SERENA®

RELEASE MANAGER 2.1

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: June 28, 2012

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 across mainframe and distributed systems with 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 document.

Procedures and figures in this documentation are examples based on the default implementation of Serena Release Manager. Examples are based on SBM 10.1 unless otherwise stated. The details may not match your implementation of Serena Release Manager exactly, but provide a reference to get you started with your implementation and use of Serena Release Manager.

Guide to Serena Release Manager Documentation

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

Manual or Tutorial	Description
Serena Release Manager Getting Started Guide	Describes how to use the default implementation of Serena Release Manager to manage application releases.
Serena Release Manager Installation and Configuration Guide	Describes how to install and configure the Serena Release Manager suite of products.
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 information on the Web services provided for Serena Release Manager.

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.
Serena Business Manager Composer Guide	Provides details on using SBM Composer to create the tables, fields, workflows, forms, 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.
Serena Business Manager Application Administrator Guide	Explains how to configure deployed applications. Instructions for managing projects, user and group accounts, and notifications are included.
Serena Business Manager Application Repository Guide	Provides information on using SBM Application Repository to deploy process apps to runtime environments and to promote configured applications from one environment to another.
Serena Business Manager System Administrator Guide	Provides information on administering the SBM Application Engine. Instructions for database utilities, system settings, and authentication are included.
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.
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.
Serena Release Automation Installation and Administration Guide	Describes how install and configure Serena Release Automation.
Serena Dimensions CM Installation Guides for Windows and *for UNIX*	Describes how install and configure Dimensions CM for the respective platforms.
Serena Dimensions CM Administrator's Guide	Describes how to administer the Dimensions CM tool.
Serena Dimensions CM and RM Installing the Serena License Manager	Describes installing the Windows version of SLM.
Serena Dimensions CM User's Guide	A user guide to the Dimensions CM Client Tools.
Serena Dimensions CM Process Modeling Guide	Describes how to configure the process model using the Administration Console.

Manual or Tutorial	Description
Serena ChangeMan ZMF Administrator's Guide	Describes ChangeMan ZMF features and functions with instructions for choosing options and configuring global and application administration parameters.
Serena ChangeMan ZMF Installation Guide	Provides step-by-step instructions for initial installation of ChangeMan ZMF. Assumes that no prior version is installed or that the installation will overlay the existing version.
SER10TY User's Guide	Gives instructions for applying licenses to enable ChangeMan ZMF and its selectable options.
Serena Orchestration Ops Installation and Configuration Guide	Provides information on installing and configuring Serena Service Manager.
Serena Service Manager ITIL Guide	Provides information about the Serena Service Manager default implementation user interface and is intended for end users.

Accessing the Documentation

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

When you click the **Help** link in Serena Release Control, the Serena Release Manager online Help appears. You can search, use the index, and view the glossary for information on installing, configuring, and using the product. From the Help you can also access the full set of documentation PDFs for the Serena Release Manager suite.

You can view and download Serena Release Manager readme files and PDFs from the Related Documentation page. Documentation for all supported releases for all of the Orchestrated ALM suites is available from the All Orchestrated ALM Documentation page.

Related Topics

- View the Online Help
- View the Related Documentation page
- View the All Orchestrated ALM Documentation page

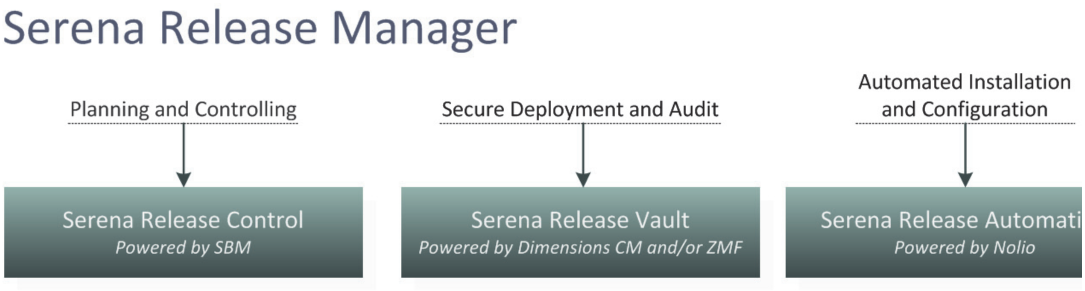
Chapter 1

Installation Prerequisites and Planning

This section tells you what you must 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.

