");
   ShowField("UAT_ENDDATE");
    ShowField("PAT_STARTDATE");
    ShowField("PAT_ENDDATE");
   ShowField("PROD_DEPLOYMENT_STARTDATE");
   ShowField("PROD_DEPLOYMENT_ENDDATE");
   ShowField("TrainSchedLable");
}
```

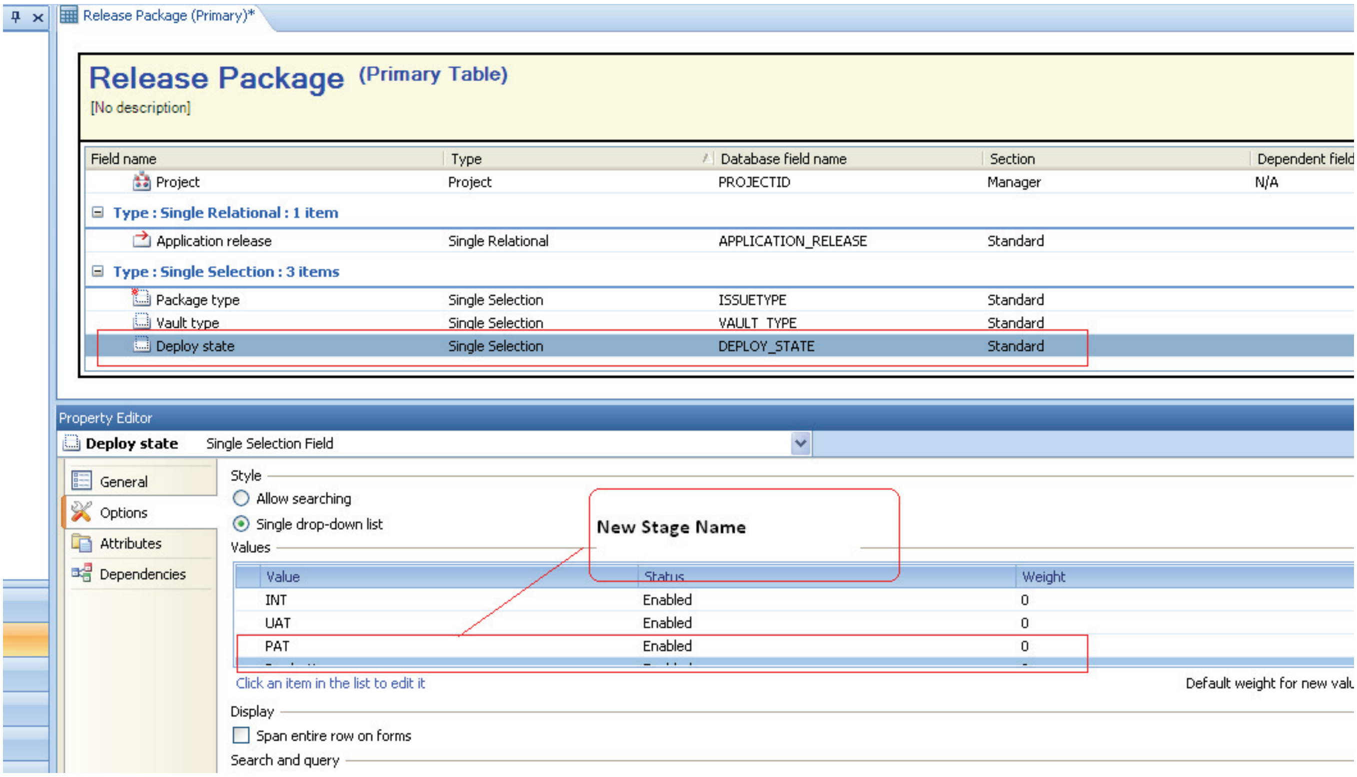
4 Include the new Start and End Date fields in the report **All Release Train** as shown in the following figure.



Adding the Stage for the Release Package Process App

To add the stage for the Release Package process app:

1 Add the new stage name value in the primary table Release package, in the **DB DEPLOY_STATE** single selection field as shown in the following figure.



Release Package (Primary Table)

[No description]

Field name	Type	Database field name	Section	Dependent field
Project	Project	PROJECTID	Manager	N/A
Type : Single Relational : 1 item				
Application release	Single Relational	APPLICATION_RELEASE	Standard	
Type : Single Selection : 3 items				
Package type	Single Selection	ISSUETYPE	Standard	
Vault type	Single Selection	VAULT_TYPE	Standard	
Deploy state	Single Selection	DEPLOY_STATE	Standard	

Deploy state Single Selection Field

Style

○ Allow searching

● Single drop-down list

Values

Value	Status	Weight
INT	Enabled	0
UAT	Enabled	0
PAT	Enabled	0

Click an item in the list to edit it

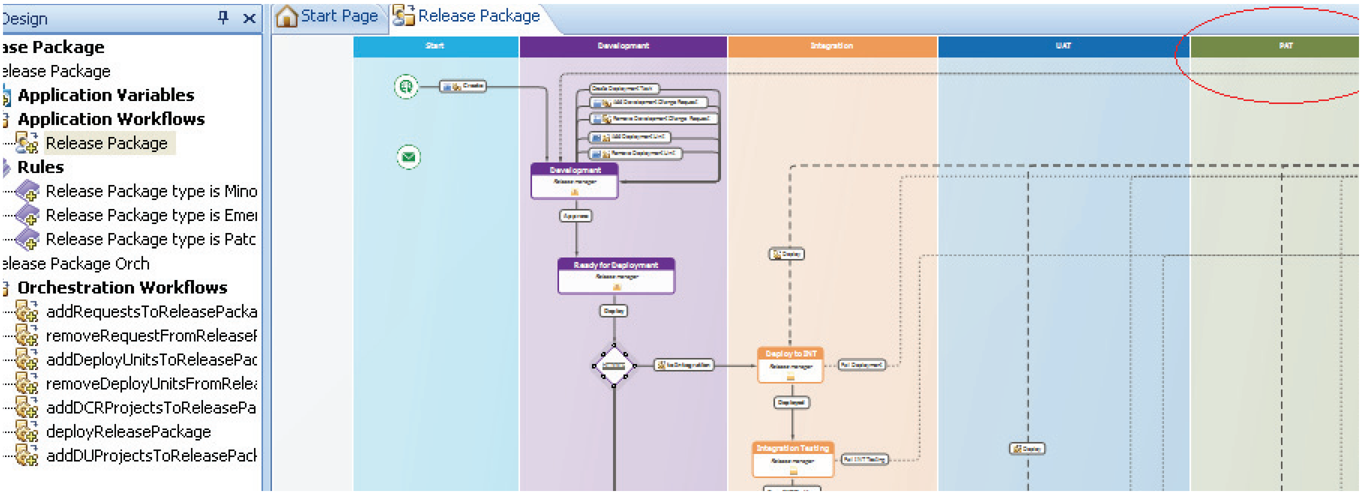
Default weight for new valu

Display

☐ Span entire row on forms

Search and query

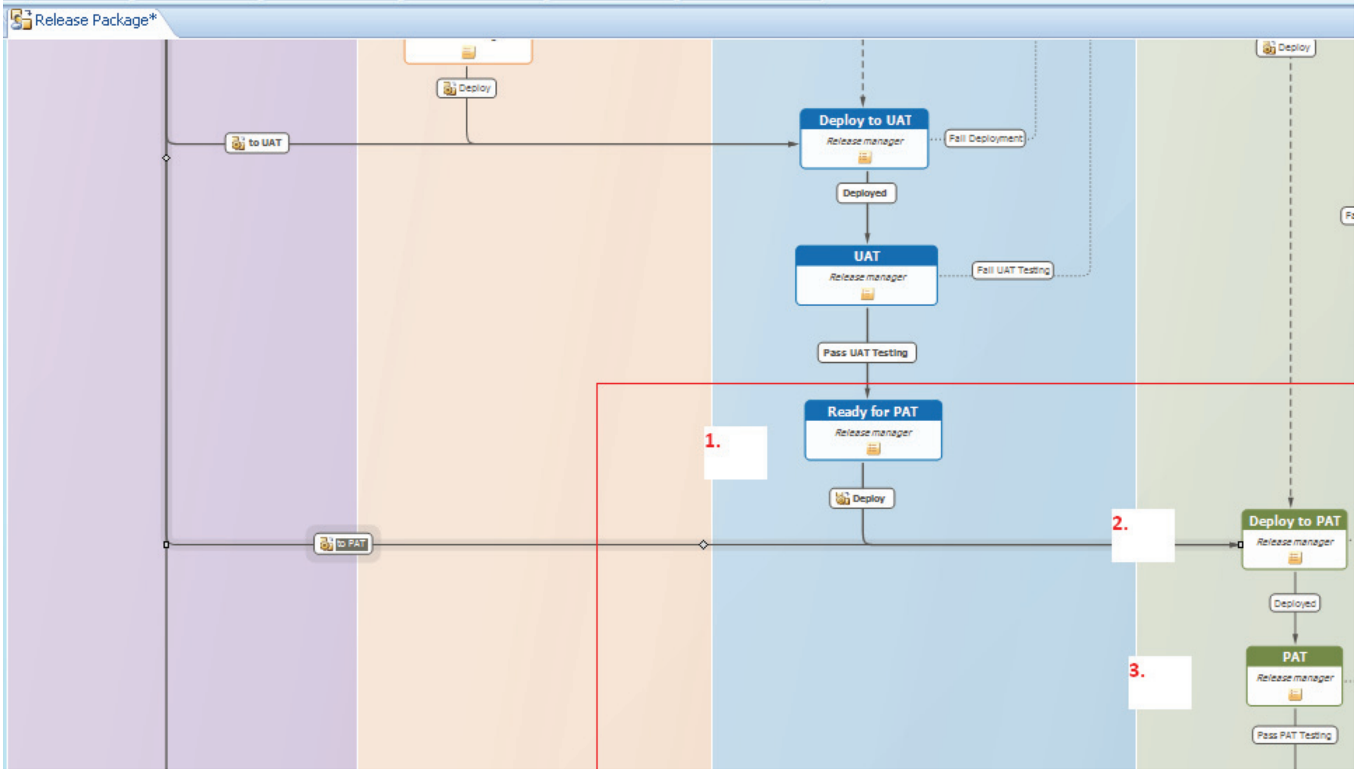
2 Create a new swim lane for the new stage process and arrange it according to the Release Package staging process sequence as shown in the following figure.



3 Create required states for the new stage process. For this example, those are:

- Ready for PAT
- Deploy to PAT
- PAT

The added states are shown in the following figure.



4 Add From and To transitions with all associated forms, mappings, and overrides for the new states as shown in the following table.

State	Transition	Option	Selection
Ready for PAT	From Transition: Pass UAT Testing		
	To Transition: Deploy	Options	Quick transition
		Form	None
		Field Privileges	default values
		Field Overrides	Failed Deploy Transition ■ Read Only ■ Set to default: Fail Deployment
		Actions	Invoke deployReleasePackage Orchestration workflow
		Restrict by Type	default values
		Restrict by Role	default values

State	Transition	Option	Selection
Deploy to PAT	From transition: Deploy		
	To transition: Deployed	Options	Quick transition
		Form	None
		Field Privileges	default values
		Field Overrides	None
		Actions	None
		Restrict by Type	default values
		Restrict by Role	default values
	To transition: Fail Deployment	Options	Quick transition
		Form	None
		Field Privileges	default values
		Field Overrides	Deploy state ■ Read only ■ Set to default: PAT
		Actions	None
		Restrict by Type	default values
		Restrict by Role	default values
PAT state	From transition: Deployed		
	To transition: Pass PAT Testing	Options	Quick transition
		Form	None
		Field Privileges	default values
		Field Overrides	None
		Actions	None
		Restrict by Type	default values
		Restrict by Role	default values
	To transition: Fail PAT Testing	Options	Quick transition
		Form	None
		Field Privileges	default values
		Field Overrides	None
		Actions	None
		Restrict by Type	default values
		Restrict by Role	default values

5 Map a new deployment transition action based on the **Deploy** state single selection field. To do so, create a new **Deploy** transition from the state **Failed Deployment** to the state **Deploy to PAT** as shown in the following table.

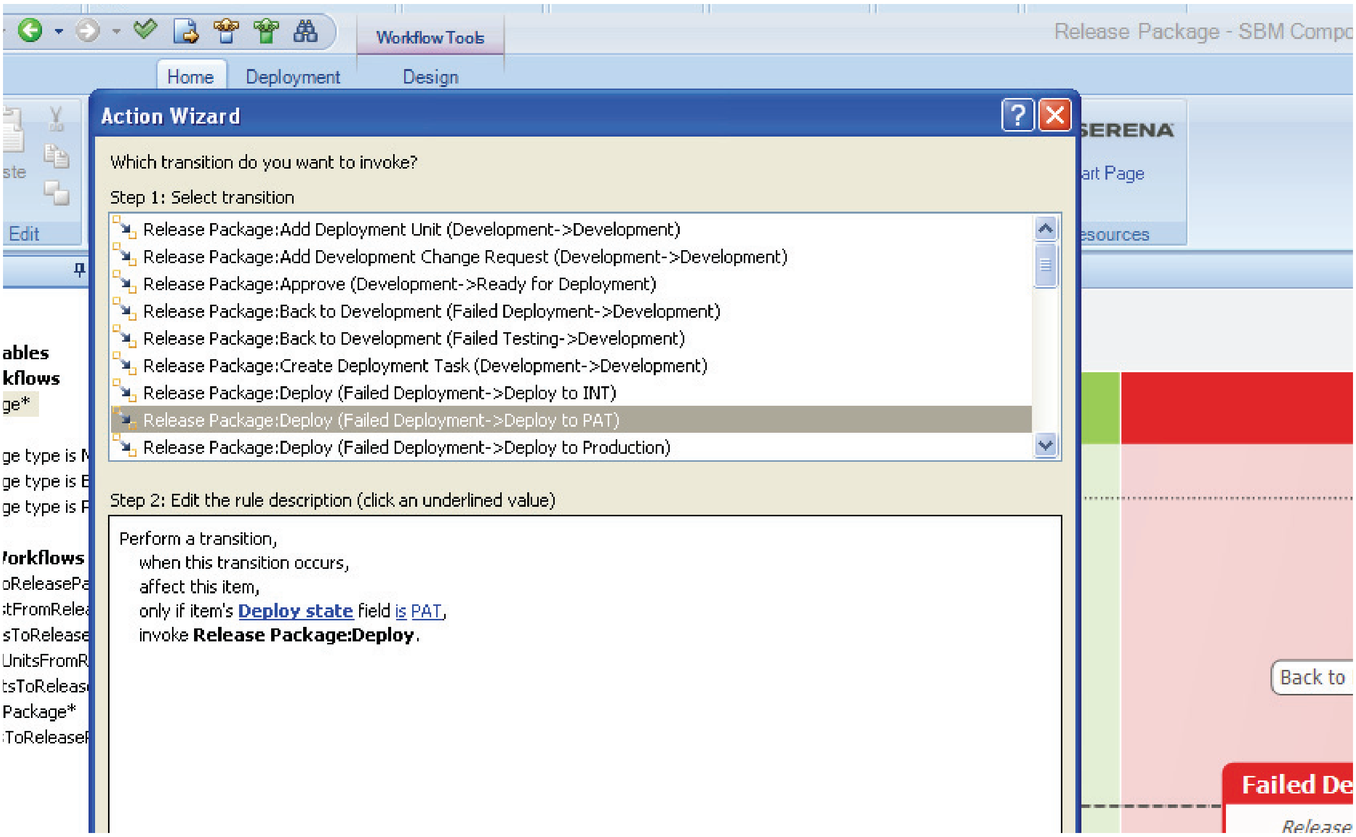
State	Transition	Option	Selection
From state: Failed Deployment To state: Deploy to PAT	Deploy	Options	Quick transition and Hide button on form
		Form	None
		Field Privileges	default values
		Field Overrides	Failed Deploy Transition ■ Read Only ■ Set to default: Fail Deployment
		Actions	Invoke deployReleasePackage Orchestration workflow
		Restrict by Type	default values
		Restrict by Role	default values

6 Add a new re-deploy transition action.

Set the rule as follows:

> Perform a transition > when this transition occurs > affect this item > only if item's Deploy state field is PAT > invoke Release Package:Deploy

The rule settings are shown in the following figure.



Action Wizard

Which transition do you want to invoke?

Step 1: Select transition

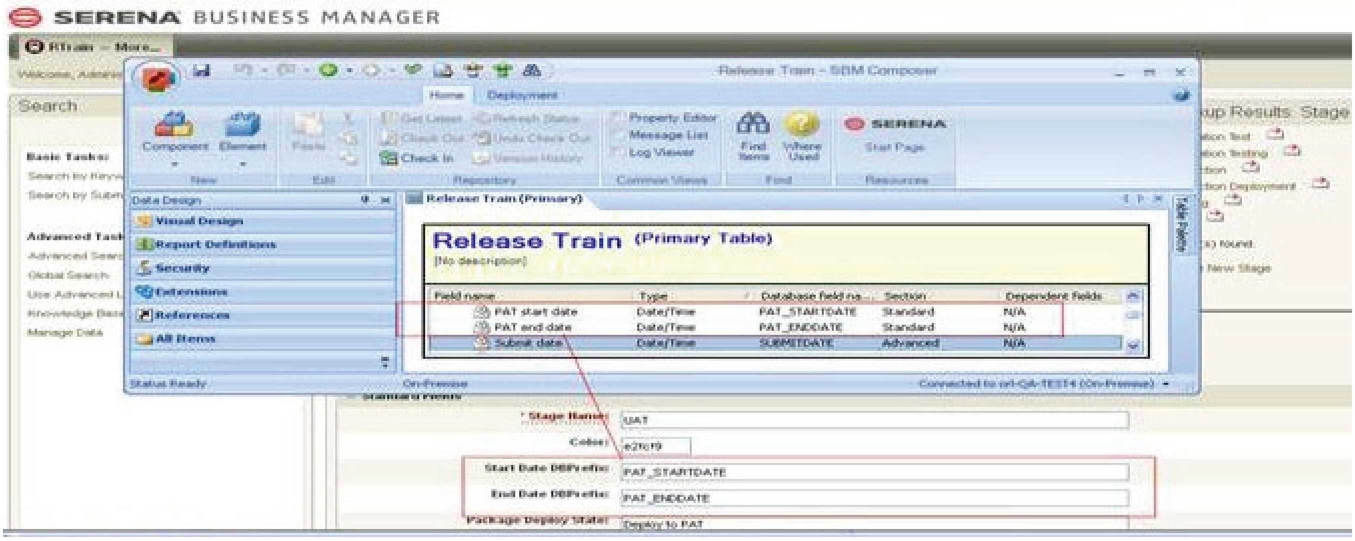
- Release Package:Add Deployment Unit (Development->Development)
- Release Package:Add Development Change Request (Development->Development)
- Release Package:Approve (Development->Ready for Deployment)
- Release Package:Back to Development (Failed Deployment->Development)
- Release Package:Back to Development (Failed Testing->Development)
- Release Package:Create Deployment Task (Development->Development)
- Release Package:Deploy (Failed Deployment->Deploy to INT)
- Release Package:Deploy (Failed Deployment->Deploy to PAT)
- Release Package:Deploy (Failed Deployment->Deploy to Production)

Step 2: Edit the rule description (click an underlined value)

Perform a transition,
when this transition occurs,
affect this item,
only if item's **Deploy state** field is PAT,
invoke **Release Package:Deploy**.

7 Configure the Stage auxiliary table to relate the new stage to the Release Train Start and End Dates and the Release Package Deploy state.

- Relate the PAT Start and End date fields in the Release Train table. This relationship is indicated in the following figure.



Field name	Type	Database field na...	Section	Dependent fields
PAT start date	Date/Time	PAT_STARTDATE	Standard	N/A
PAT end date	Date/Time	PAT_ENDDATE	Standard	N/A
Submit date	Date/Time	SUBMITDATE	Advanced	N/A

* Stage Name: UAT

Color: e2fcf9

Start Date DBPrefix: PAT_STARTDATE

End Date DBPrefix: PAT_ENDDATE

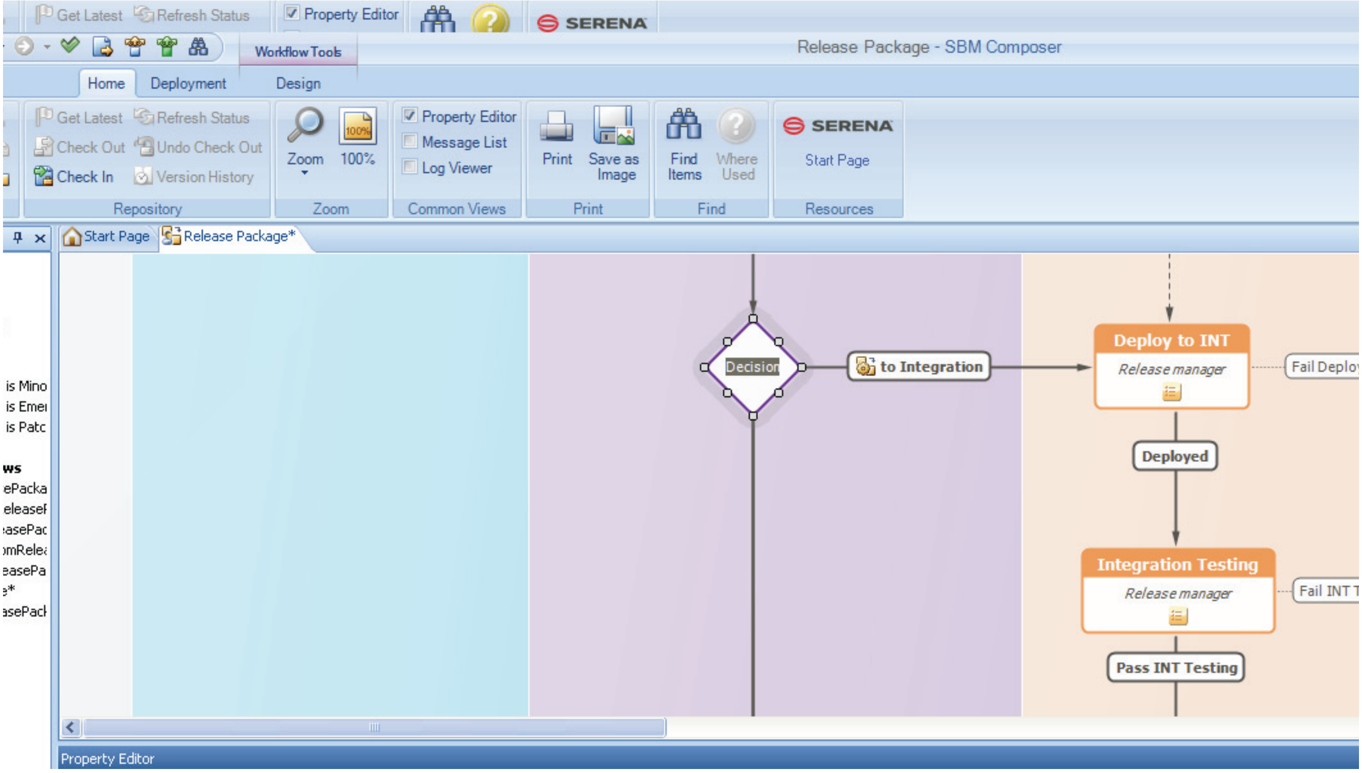
Package Deploy State: Deploy to PAT

- Relate the new Release Package Deploy state to the Stage Package Deploy State text field. This will allow any deployment tasks that are associated to this new Stage to execute when the Deploy transition is executed in the Release Package for the new Stage process.

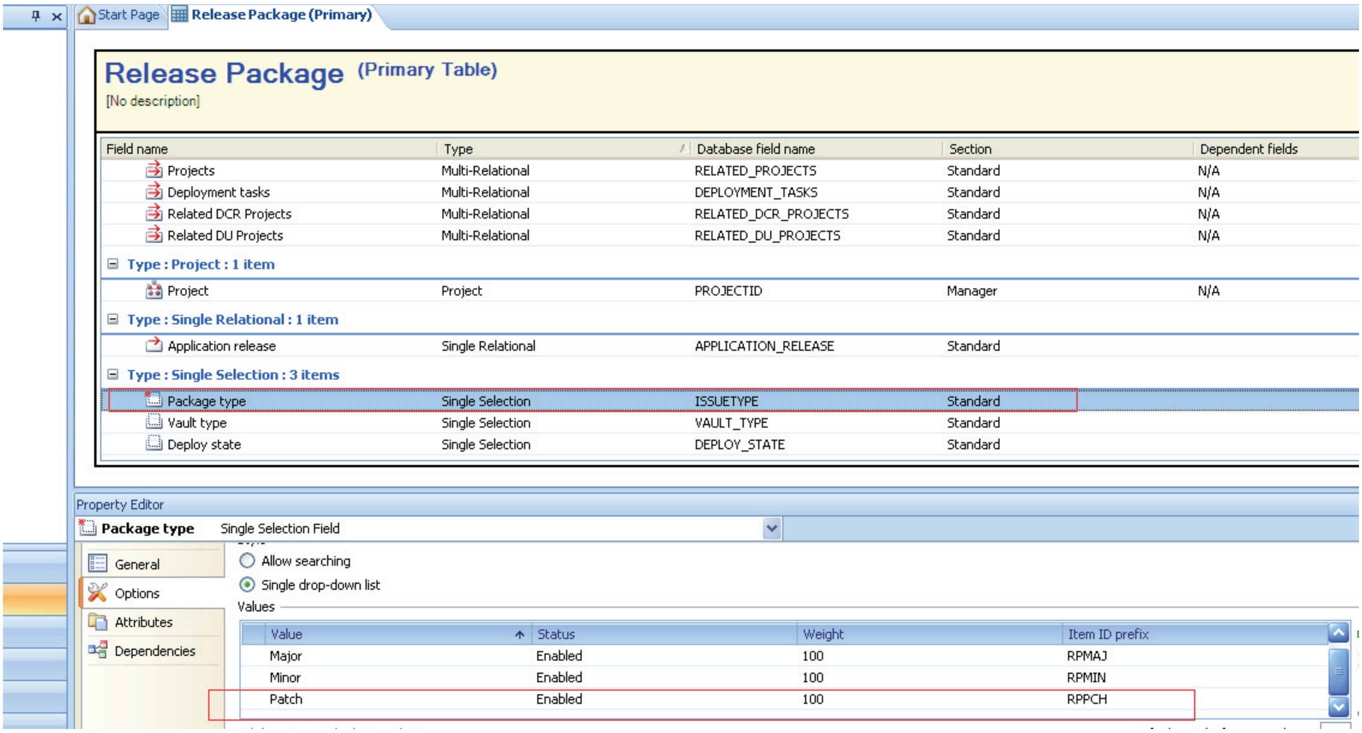
8 Optionally add a new deploy decision rule to the Release Package stage process.

In our example, the new stage process execution is controlled by Release Package release type, so we will add a decision rule. Stages that are used for every release type do not require a decision rule.

a Add the decision in the workflow as shown in the following figure.



b Add a new **Package type** single-selection field value as shown in the following figure.



Release Package (Primary Table)

[No description]

Field name	Type	Database field name	Section	Dependent fields
Projects	Multi-Relational	RELATED_PROJECTS	Standard	N/A
Deployment tasks	Multi-Relational	DEPLOYMENT_TASKS	Standard	N/A
Related DCR Projects	Multi-Relational	RELATED_DCR_PROJECTS	Standard	N/A
Related DU Projects	Multi-Relational	RELATED_DU_PROJECTS	Standard	N/A
Type : Project : 1 item				
Project	Project	PROJECTID	Manager	N/A
Type : Single Relational : 1 item				
Application release	Single Relational	APPLICATION_RELEASE	Standard	
Type : Single Selection : 3 items				
Package type	Single Selection	ISSUETYPE	Standard	
Vault type	Single Selection	VAULT_TYPE	Standard	
Deploy state	Single Selection	DEPLOY_STATE	Standard	

Package type Single Selection Field

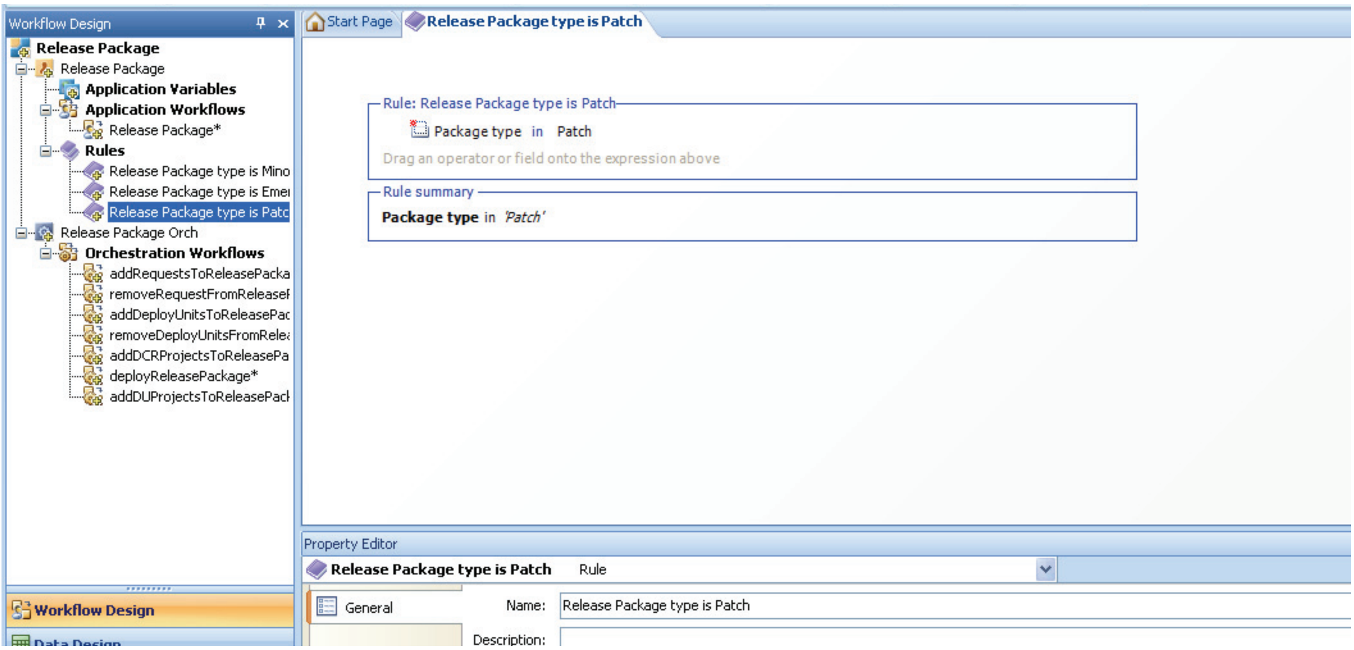
○ Allow searching

● Single drop-down list

Values

Value	Status	Weight	Item ID prefix
Major	Enabled	100	RPMAJ
Minor	Enabled	100	RPMIN
Patch	Enabled	100	RPPCH