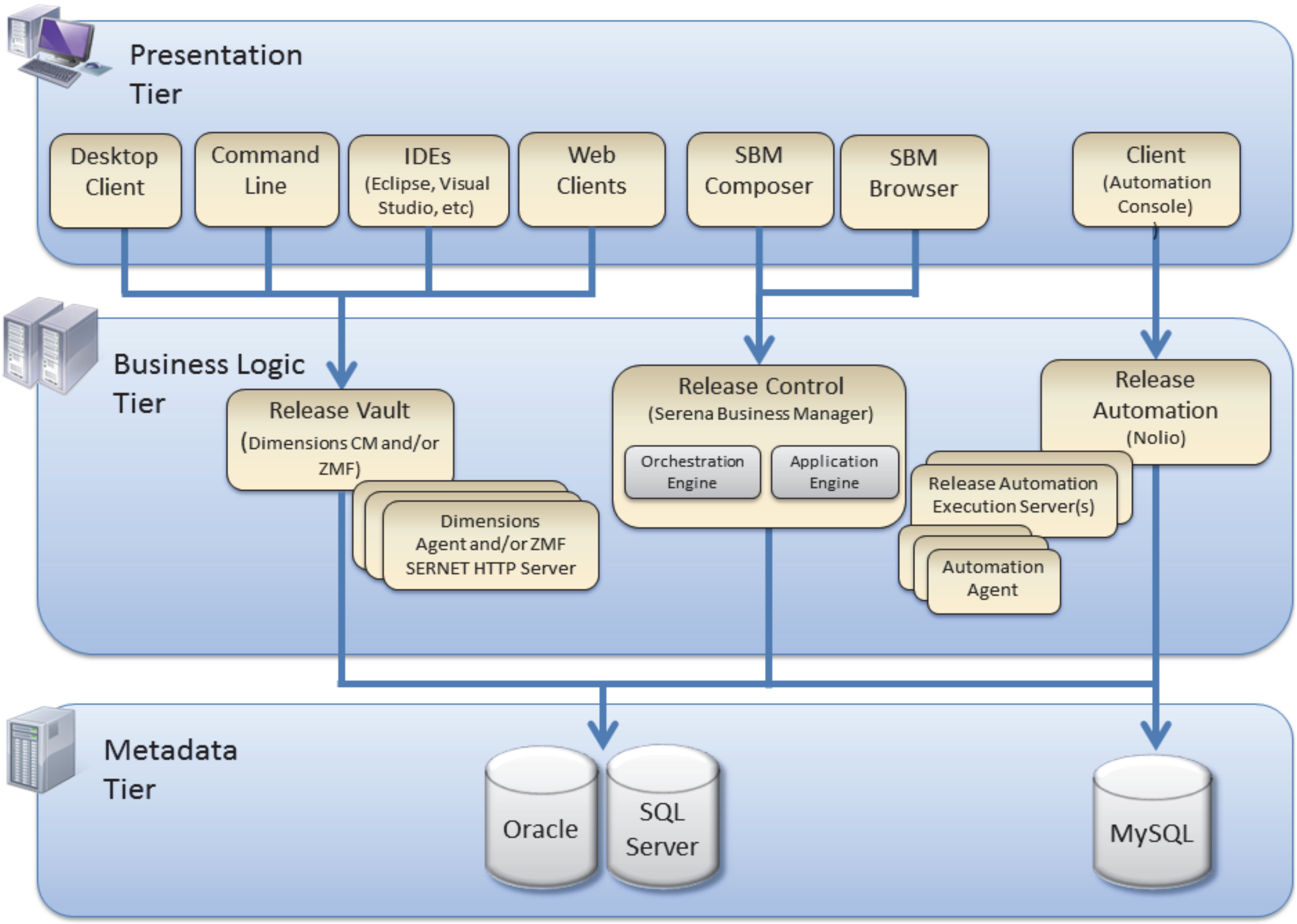


- **Serena Release Control**, powered by Serena Business Manager, helps you plan and control your application release processes across your enterprise, from definition to deployment.
- **Serena Release Vault**, powered by Dimensions CM for distributed systems and ChangeMan ZMF for IBM z/OS systems, ensures 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 Manager, see the Serena Release Manager Getting Started 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.