c Add a rule for the selection, **Package type** in Patch, as shown in the following figure.


Workflow Design
Release Package
Release Package
Application Variables
Application Workflows
Release Package*
Rules
Release Package type is Mino
Release Package type is Emer
Release Package type is Patc
Release Package Orch
Orchestration Workflows
addRequestsToReleasePacka
removeRequestFromReleaseF
addDeployUnitsToReleasePac
removeDeployUnitsFromRelea
addDCRProjectsToReleasePa
deployReleasePackage*
addDUProjectsToReleasePack
Workflow Design
Data Design
Start Page
Release Package type is Patch
Property Editor
General
Description:

Rule: Release Package type is Patch

Package type in Patch

Drag an operator or field onto the expression above

Rule summary

Package type in *'Patch'*

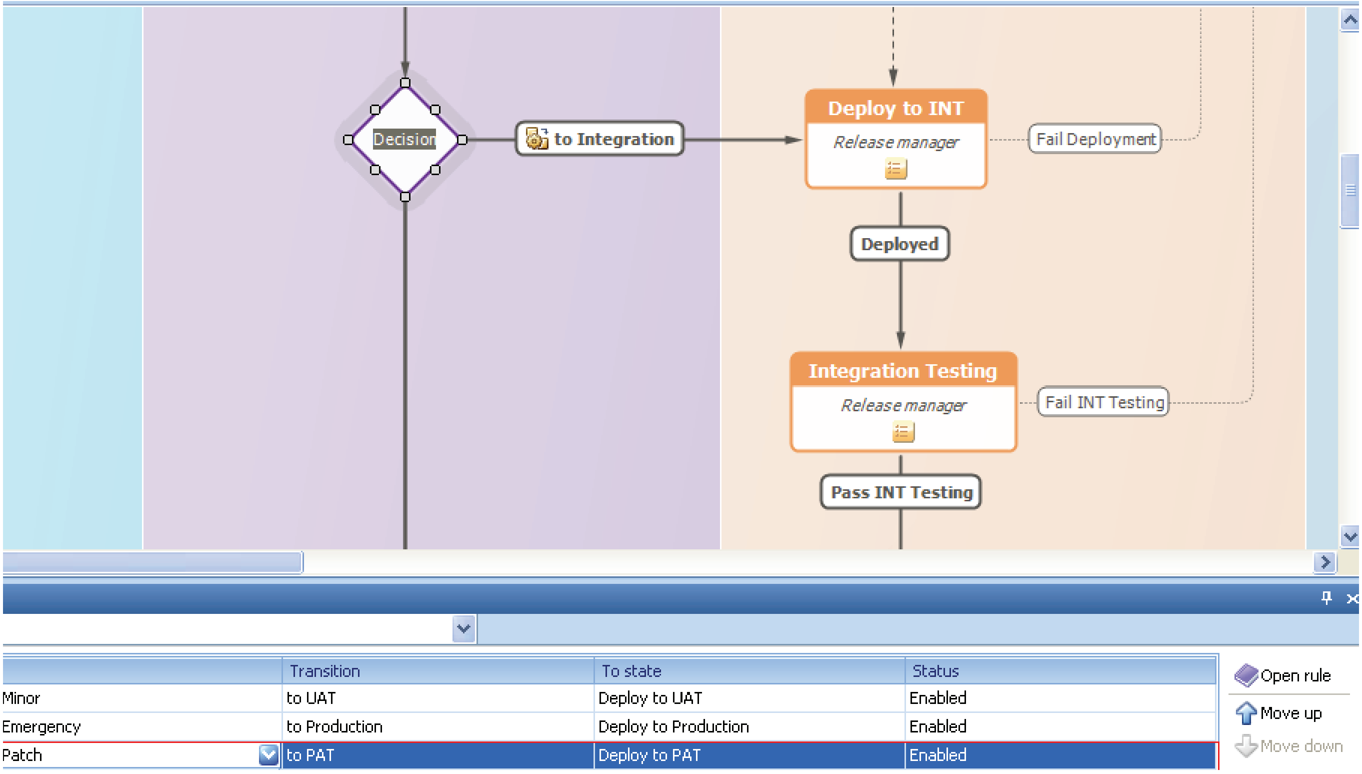
Release Package type is Patch Rule

Name: Release Package type is Patch

d Add a new transition for the deploy decision: add a transition **to PAT** from the Deploy Decision box to the state **Deploy to PAT** as shown in the following table.

To and From	Transition	Option	Selection
From decision box: Deploy To state: Deploy to PAT	to PAT	Options	Quick transition and Hide button on form
		Form	None
		Field Privileges	default values
		Field Overrides	Failed Deploy Transition ■ Read Only ■ Set to default: Fail Deployment
		Actions	Invoke deployReleasePackage Orchestration workflow
		Restrict by Type	default values
		Restrict by Role	default values

e Add the new rule to the deploy decision box. Specify the To transition and state as shown in the following figure.



	Transition	To state	Status
Minor	to UAT	Deploy to UAT	Enabled
Emergency	to Production	Deploy to Production	Enabled
Patch	to PAT	Deploy to PAT	Enabled

Documentation References

- Complete documentation on configuring workflows in SBM is in the *Serena Business Manager SBM Composer Guide* in "Managing Workflows".
- Complete documentation on configuring tables in SBM is in the *Serena Business Manager System Administrator Guide* in "Table Configuration".

Adding Development Change Request and Deployment Unit Providers

The Serena Release Manager default implementation includes Development Change Request (DCR) provider connections for SBM and Dimensions CM and a Deployment Unit (DU) provider connection for Dimensions CM. You can extend the integration to use DCRs and DUs from other systems using the Serena Release Manager configurable DCR and DU providers.

In Serena Release Manager, a provider is any Java implementation of the DCR or DU interface that is implemented and registered following the procedures described in this section.

To see if a provider you are looking is already implemented for Serena Release Manager, please check the most current Serena Release Manager documentation and the online knowledgebase on the Serena Customer Support website.

To implement the DCR and DU providers, see the following sections:

- "Creating a Class for Your Provider" on page 93
- "Creating Properties Files for Your Providers" on page 93
- "Building and Packaging" on page 96
- "Telling Serena Release Control to Use This Provider" on page 96

Creating a Class for Your Provider

Create a class file that implements the IRequestsProvider interface or the IDeployUnitsProvider interface.

Authentication information is kept inside the ISessionData structure and is populated before each initialization of providers. You can keep session-sensitive data using setAttribute and getAttribute methods in ISessionData.

See the Java documentation for more information about methods.

Examples

A snippet of the Java code that creates a simple file system class by implementing the IRequestsProvider interface is shown in the following figure.

com.serena.rlm.provider.fs.FSRequestsProvider

```
public class FSRequestsProvider extends FSCustomProvider implements IRequestsProvider {

    private String requestsFile;
```

A snippet of the Java code that creates a simple file system class by implementing the IDeployUnitsProvider interface is shown in the following figure.

com.serena.rlm.provider.fs.FSDeployUnitsProvider

```
public class FSDeployUnitsProvider extends FSCustomProvider implements
     IDeployUnitsProvider {

    private String depunitsFile;
    private String stagesFile;
    private String depareaFile;
```

Creating Properties Files for Your Providers

Using the recommended spring dependency injection mechanism, as shown in the included examples, create separate properties files for provider definition and provider instance-specific parameters as follows:

- Define your provider's class and its parameter definition, but not values, in an XML definition file. See "Designating the Details for Each Provider" on page 93.
- Define all instance-specific values for parameters in a properties file. See "Telling Serena Release Control to Use This Provider" on page 94.

Designating the Details for Each Provider

Using the spring dependency injection mechanism, you define your provider's class and its parameter definition, but not values, in an XML definition file.

Serena provides the `provider-dm.xml` file for Dimensions CM, provider of DCRs and DUs and `provider-sbm.xml` file for SBM, provider of DCRs.

The following example implements the spring dependency injection mechanism for a simple file system provider.

Example

`provider-fs.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:context="http://www.springframework.org/schema/context"
     xmlns:util="http://www.springframework.org/schema/util"
     xsi:schemaLocation="http://www.springframework.org/schema/beans
         http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
         http://www.springframework.org/schema/context
         http://www.springframework.org/schema/context/spring-context-3.0.xsd
         http://www.springframework.org/schema/util
         http://www.springframework.org/schema/util/spring-util-3.0.xsd"
         default-lazy-init="true">

    <!-- enable processing of annotations such as @Autowired and @Configuration -->
    <context:annotation-config/>
    <context:component-scan base-package="com.serena.rlm.provider.fs"/>

    <bean
     class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
        <property name="ignoreUnresolvablePlaceholders" value="true"/>
        <property name="order">
            <value>1</value>
        </property>
    </bean>

    <bean id="requestsProvider" class="com.serena.rlm.provider.fs.FSRequestsProvider">
        <property name="providerName" value ="${requests.provider.name}"/>
       <property name="providerDescription" value ="${requests.provider.description}"/>
        <property name="requestsFile" value ="${provider.fs.requests.file}"/>
    </bean>


    <bean id="deployUnitsProvider"
     class="com.serena.rlm.provider.fs.FSDeployUnitsProvider">
        <property name="providerName" value ="${deploy.units.provider.name}"/>
        <property name="providerDescription" value
     ="${deploy.units.provider.description}"/>
        <property name="depunitsFile" value ="${provider.fs.depunits.file}"/>
        <property name="stagesFile" value ="${provider.fs.stages.file}"/>
        <property name="depareaFile" value ="${provider.fs.deparea.file}"/>
    </bean>

</beans>
```

Telling Serena Release Control to Use This Provider

Using the spring dependency injection mechanism, you define all instance-specific values for parameters in a properties file.

It is not required to use a properties file separate from the xml file in the provider implementation. However, usage of a properties file is a good practice and is included in the example provided. Using a properties file allows you to define several possible configurations so you can easily change details without code modification. Without a properties file, you must hard code name, description, and other specific parameters for your provider.

Examples

`fs_example.properties`

```
# requests provider definitions
requests.provider.name = filesystem
requests.provider.description = Simple file-system Request Provider

# deploy units provider definitions
deploy.units.provider.name = filesystem
deploy.units.provider.description = Simple file-system Deployment Unit Provider

#
provider.fs.requests.file=requests.txt
provider.fs.depunits.file=depunits.txt
provider.fs.stages.file=stages.txt
provider.fs.deparea.file=areas.txt
```

The text files referenced in the preceding example, `requests.txt`, `depunits.txt`, `stages.txt`, and `areas.txt` are shown in the following examples. This is a simple file-system example where the content of these could be populated by any mechanism you implement, such as JDBC, Web services, and other protocols.

`requests.txt`

```
# list of mocked requests should be defined here
# use the following format
# <request_id>|<request_name>|<request_status>|<request_url>
ECR0001|Old delete icon in POA toolbars|Assigned to QA|http://almmashups.serena.com/
     tmtrack/tmtrack.dll?IssuePage&RecordId=78506&Template=view&TableId=1000
ECR0002|WEB-Cannot set owner for a project|Closed|http://almmashups.serena.com/tmtrack/
     tmtrack.dll?IssuePage&RecordId=78501&Template=view&TableId=1000
ECR0003|Approved dialog from Project node inconsistent to the same functionality on
     Area|Code & Unit Test|http://almmashups.serena.com/tmtrack/
     tmtrack.dll?IssuePage&RecordId=11531&Template=view&TableId=1000
```

`depunits.txt`

```
# list of mocked deployment units should be defined here
# use the following format
# <depunit_id>|<depunit_name>|<depunit_project_name>
DEP0001|Deployment unit 1|FS:RLM_TEST_1
DEP0002|Deployment unit 2|FS:RLM_TEST_2
DEP0002|Deployment unit 3|FS:RLM_TEST_3
```

`stages.txt`

```
# list of mocked stages should be defined here
# use the following format
# <stage_id>|<stage_name>|<stage_projects>
ST0001|SIT|QLARIUS:Q1S,QLARIUS:Q2S,QLARIUS:RLM_TEST
```

`areas.txt`

```
# list of mocked areas should be defined here
# use the following format
#
     <area_id>|<area_name>|<area_directory>|<area_stage_id>|<area_status>|<depunit_proj
     ect_name>
AR0001|Dev area|c:\work\|SIT|Open|QLARIUS:RLM_TEST
AR0002|Dev area|c:\work2\|SIT|Open|QLARIUS:RLM_TEST2
```

Building and Packaging

After you have created the Java class and supporting files as described in the preceding sections, you should build and package your provider `jar` file to be distributed to your Serena Release Managerserver.

The compilation and packaging should be compliant with Java 6 and Tomcat 6.x.

1. Compile your sources. For example, use `ant` to compile and create a file with a name similar to the following:

   ```
   com.serena.rlm.provider.fs.jar
   ```

2. Create a `zip` file with all the folders and properties files for the new provider. For example:

   ```
   provider_fs.zip
   ```

3. Copy the `zip` file to the folder structure under the `webapps` folder. For example:

   ```
   C:\Program Files\Serena\common\Tomcat\6.0\webapps\rlm
   ```

Telling Serena Release Control to Use This Provider

After you have built and packaged your new provider, you tell Serena Release Control to use this provider, or register it, as instructed in the following procedure.

Add provider instructions (replace `provider_fs.zip` with your archive `zip` file).

To add provider instructions:

1. Copy your archive `zip` file to your target server. For example, `provider_fs.zip`.

2. Back up your Serena Release Manager common Tomcat Web server `rlm` folder. For example:

   ```
   C:\Program Files\Serena\common\Tomcat\6.0\webapps\rlm
   ```

3. Stop the Serena Common Tomcat service.

4. Unzip your archive `zip` file, such as `provider_fs.zip`, to your `*\Tomcat 6.0` folder. For example:

   ```
   C:\Program Files\Serena\common\tomcat\6.0
   ```

5. Navigate to the Serena Release Manager common Tomcat Web server `classes` folder. For example:

```
C:\Program Files\Serena\common\tomcat\6.0\webapps\rlm\WEB-
INF\classes
```

6 Open `providers.properties` in your text editor.

7 Adding providers:

a Add your new development change request provider to `requests.providers.keys`. For example:

```
requests.providers.keys = sbm_issues,fs_example
```

a Add your new deployment units provider to `deploy.units.providers.keys`. For example:

```
deploy.units.providers.keys = dm_qlarius,fs_example
```

8 Save `providers.properties`.

9 Start the Serena Common Tomcat service.

Example

In the following example, the DCR provider `fs_example` has been added to the `requests.providers.keys` and the DU provider `fs_example` has been added to the `deploy.units.providers.keys`. In this example, Serena Release Manager would retrieve requests from the `sbm_issues` provider and the `fs_example` provider, and would retrieve deployment units from the `dm_qlarius` provider and the `fs_example` provider.

`providers.properties`

```
# requests provider keys
requests.providers.keys = sbm_issues,fs_example

# deploy units provider keys
deploy.units.providers.keys = dm_qlarius,fs_example
```

Addressing Web Server Port Conflicts

Serena Release Control runs using the Serena Common Web server, which is an Apache Tomcat Web server. The Serena Release Control installer automatically installs and configures the Serena Common Web server to run on the default port of 9095. If this port is already in use by another application on your server, or if you already have an instance of the Serena Common Web server running on a different port on this server, you will need to decide how you want to proceed.

Scenarios you may encounter in which you will need to use non-default installation and configuration procedures are as follows:

1 There is a Serena Common Tomcat installed on port 8080 or a port other than 9095.

The Serena Release Control installer will detect its presence, although it does not check the port number. For this scenario, the options are as follows:

a Change the Serena Common Web server port to 9095.

> **CAUTION!** Serena Release Control is configured to use port 9095 by default, but changing the Web server to use port 9095 may impact other Serena products if they rely on a previously configured port, such as 8080. Make sure all Serena products using the same Serena Common Web server are configured to use the same port number, or install the products on separate servers so that they can use different Serena Common Web servers with different port numbers.

b Do not change the Serena Common Web server port, but instead reconfigure Serena Release Control restful widgets to use a port other than 9095. See "Configuring a Non-Default Web Server Port in the Process Apps" on page 99.

2 There is a non-Serena Tomcat, IIS, or other Web server installed on port 9095 on the Serena Release Control server, so port 9095 is already in use.

The options are as follows:

a Change the non-Serena Web server configuration to use a different port.

b Install the Serena Common Tomcat Web server on different port and reconfigure Serena Release Control to use that port. See "Changing the Port on Which the Common Web Server Runs" on page 98 and "Configuring a Non-Default Web Server Port in the Process Apps" on page 99.

> **NOTE** Use the default installation and configuration procedures on the Serena Release Control server in the following scenarios:
>
> - There is a non-Serena Tomcat, IIS, or other Web server installed on this server on a port other than 9095, and port 9095 is free.
> - There is a Serena Common Tomcat Web server already installed on this server on port 9095.
> - There is not a Web server on this server.
>
> See Chapter 2, "Installing Serena Release Control" on page 18 and Chapter 3, "Activating Serena Release Control" on page 25.

Changing the Port on Which the Common Web Server Runs

The default port on which this is configured to run is 9095. If you want to use a different port, you must first install the Common Web server and Serena Release Control, and then change the port number as follows:

To change the Serena Release Control Web server port number:

1. Invoke the Command Prompt, or command line interface, for your Windows environment. For example, from the Start menu, run `cmd.exe`.

2. At the command prompt, navigate to the folder where you downloaded the Serena Release Control installation file executable, `Release Control.exe`. For example:

   ```
   cd \Downloads\RCBuild\Win64
   ```

3. Enter the following command at the prompt:

   ```
   "Release Control.exe" /V"/L*v "%TEMP%\<logfilename>.log" TC_PORT=<port#>"
   ```

For example, here is the command to log the install and change the Tomcat Web server port to 1234:

```
"Release Control.exe" /V"/L*v "%TEMP%\test_install.log" TC_PORT=1234"
```

The Serena Release Control installer appears.