Presentation Tier
Desktop Client
Command Line
IDEs (Eclipse, Visual Studio, etc)
Web Clients
SBM Composer
SBM Browser
Client (Automation Console)
Business Logic Tier
Release Vault (Dimensions CM and/or ZMF)
Dimensions Agent and/or ZMF SERNET HTTP Server
Release Control (Serena Business Manager)
Orchestration Engine
Application Engine
Release Automation (Nolio)
Release Automation Execution Server(s)
Automation Agent
Metadata Tier
Oracle
SQL Server
MySQL

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.

- "Installation Prerequisites" on page 16
- "Database Requirements" on page 16
- "Software Compatibility Requirements" on page 17
- "System Requirements" on page 17
- "Planning Checklists and Worksheets" on page 17

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**

  Before you install Dimensions CM, please see the "Fresh Installations Checklist" section of the *Serena Dimensions CM Installation Guide* for your operating system.

- **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.

NOTE If you are using ChangeMan ZMF for your vault, you will need a supported installation of ChangeMan ZMF.

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.

- **Dimensions CM**

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

- **Serena Release Automation**

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

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 your version of Serena Release Manager on the `Serena Support Product Roadmap`.

NOTE This link requires that you are logged into the Customer Support website. If the link fails, enter `http://support.serena.com` in your Web browser, sign in, and select **Support > Product Roadmap** from the toolbar menu. On the high-level timeline, select **Release Manager** and then select your version.

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 Serena Release Automation 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.
Serena Release Automation	"System Requirements" in *Serena Release Automation Installation and Administration Guide.*
ChangeMan ZMF (Optional)	"System Requirements" in *Serena ChangeMan ZMF Installation Guide.*

Server Requirements

For optimal performance, Serena recommends that SBM and Dimensions CM be installed on separate physical servers.

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 database names and information.

For a checklist of the installation and configuration activities for Serena Release Manager, see Chapter 2, "Installation and Configuration Quickstart" on page 19.

Chapter 2

Installation and Configuration Quickstart

This section gives a concise high-level procedure for the installation and configuration activities.

Installation and Configuration Checklist

You can use this checklist to guide you through the installation and configuration, referring to more detailed procedures in the related sections as needed.

Step	Action
1: Installation	❑ Install the products you plan to use in the suite and apply licenses to the products as needed. See "Serena Release Manager Installation" on page 25 for details.
2: User Interface Shell	Put the user interface shell files into the SBM database to ensure Serena Release Control elements appear as designed. See "Installing the UI Shell files" on page 35 for details.
3: Process Apps	Import the Serena Release Control SBM solution and promote the snapshots of the process apps. ❑ Import the Serena Release Control solution, which contains a snapshot of each of the process apps. ❑ Create a new application server environment for the process apps if needed. ❑ Promote each of the snapshots, creating and selecting endpoints as needed. Use Security Token authentication for the endpoints. ❑ Configure hostname in each of the ZMF deployment task form Web page widgets. ❑ SBM 10.1.1.1 only: Select the SSO Authentication option in forms with RESTgrid widgets. ❑ Publish and deploy the process apps to upgrade the structure of the RESTGrid widgets. See "Importing and Promoting Serena Release Control" on page 36 for details.
4: Release Control Configuration	Configure required objects in Serena Release Control as follows: ❑ Create an administrative user and set all privileges for that user to the Serena Release Control objects, such as projects, reports, and auxiliary tables. For example, `rlmadmin`. ❑ Enable roles for Serena Release Control projects and verify that Serena Release Control is activated. ❑ Configure the Dashboard page as needed. See "Configuring Required Objects in Serena Release Control" on page 46 for details.