4 Follow the prompts to modify your Web server installation with the specified port.

NOTE Versions of Serena Release Control prior to version 3.1 use a default Web server port of 8080.

Configuring a Non-Default Web Server Port in the Process Apps

If you want to run the Serena Common Web server on a port other than 9095, you need to modify the port numbers in the Serena Release Control process apps that have restful widgets. After you modify the process apps, you need to redeploy them.

You must install and configure Serena Release Control according to the normal installation procedures before you can reconfigure the process apps with the non-default port number. See Chapter 2, "Installing Serena Release Control" on page 18 and Chapter 3, "Activating Serena Release Control" on page 25.

The form controls in which you need to update the Web server port number are shown in the following table:

Process App	Form	Controls
Release Package	createRelPackage	listDCRProjects, listDUProjects
	addDepUnit	listDepUnits
	addDevChRequest	listRequests
Deployment	createAutomationTask	gridApps, gridEnv, gridProc, gridServ
	createVaultTask	gridDepUnits, gridDepStages, gridDepAreas
	editAutomationTask	gridApps, gridEnv, gridProc, gridServ
	editVaultTask	gridDepUnits, gridDepStages, gridDepAreas

NOTE No updates are needed for Release Train, Application release, and RLM_AUX process apps. They are implemented using embedded reports and do not use restful widgets.

Changing the Port Number in the Forms

You must find the restful widgets in the forms as listed in the preceding table, and update each occurrence of the default port number, 9095, to the port number for your Web server installation. Once you have completed the updates, you must redeploy the applications.

To modify and redeploy the process apps:

1 Open the process app you want to change in SBM Composer. For example, open Release Package from the Application Repository.

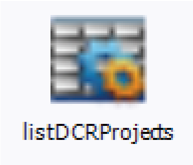
2 Display the **Visual Design** view.

3 Under **Forms**, select one of the forms listed in the preceding table of process apps, forms, and controls. For example, select `createRelPackage`.

The selected form displays.

4 Verify that the form is checked out. If the message "This item is not checked out. Click here to check it out." displays at the top, click to check it out.

5 Scroll until you see a control that contains a restful widget and select it. For example, you'll see the restful widgets icon and the name of the control, such as:



6 Update the control as follows:

a In the **Property Editor** view, select the **General** tab.

b Click the **Configure URL** button.

c In the URL displayed at the top, change the port value of 9095 to the non-default port number you specified during your common Tomcat installation. For example, your modified URL would look something like this:

```
http://localhost:8088/rlm/services/ReleaseRequestService/
getRequestProjects
```

You should leave the host name pointing to localhost, since all Web services communication goes through localhost and Serena Release Manager uses configuration files to resolve the host names for integrating systems.

d Click **Update outputs**.

NOTE If you receive the error "Unable to Configure Service", you may need to change **designMode** to `true` and then retry.

e Click **OK**.

7 Save and check in your changes.

8 Continue for each restful widget control in each process app.

9 When you are finished, redeploy the process apps as follows:

a In SBM Composer, publish each of the Serena Release Control process apps.

b After successfully publishing each process app, deploy each of the process apps.

As you deploy, verify that the endpoints of the process application destinations are pointing to the port number you specified during the installation rather than the default port number of 9095. If they are not, update them as needed before deploying.

Chapter 7

Troubleshooting

This chapter gives information on troubleshooting issues in Serena Release Manager.

Troubleshooting Overview

When you encounter an issue in Serena Release Control, there are several places you can look to determine the problem.

Depending on the area of the product where the error occurs, you may use one or more of the following:

- Information displayed in the Serena Release Control user interface.
- Information stored in the Serena Release Control product log file, `rlm.log`.
- Troubleshooting information for integrating products, such as SBM, Serena Release Vault, and Serena Release Automation.

Information from the Serena Release Control User Interface

Serena Release Control provides as much information as possible in the user interface to help you determine the cause of failures.

Error Messages

Error messages displayed in Serena Release Control are your first indication of problems that have occurred. In many cases, the message will give you enough information to help you resolve a problem. For example, if you have not filled in a required field, the error message will prompt you to do so.

If an error message is returned from one of the integrating products, it may not direct you to the exact solution to the problem. See "Symptoms and Solutions" on page 105 for more assistance in these situations.

Activity Logs

When you deploy a release package, you can click the **Activity Log** tab in the Release Package view to see the status of the deployment tasks that are initiated. This lets you know if the deployment task was successfully initiated and whether it successfully completed or failed.

For deployment task failures, see "Information from the Product Log File" on page 105 for more troubleshooting techniques.

History

To see the change history of a release train, application release, release package, or deployment task, click the History tab in that item’s view. This shows the date and time of the change, a description of the change, and the user who made the change.

Information from the Product Log File

The product log file is `rlm.log`. Logging is done through the common Tomcat Web server using the Apache log4j Java-based logging utility.

The `rlm.log` file contains status messages returned to the AFS server. The level of messaging depends on the settings in the log4j settings. Debugging and append are on by default.

The default location of the `rlm.log` file is the catalina home directory, set by the variable `${catalina.home}`. For example:

```
C:\Program Files\Serena\common\tomcat\6.0
```

To change the location of the log files:

1. Navigate to the Serena Release Manager common Tomcat Web server `classes` folder. For example:

   ```
   C:\Program Files\Serena\common\tomcat\6.0\webapps\rlm\WEB-
   INF\classes
   ```

2. Open the `log4j.properties` file.

3. Change the following line to specify the location for the file.

   ```
   log4j.appender.RLM.file=<drive:path>\rlm.log
   ```

Symptoms and Solutions

Symptoms of unexpected results and their possible solutions are explained in this section as follows:

- "Unexpected Display Results" on page 105
- "Matches Not Found Selections" on page 106
- "Release Package Deployment Fails" on page 107
- "Installer Errors" on page 108
- "Slow Response Time" on page 108

Unexpected Display Results

If the Serena Release Control user interface and data doesn’t appear as it should, here are some possible solutions.

Test Connection is successful but Serena Release Control doesn’t work

If the Test Connection is successful when you configure the destination endpoints for the snapshots but Serena Release Control fails to appear in SBM, verify that the Serena Release Manager service is running in the common Tomcat Web server.

To verify the Serena Release Control Web Services in Apache Tomcat:

1. In your Web browser, browse to the home page for your Tomcat installation. For example:

   ```
   http://localhost:8080
   ```

   The Serena Common Tools page appears.

   If you do not see this page, the common Tomcat Web server is not running or you may have entered an incorrect host and port number combination.

2. Once you have verified that the Serena Common Tools Web server is running, browse to the `rlm` subdirectory of your Tomcat installation. For example:

   ```
   http://localhost:8080/rlm
   ```

   Services that should be active include:

   - DeploymentAutomationService
   - RLMUtilService
   - ReleaseRequestService

   If any of these are not active, double-check your configuration for that service and start the service.

Serena Release Control data is not displayed as expected

If you add something and it doesn't appear in the place it should in the UI, or information you expect to see in a view does not appear, you may not have proper privileges, roles, or ownership set for Serena Release Control in SBM. For example, if you add release trains and they do not appear in the calendar, your privileges probably aren't set properly. If you create a deployment process template and do not have the option to add deployment tasks, the owning role may not be enabled for the RLM Aux project.

See Chapter 3, "Configuring the Administrative User" on page 30.

Similarly, if you create an item as one user and log in as another, you may not see the item you created as the other user, depending on privilege settings and ownership.

Matches Not Found Selections

If selections from integrating systems are not found, such as projects from Serena Release Vault or applications from Serena Release Automation, here are some possible solutions.

We did not find any matches for your request for a release package project selection

If Serena Release Control fails to find any matches for your request when you attempt to select a project for a release package, check the `rlm.log` file for specific error messages.

For example, if you see a message similar to the following in the rlm.log file,

> DimClientException ... Error: Not an authorized user

check to make sure that the same administrative password is set up with the same password in both SBM and Dimensions CM. Also verify that both SBM and Dimensions CM have SSO enabled and share the same SSO server.

We did not find any matches for your request for an automation deployment task application

If Serena Release Control fails to find any matches for your request when you attempt to select an application for an automation deployment task, check the `rlm.log` file for specific error messages.

Examples

- If you see a message similar to the following in the `rlm.log` file,

  ```
  com.mysql.jdbc.exceptions.jdbc4.communicationsException:communication link failure
  ```

  verify that the Nolio server support is configured correctly in the `nolio.properties` file.

- If you see a message similar to the following in the `rlm.log` file,

  Could not connect to MySQL database; not able to connect to the DB.

  verify that the correct URL is specified in the `nolio.properties` file with the right server name.

We did not find any matches for your request for an automation deployment task server

If Serena Release Control fails to find any matches for your request when you attempt to select a server for an automation deployment task, check to make sure that the Nolio process runs outside of Serena Release Control.

In Nolio Automation Center, verify that a server is set up for the selected application, environment, process combination.

Release Package Deployment Fails

If deployment of a release package fails, here are some possible solutions.

A deployment vault task fails when you deploy a release package

Before you try deployment tasks in Serena Release Manager, make sure that the same type of task works in native Dimensions CM.

If you are sure the task works in native Dimensions CM, check the **Activity Log** tab for information. If that does not give enough information for you to identify the problem, check the details in the `rlm.log` file.

A common source of the problem is the user ID setup. Make sure that all of the requirements are met as follows:

- The same user ID and password must be used in both SBM and Dimensions CM.
- SSO must be enabled for both SBM and Dimensions CM on the same SSO server.
- The user ID must be given a role in the Serena Release Control projects, such as Release Packages and Deployment Tasks.

An automation deployment task fails when you deploy a release package

If an automation deployment task fails when you promote a release package, test the server process in native Nolio.

If the process works in native Nolio but not in Serena Release Control, here are some other things to try:

- Verify that the Nolio service is running properly.
- Close the Deployment Task dialog box and reopen it.
- Clear the Web browser cache and retry.

Installer Errors

If the installer fails, here are some possible solutions.

Common Tools files are missing from the install

If it doesn’t detect the Serena Common Web server, the installer may fail with a message similar to this message:

> "Common Tools files are missing from the install. Please ensure these files are present under the common folder before continuing with this install."

If you have Serena Common Web services installed, check to make sure the services are started. If they aren’t, start them.

If you do not have Serena Common Web services installed, you must make sure the Common folder for the Serena Common Web services installer is in the same directory as the `Release Control.exe` file and that the folder has the Common Web services files.

Slow Response Time

A number of factors can affect response time across the network. Some things to check if you are experiencing slow response time include the following.

Using a single physical Server for Serena Release Manager

If you install Serena Release Control, all SBM server components, all Serena Release Vault server components, and Serena Release Automation on the same physical server, you may experience slow response times. The performance of a single server configuration depends on the physical memory allocation to the server, processor type and speed, level of application usage, and network configuration factors such as routing to client machines and network latency.

In medium to large organizations, heavy usage of these applications can have a negative impact on performance. For optimal response time and end user productivity, Serena suggests a minimum of four servers be used for the Serena Release Manager suite.

For example, a recommended architecture is to install SBM, Serena Release Vault, and Serena Release Automation on separate physical servers and make use of a dedicated database server to house the SBM database and the Dimensions CM database.

For small scale usage, testing and proof of concept purposes, installing the entire suite on a single high specification server should not pose a problem.