Step	**Action**
5: Client Connections	Configure the connections to each of the integrating clients using the Serena Release Manager Configurator. This information is stored in stored in the client connection properties files in the common Tomcat Web server `webapps\rlm\WEB-INF\classes` folder. For example, the Dimensions CM client connection file is `dm-client-connections.properties`. Select the corresponding tab and fill out the form to specify connection information for each of the following: □ ALF □ SBM □ BCR □ RFC □ Dimensions CM □ Release Automation □ ZMF See "Configuring Connections using the Release Manager Configurator" on page 49.
6: Dimensions CM Communication	Configure communication with Dimensions CM (Windows/UNIX systems release vault). For an overview of the integration, see "Dimensions CM Communication Configuration Overview" on page 56. **1** Configure Dimensions CM communication on the Dimensions CM server. □ Specify ALF event configuration information in the `dm.cfg` file. □ Specify selection criteria for the Dimensions CM events and objects by updating the ALF event configuration file, `ALF_EVENTS_CONFIG.XML`. Specify your Dimensions CM database name, project name, baseline type, and deploy event. See "Configuring Communication on the Dimensions CM Server" on page 56 for details. **2** Configure Dimensions CM communication in Serena Release Manager. □ Specify the connection information. You should have already done this through the Serena Release Manager Configurator Dimensions CM page. □ Specify Dimensions CM client-specific information in the common Tomcat Web server `webapps\rlm\WEB-INF\classes` folder `dm-client.properties` file. See "Configuring Dimensions CM Communication in Release Manager" on page 58 for details.

Step	Action
7: ChangeMan ZMF Communication	Configure communication with ChangeMan ZMF (z/OS systems release vault). For an overview of the integration, see "ZMF Communication Configuration Overview" on page 60. **1** Configure ChangeMan ZMF communication on the z/OS mainframe. ❑ Configure the NTFYURL; this is the URL Serena Release Manager uses to send information to SERNET through the server. ❑ Configure the SERNET HTTP server; this is the server Serena Release Manager uses to populate the UI widgets with ZMF information. ❑ Configure a proxy user ID for each mainframe host, or LPAR, that Serena Release Manager uses to log in to ChangeMan ZMF. ❑ Configure TSO user IDs that match the SBM user IDs. ❑ Configure approvers. See "Configuring ZMF Communication on the Mainframe" on page 60 for details. **2** Configure ChangeMan ZMF communication in Serena Release Manager. ❑ Specify the connection information. You should have already done this through the Serena Release Manager Configurator ZMF page. ❑ Specify ALF event manager information for ChangeMan ZMF in the Serena Release Manager common Tomcat Web server `webapps\almzmfalf\WEB-INF\conf` folder `alfzmf_resource.properties` file. ❑ Specify other ZMF client-specific information in the Serena Release Manager common Tomcat Web server `classes` folder `zmf-client.properties` file. See "Configuring ZMF Communication in Release Manager" on page 63 for details.

Step	Action
8: Release Automation Communication	Configure communication with Serena Release Automation. For an overview of the integration, see "Serena Release Automation Communication Configuration Overview" on page 66. **1** Configure Serena Release Automation communication on the Serena Release Automation server. ❑ Specify the Serena Release Automation server to notify when an event occurs in the Serena Release Automation `rest.integration.properties` file. If the file does not exist, create it. ❑ Update the Serena Release Automation environment notifications for each application to tell Serena Release Automation the events about which to notify Serena Release Manager. See "Configuring Communication on the Release Automation Server" on page 66 for details. **2** Configure Serena Release Automation communication in Serena Release Manager. ❑ Update the Serena Release Automation ALF sign-on credentials. You should have already done this through the Serena Release Manager Configurator ALF page. ❑ Specify the Serena Release Automation client-specific information in the Serena Release Manager common Tomcat Web server `classes` folder `nolio-client.properties` file. ❑ Specify the Serena Release Automation client query information in the Serena Release Manager common Tomcat Web server `classes` folder `nolio-client-queries.properties.properties` file. See "Configuring Release Automation Communication in Release Manager" on page 69 for details.
9: Integrating objects	Configure the integrating application objects that are accessed by or provided to Serena Release Manager during the release management process. These may include but aren't limited to the following: ❑ Release Control (SBM) users, general reports, Dashboard page, and notifications ❑ SBM projects (to provide BCRs or DCRs, for example) ❑ Applications and environments (servers) (specific to your organization; add to the respective auxiliary tables) ❑ Dimensions CM process model (GSL), projects and streams, baselines, and requests ❑ ChangeMan ZMF applications, sites, promotion levels, approvals, and change packages ❑ Release Automation environments, applications, processes, and servers ❑ Serena Service Manager projects and change requests (to provide RFCs, for example) See "Configuration and Administration of the Integrating Objects" on page 73 for details.