Appendix A

Workflow Reference

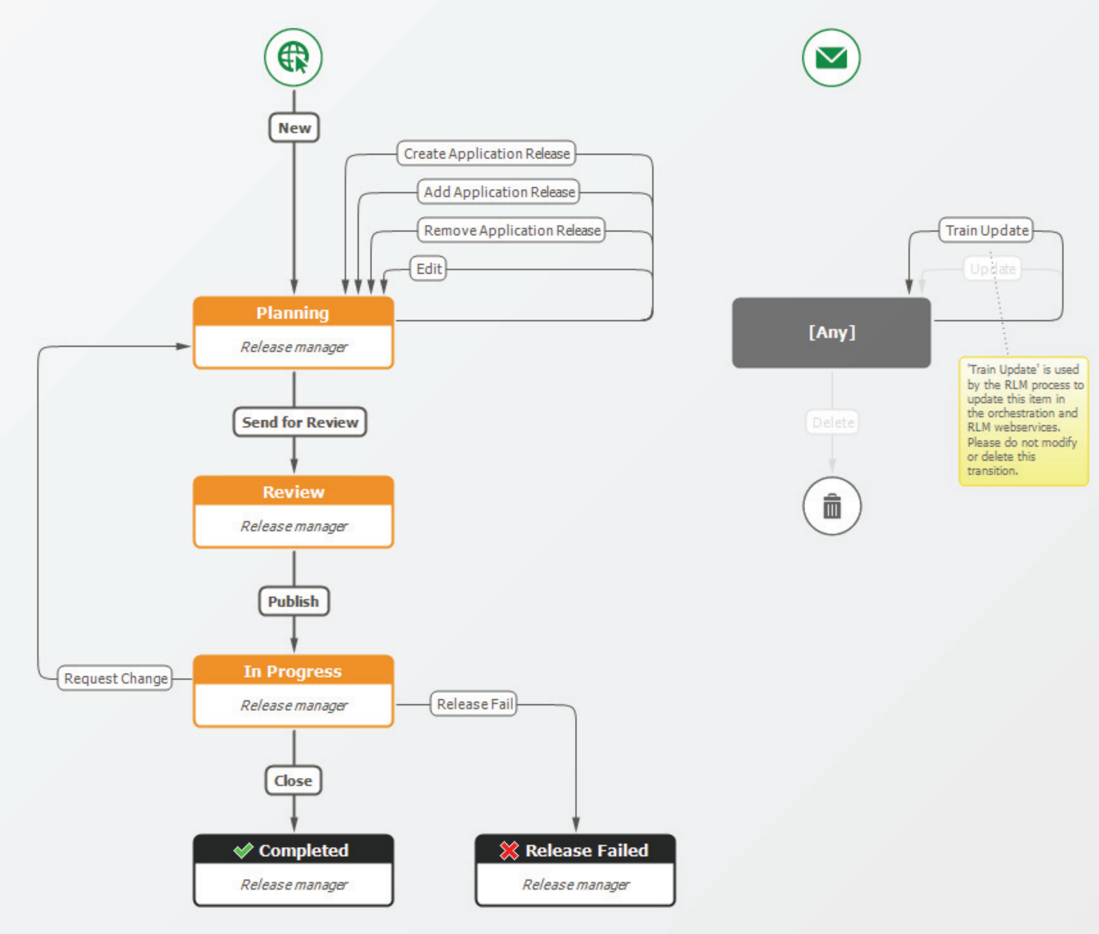
Each major aspect of a release, from the Release Train level to the Deployment Task level, has a lifecycle associated with it. The default release process lifecycles in Serena Release Control for Release Train, Application Release, Release Packages, Deployment Tasks, and Deployment Process Templates are based on typical release workflows.

The states in the lifecycle reflect the actions users take in Serena Release Control as they work with release information. The workflow information is reflected in the actions that appear on the user interface. As you use Serena Release Control, the user interface leads you through the workflow, so you always know the state of the release entities and what actions are pending.

This appendix gives a reference of workflows provided in the default version of Serena Release Control.

Release Train Workflow

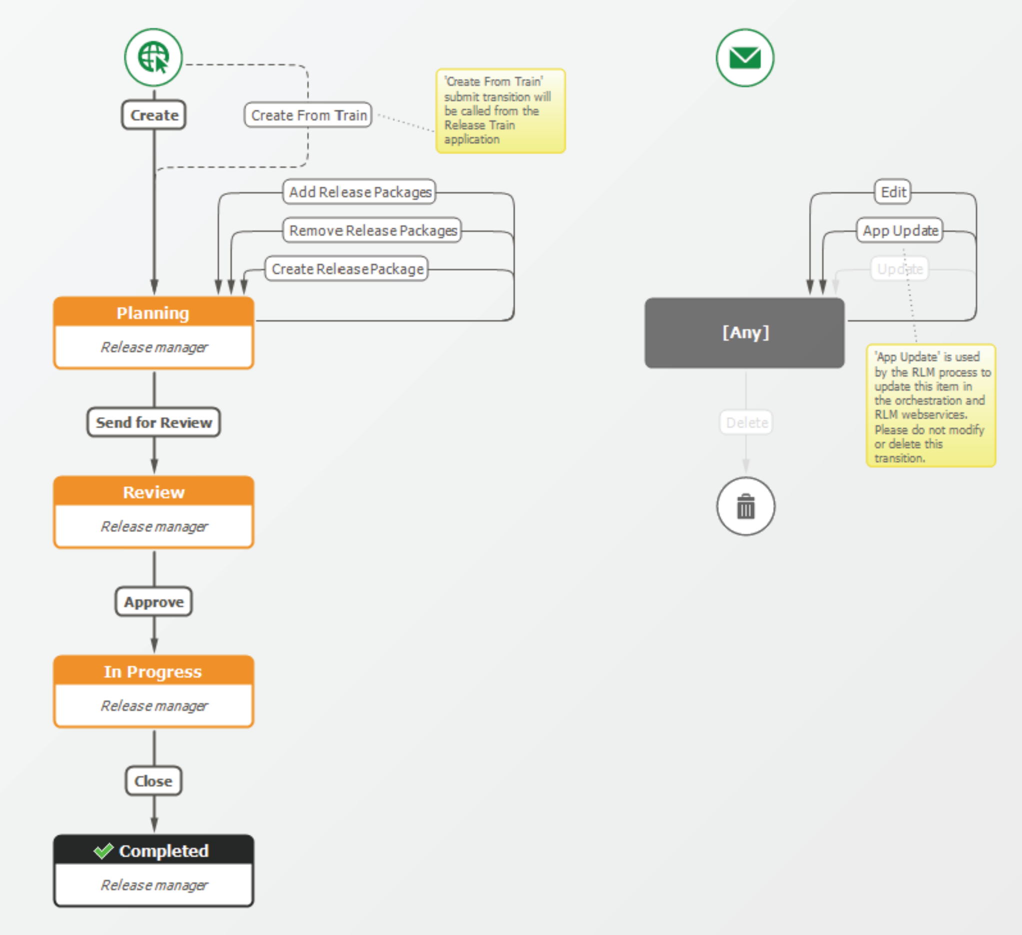
The Release Train workflow in the default version of Serena Release Control is shown in the following figure.



'Train Update' is used by the RLM process to update this item in the orchestration and RLM webservices. Please do not modify or delete this transition.

Application Release Workflow

The Application Release workflow in the default version of Serena Release Control is shown in the following figure.



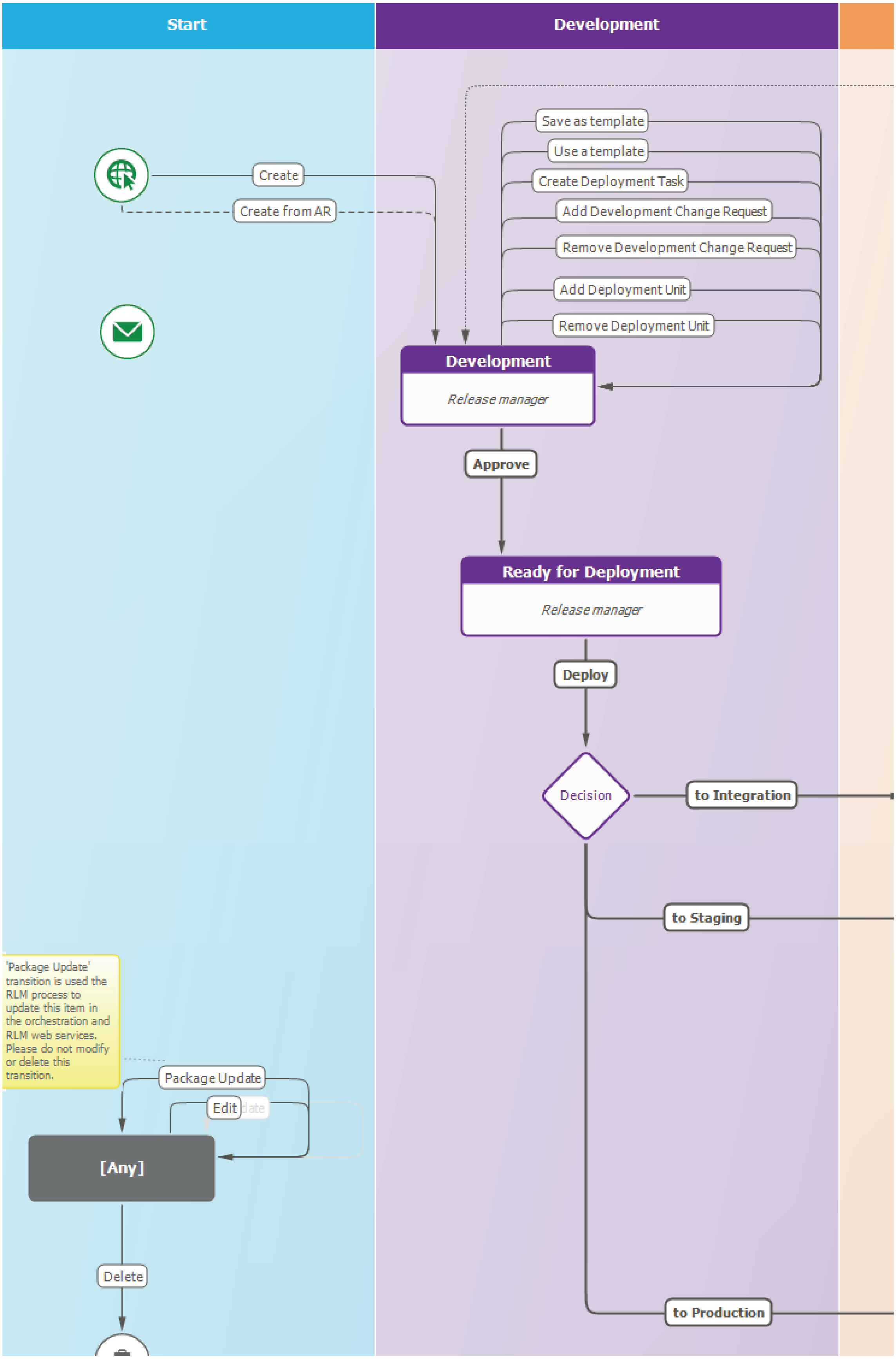
'Create From Train' submit transition will be called from the Release Train application

'App Update' is used by the RLM process to update this item in the orchestration and RLM webservices. Please do not modify or delete this transition.

Release Package Workflow

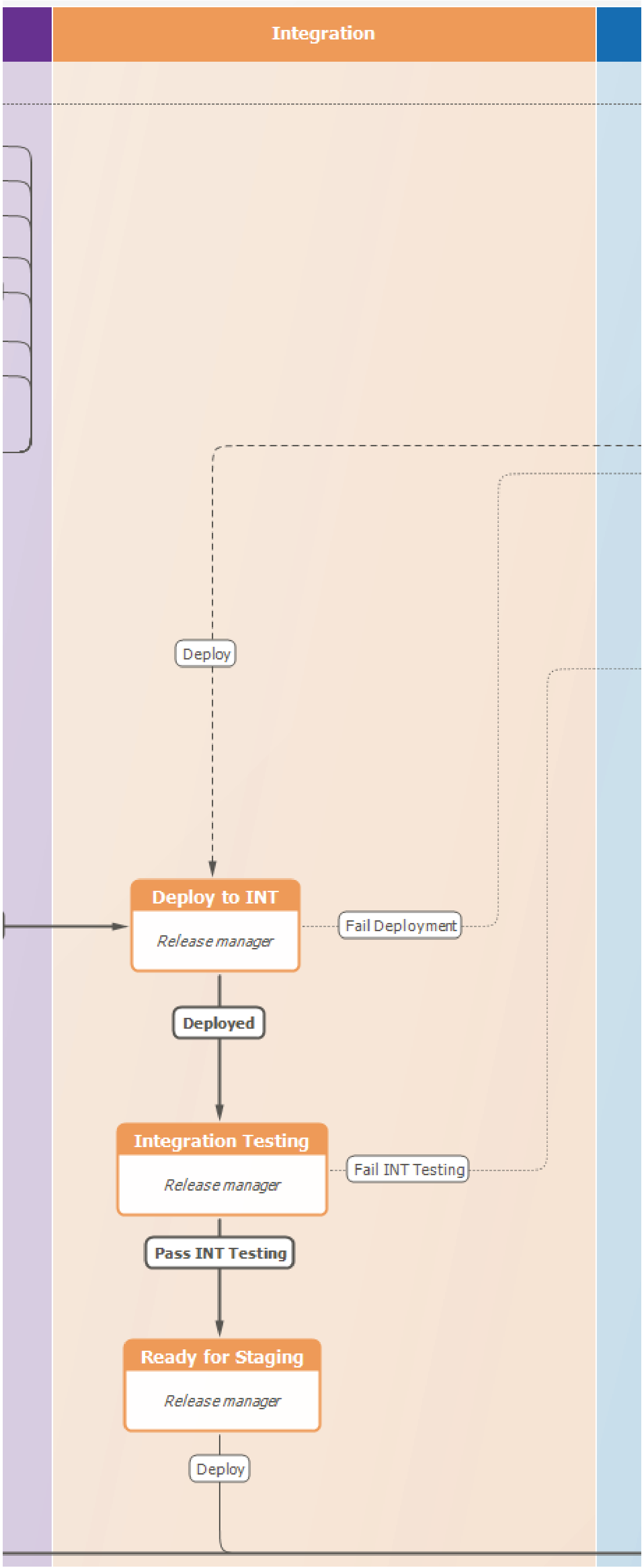
The Release Package workflow in the default version of Serena Release Control is shown in the following figures. Because the workflow is a conditional workflow with multiple swimlanes, the workflow is shown in parts for ease of viewing.

Start and Development States

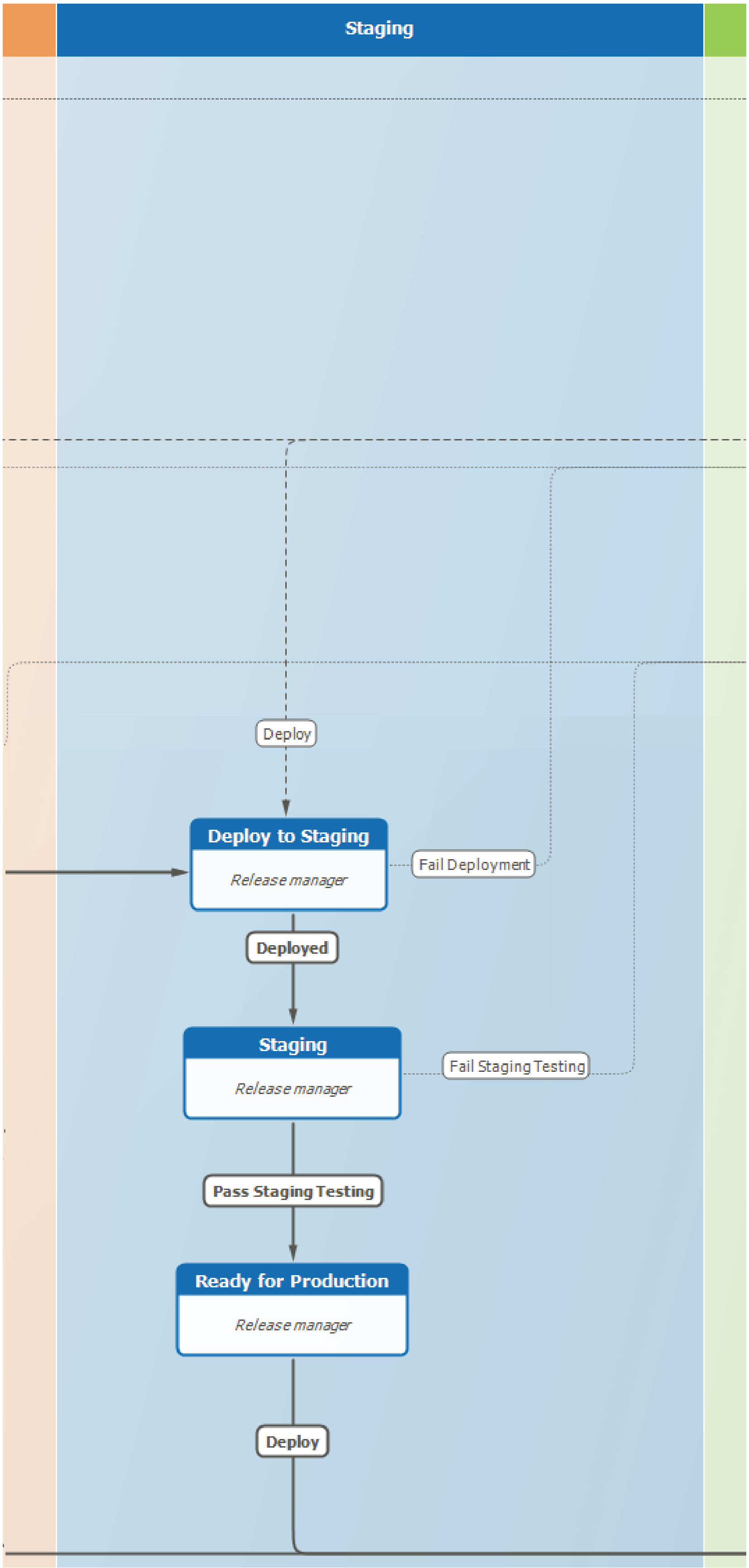


'Package Update' transition is used the RLM process to update this item in the orchestration and RLM web services. Please do not modify or delete this transition.