Step	Action
10: Provider properties	Configure the provider properties as needed in the common Tomcat Web server `webapps\rlm\WEB-INF\classes` folder. These may include but aren't limited to the following: ❑ Serena Service Manager (SSM) request for change (RFC) properties (for example: `itsm.properties` and `providers.properties`) ❑ Serena Business Manager (SBM) business change request (BCR) properties (for example: `bcr.properties` and `providers.properties`) ❑ Serena Business Manager (SBM) development change request (DCR) properties (for example: `sbm-issues.properties` and `providers.properties`) ❑ Dimensions CM deployment unit (DU) properties (for example: `dm_qlarius.properties` and `providers.properties`) ❑ ChangeMan ZMF deployment unit (DU) properties (for example: `zmf_packages.properties` and `providers.properties`) See "Provider Configuration" on page 93 for details.
11: Customization	Complete customization and additional configuration as needed. See "Serena Release Manager Customization" on page 115 for details.

IMPORTANT! You must restart the appropriate services after updating the properties files, such as Serena Common Tomcat, Serena Common JBOSS, and IIS Admin Services.

TIP Clear your browser cache if Serena Release Control has been run from your browser on this machine before to ensure that your user interface is displaying the most current shell elements.

Chapter 3

Serena Release Manager Installation

This section 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 section. 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 a Serena release vault, such as Dimensions CM or ChangeMan ZMF, are required components of the default implementation of Serena Release Manager.

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 Dimensions CM. See "Installing Serena License Manager" on page 26.
2. Ensure that Serena Business Manager is installed as required. Serena Business Manager is the platform on which Serena Release Manager runs. See "Installing Serena Business Manager" on page 27.
3. Ensure that Dimensions CM is installed as required. Dimensions CM provides a release vault that enables you to securely deploy and audit your releases that run on distributed environments such as Windows and UNIX. See "Installing Serena Release Automation" on page 31.
4. 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 28.
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 31.
6. Ensure that other providers that you plan to use are installed as needed. See "Installing Other Integrating Serena Products" on page 31.

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 Manager ready to use.

Installing Serena License Manager

You must install Serena License Manager to license and run Serena Business Manager and Dimensions CM.

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. However, you must 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, apply the licenses as needed. See "Applying Licenses" on page 32.

Please continue to "Installing Serena Business Manager" on page 27.

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. However, you must 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 28.

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 Dimensions CM

Dimensions CM must be installed to use the deployment capabilities of Serena Release Manager. You can use both Dimensions CM and ChangeMan ZMF as release vaults from within Serena Release Control.

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 must 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 31.

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.

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 Control

Serena Release Control provides the user interface and the infrastructure that integrates the components of Serena Release Manager.

Existing Serena Release Control Systems

If you already have an installation of Serena Release Control and are upgrading, see "Serena Release Manager Upgrade" on page 107.

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 27.

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 4, "System Activation and Configuration" on page 33.

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 8, "Configuring Release Manager to Use a Different Port" on page 143 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 Agreement** page appears.

3. Confirm and click **Next**.

   The **Destination Folder** page appears.

4. Optionally click **Change** to change the target location for the installation.

5. Click **Next**.

   The **Setup Type** page appears.

6. Select **Complete** or **Custom** and click **Next**.

   For a custom installation:

   a. Click to select or deselect components to install.

      - ALM Foundation Services

- Release Control Process Application
- Release Manager Configurator SSO Support

b Click **Next** again.

If you selected **Release Manager Configurator SSO Support**, the **Serena Single Sign On** page appears.

c Specify the host name and port for an existing installation of a Serena SSO server.

d If you want the connection to use HTTPS, select **Secure (HTTPS) Connection**.

e Click **Next**.

The **Ready to Install the Program** page appears.

7 Click **Install**.

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

The installer displays whether a valid version of SBM, Dimensions CM, the Serena Release Control common Apache Tomcat Web server, and the Serena Reporting Server are installed. It installs the common Tomcat Web server if needed.