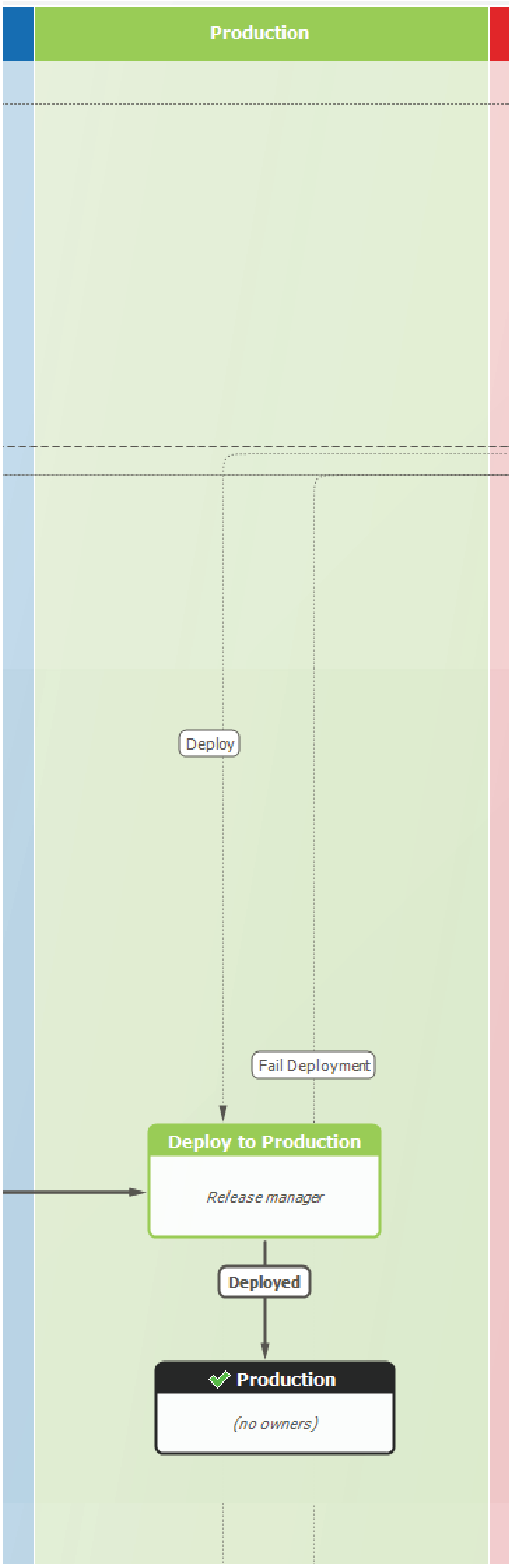
Integration State



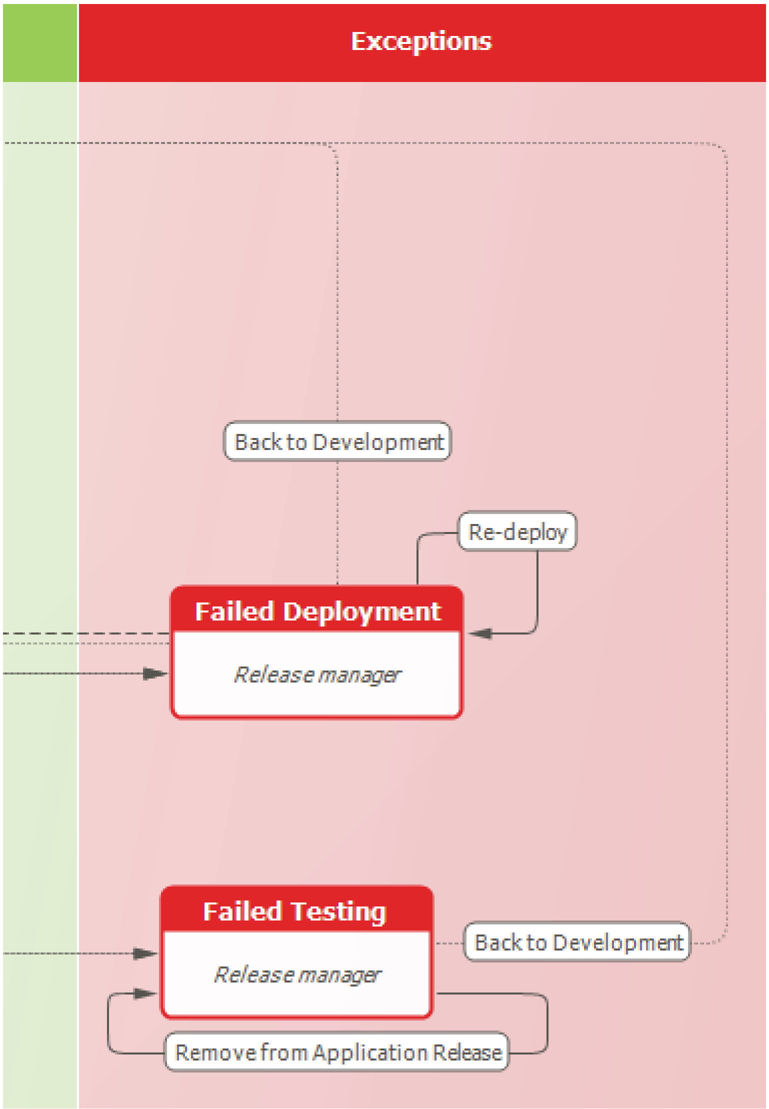
Staging State



Production State



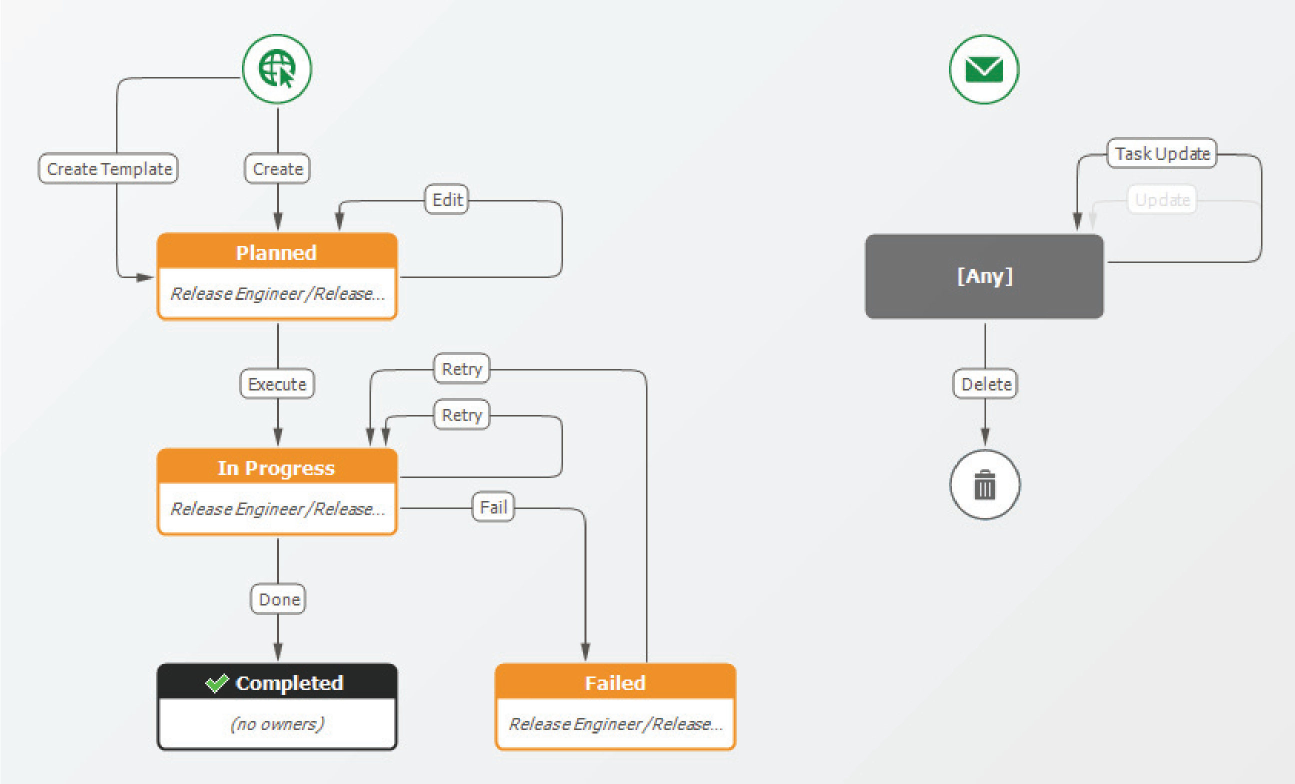
Exceptions State



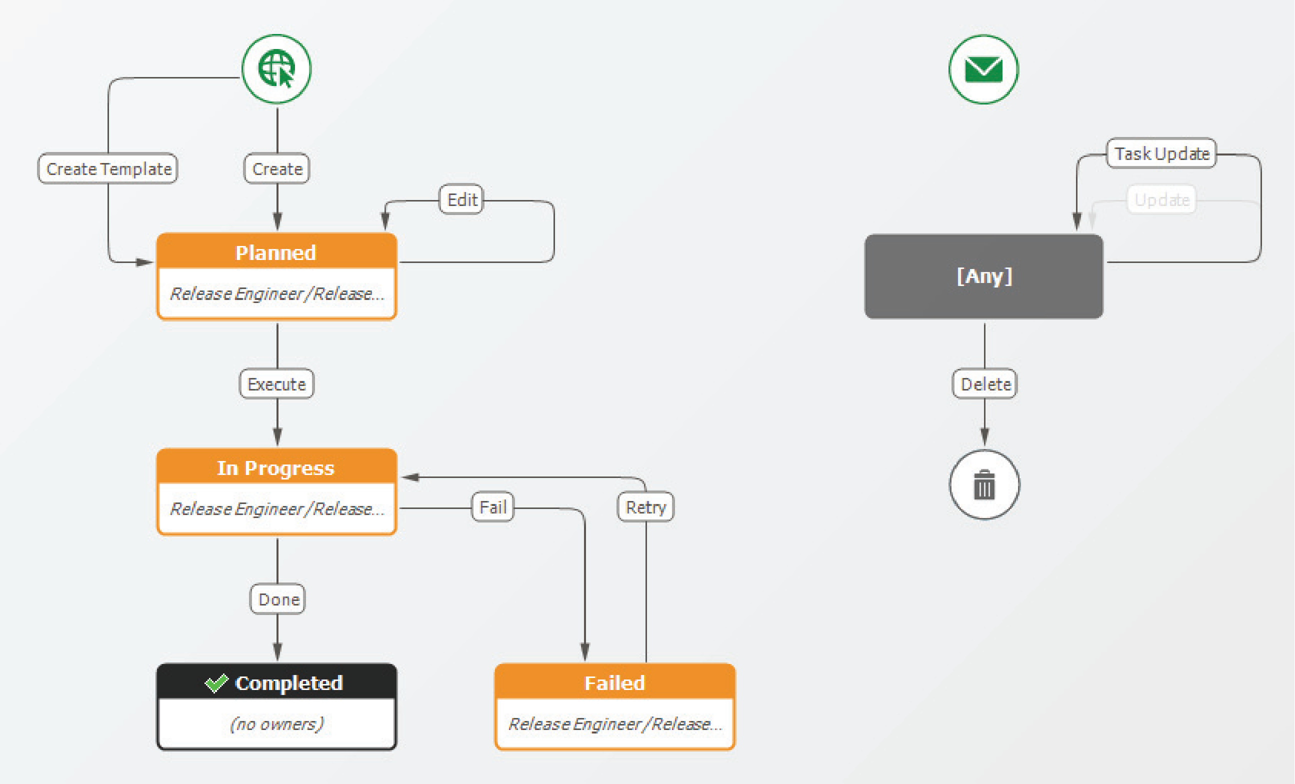
Deployment Task Workflows

The Deployment Task workflows in the default version of Serena Release Control are shown in the following figures.

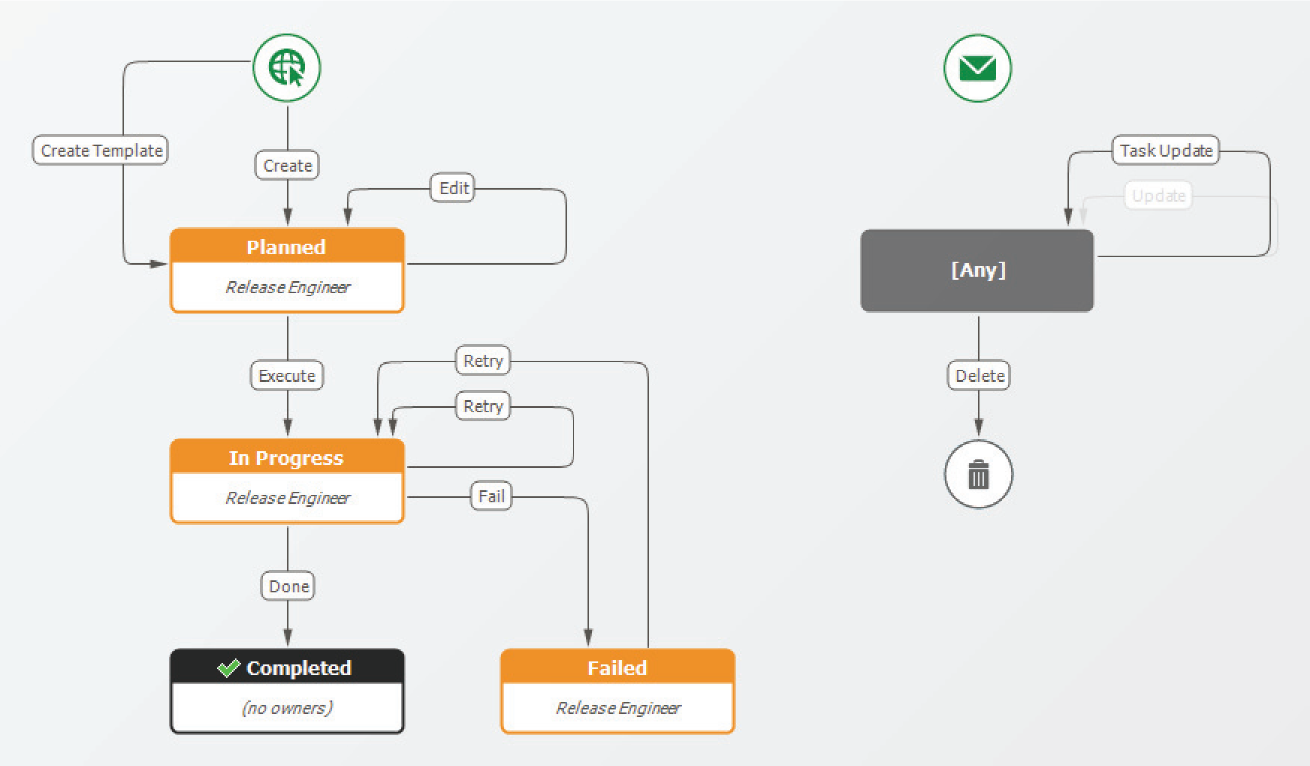
Automation Deployment Task Workflow



Manual Deployment Task Workflow

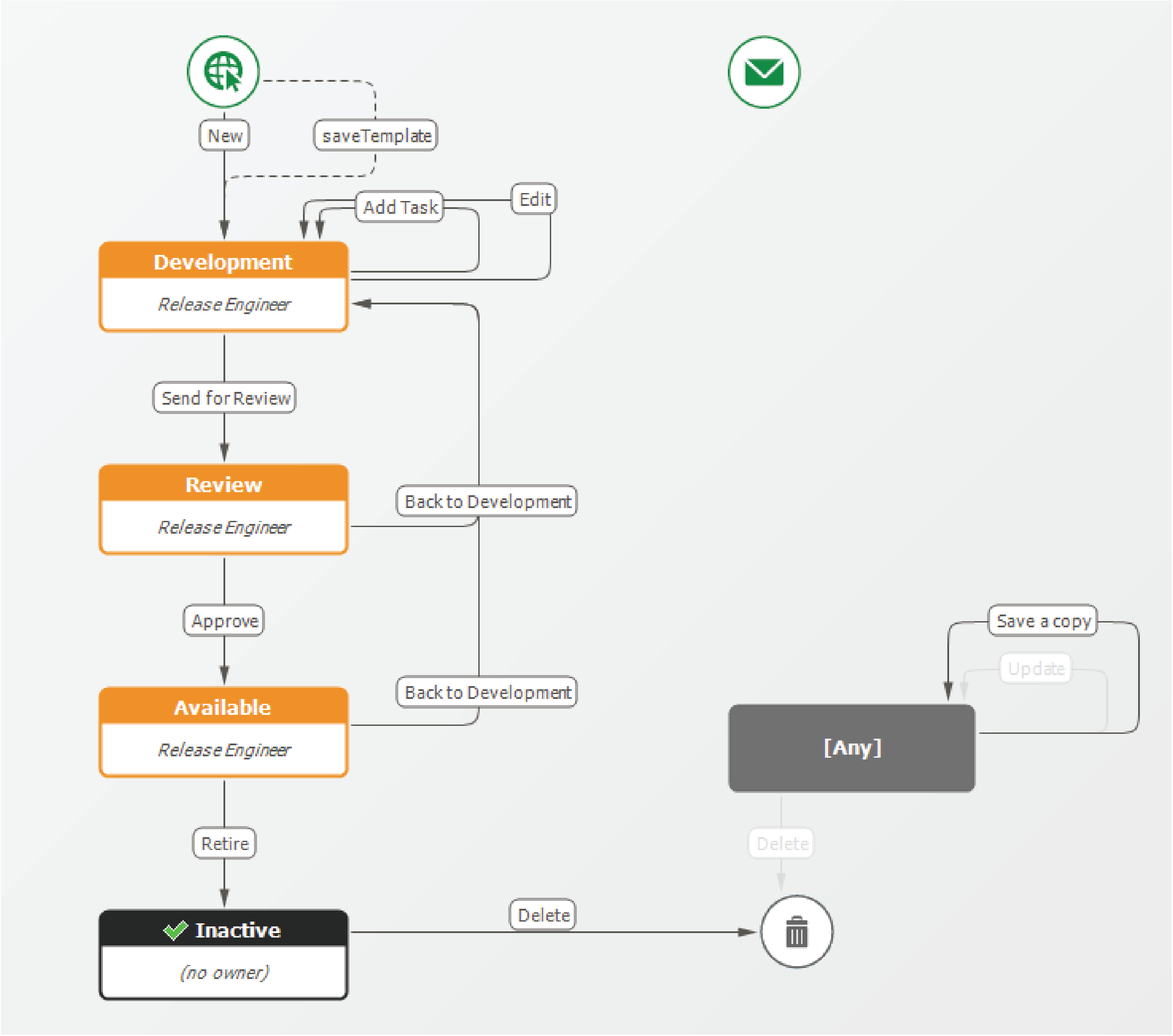


Vault Deployment Task Workflow



Deployment Process Template Workflow

The Deployment Process Template workflow in the default version of Serena Release Control is shown in the following figure.


New
saveTemplate
Add Task
Edit
Development
Release Engineer
Send for Review
Review
Release Engineer
Back to Development
Approve
Available
Release Engineer
Back to Development
Retire
Inactive
(no owner)
Delete
Save a copy
Update
[Any]
Delete

Index