> **NOTE** The installer is shared with other Orchestrated ALM suites so some products may appear in this list as not required. Some of the products may be required to be installed separately, such as Dimensions CM.

8 Optionally select **Show the Windows Installer log** and click **Finish**.

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`.

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 `war` files)

The installer does the following automatically:

- Copies the `war` files to the Tomcat Web server and restarts Tomcat.
- Puts the user interface shell files in the appropriate SBM folder.

Related Topics

- Chapter 4, "System Activation and Configuration" on page 33
- Chapter 7, "Serena Release Manager Upgrade" on page 107
- Chapter 9, "Installer Log Files" on page 161

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 must 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.

If you plan to use ChangeMan ZMF as one of your release vaults, please continue to "Installing Other Integrating Serena Products" on page 31. Otherwise, please continue to Chapter 4, "System Activation and Configuration" on page 33.

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

Documentation References

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

Installing Other Integrating Serena Products

ChangeMan ZMF can optionally be used as the release vault for your z/OS mainframe applications. You can use both Dimensions CM and ChangeMan ZMF as release vaults from within Serena Release Control.

Serena Service Manager can optionally be used as the provider for your RFCs and other change request associations.

Documentation References

- Complete documentation on installing ChangeMan ZMF is in the *ChangeMan ZMF Installation Guide.*
- Complete documentation on installing SSM is in the *Serena Service Manager Installation and Configuration Guide.*

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.
- **Dimensions CM**: Use Serena License Manager to apply licenses for Dimensions CM.
- **Serena ChangeMan ZMF**: Use Serena SER10TY to apply licenses for ChangeMan ZMF.
- **Serena Release Automation**: Enter license keys for Serena Release Automation from the Help menu of Serena Release Automation.

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 Dimensions CM is in the *Serena Dimensions CM Administrator's Guide* in "Administering Your Licenses and the License Server".
- Complete documentation on applying licenses for ChangeMan ZMF is in the *SER10TY User's Guide*.
- Complete documentation on applying licenses for Serena Release Automation is in the *Serena Release Automation Installation and Administration Guide* in "Updating Serena Release Automation License".

Chapter 4

System Activation and Configuration

This section 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 Manager. 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

In the system configuration, you must:

- Put the user interface templates into your SBM database.
- Put the Serena Release Manager process apps into your SBM application engine environment.
- Map the end points of the Serena Release Manager Web services to the server where the Serena Common Tomcat server is running.
- Configure required objects in Serena Release Control, such as users, privileges, roles, and the Dashboard page.
- Configure the connection information and other details for the integrating clients through the Serena Release Manager Configurator and various configuration files.

You must complete the system configuration for each of the products in the Serena Release Manager suite that you plan to use for your release management.

- For Serena Release Control:
  - "Installing the UI Shell files" on page 35
  - "Importing and Promoting Serena Release Control" on page 36
  - "Configuring Required Objects in Serena Release Control" on page 46
  - "Configuring Connections using the Release Manager Configurator" on page 49
- For Dimensions CM:
  - "Dimensions CM Communication Configuration Overview" on page 56
  - "Configuring Communication on the Dimensions CM Server" on page 56
  - "Configuring Dimensions CM Communication in Release Manager" on page 58
- For ChangeMan ZMF:
  - "ZMF Communication Configuration Overview" on page 60
  - "Configuring ZMF Communication on the Mainframe" on page 60
  - "Configuring ZMF Communication in Release Manager" on page 63
- For Serena Release Automation:
  - "Serena Release Automation Communication Configuration Overview" on page 66
  - "Configuring Communication on the Release Automation Server" on page 66
  - "Configuring Release Automation Communication in Release Manager" on page 69

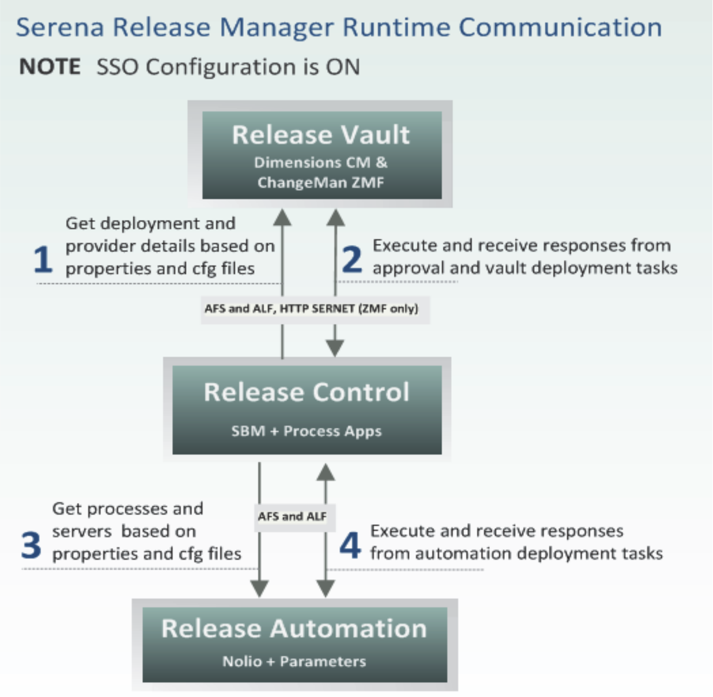
Related Topics

- "Serena Release Manager Runtime Communication" on page 35
- Chapter 1, "Installation Prerequisites and Planning" on page 13
- Chapter 2, "Installation and Configuration Quickstart" on page 19
- Chapter 3, "Serena Release Manager Installation" on page 25
- Chapter 5, "Configuration and Administration of the Integrating Objects" on page 73

Serena Release Manager Runtime Communication

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.

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
Release Vault
Dimensions CM & ChangeMan ZMF
1 Get deployment and provider details based on properties and cfg files
2 Execute and receive responses from approval and vault deployment tasks
AFS and ALF, HTTP SERNET (ZMF only)
Release Control
SBM + Process Apps
3 Get processes and servers based on properties and cfg files
AFS and ALF
4 Execute and receive responses from automation deployment tasks
Release Automation
Nolio + Parameters

All communication for Serena Release Manager goes through the Serena Release Manager Web services and supporting programmatic layers, collectively referred to as ALM Foundation Services (AFS), and SBM, with support from Application Lifecycle Framework (ALF) events for Dimensions CM, ChangeMan ZMF, and Serena Release Automation communication.

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

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

Installing the UI Shell files

To fully install the UI shell files, you must put the user interface shell files into the SBM database to ensure Serena Release Control elements appear as designed. You should also clear and refresh your Web browser cache to ensure any conflicting information saved in your Web browser is removed.

To install the UI shell files:

1. From SBM System Administrator, select **File | Put Files Into Database**. Confirm when prompted.

   This puts the UI shell files into the SBM database.

NOTE This step should be done before you promote the snapshots so that SBM will promote and deploy the correct template files.

Importing and Promoting Serena Release Control

Serena Release Control includes a bundle of process apps that run in SBM. To bring those process apps into SBM and activate them, you must do the following:

- Import the Serena Release Control solution, which contains a snapshot of each of the process apps.
- Create a new application engine server environment for the process apps if needed.
- Promote each of the snapshots into your application engine server environment, creating and selecting endpoints as needed. Use Security Token authentication for the endpoints.
- If using ChangeMan ZMF: Configure hostname in each of the ZMF deployment task form Web page widgets.
- SBM 10.1.1.1 only: Select the SSO Authentication option in forms with RESTgrid widgets.
- Publish and deploy the process apps to upgrade the structure of the RESTGrid widgets.

NOTE It is a prerequisite for SBM to be installed with Single Sign-on (SSO). See "Installing Serena Business Manager" on page 27. For the communication between SBM and Dimensions CM to work correctly through SSO, the products must share an SSO server and you must have same user ID for both SBM and Dimensions CM.

Related Topics

- "Importing the Serena Release Control Solution" on page 37
- "Creating an Environment for Serena Release Manager" on page 37
- "Promoting the Snapshots" on page 40
- "Setting Authentication for Added Endpoints" on page 41
- "Promoting the Snapshots Again to Resolve References" on page 42
- "Configuring the ChangeMan ZMF Web Page Widgets" on page 42
- "SBM 10.1.1.1 Only: Configuring SSO in RESTgrid Widgets" on page 44
- "Publishing and Deploying the Process Apps" on page 45

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 included to guide you through the Serena Release Manager-specific configuration process. For complete documentation on process apps, see the *Serena Business Manager Application Repository Guide.*

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

1. Select **Start | All Programs | Serena | Serena Business Manager | SBM Application Repository**.
2. Enter your login details.
3. In the navigation pane, click **Solutions**.
4. The Serena Release Manager solution pack is listed. For example:

   ```
   RLM_Solution_Pack 2.1.0.32
   ```

5. 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.

6. Click **OK**.

   The solution is now listed under **Imported solutions**.

7. Select the solution name and then click **Open Snapshots** to verify that the following process app snapshots are listed:
   - Application Release
   - Deployment
   - Environment
   - Release Package
   - ReleaseTemplate
   - Release Train
   - RLM_AUX

Creating an Environment for Serena Release Manager

You must create an SBM application engine server environment for your Serena Release Control process applications unless you are promoting them into an existing environment.

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

TIP Understanding SBM: An environment describes the runtime server to which you deploy process apps. An environment requires an SBM Application Engine server.

After you create the environment, you must add other target servers (such as a BPEL engine or Event Manager server) as well as any Web service end points that are required to support the process app.

If you are putting your Serena Release Control process apps into an existing environment, continue to "Promoting the Snapshots" on page 40.

Create the environment and supporting target servers as follows:

- Creating the Environment
- Specifying the BPEL Server for SBM Orchestrations
- Specifying the SBM System Event Manager

Creating the Environment

To create the environment:

1 From the SBM Application Repository 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. For example, RLM Environment.

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

c Under **Application Engine Server**, enter a name and description. For example: RLM Application Engine Server.

d In the URL, change the server to the host name for your application engine server and specify the port number for the server. For example:

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

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

5 Click **OK**.

Specifying the BPEL Server for SBM Orchestrations

For new SBM environments, you must specify the target server for the SBM orchestrations, or BPEL engine.

To set up the target server for SBM orchestrations:

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 Server.

c Enter the URL using the host name for your orchestration engine server as the hostname.

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

```
http://sbmoehost: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**.

Specifying the SBM System Event Manager

For new SBM environments, you must specify the target server for the SBM system event manager.

To set up the SBM system 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 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 the host name for your system event server as the hostname. For example:

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

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

e Click **OK**.

NOTES

- You do not need to create a target server for the SBM Common Services. This target server is created automatically when a process app snapshot associated with SSM or SLA is promoted.
- SBM Common Services are not the same as the Serena Common Tomcat Services used by Serena Release Manager.

Promoting the Snapshots

You must promote the Serena Release Manager snapshots and define the destination endpoints for each. When promoting the snapshots, make sure to create and select the endpoints as needed, and make sure the endpoints are authenticated with Security Token.

Promote the snapshots in any order.

1. Release Train
2. Application Release
3. Release Package
4. RLM_AUX
5. Deployment
6. Environment
7. ReleaseTemplate

To promote a snapshot:

1. In the SBM Application Repository 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 that you have not yet promoted 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 Manager.
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, or 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 it is being created 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>
```

where `<tomcat port>` is the port under which you are running the Serena Common Tomcat Web server.

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

Example endpoints with the default port number 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 **Security Token**.

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.

10 Click **Done**.

The **Summary** page appears.

11 Click **Promote**.

The **Promotion Started** page appears.

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

13 Repeat the procedure for each snapshot.

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

Setting Authentication for Added Endpoints

SBM automatically generates endpoints for selections that use RESTgrid widgets. You must update each of these to use Security Token authentication, so that SSO is used for these widgets.

To update the automatically-generated endpoints:

1 Select the RLM environment and then select the **Endpoints** tab.

In the resulting list of endpoints you'll see endpoints with default names that have a prefix of `<hostname><port>-`. The hostname and port are those specified in the related endpoints during the promotion of the snapshots.

2. Select an endpoint and click **Edit**.
3. In the **Authentication** field, select **Security Token**.
4. Click **OK**.
5. Repeat for each of the automatically-generated endpoints.

Promoting the Snapshots Again to Resolve References

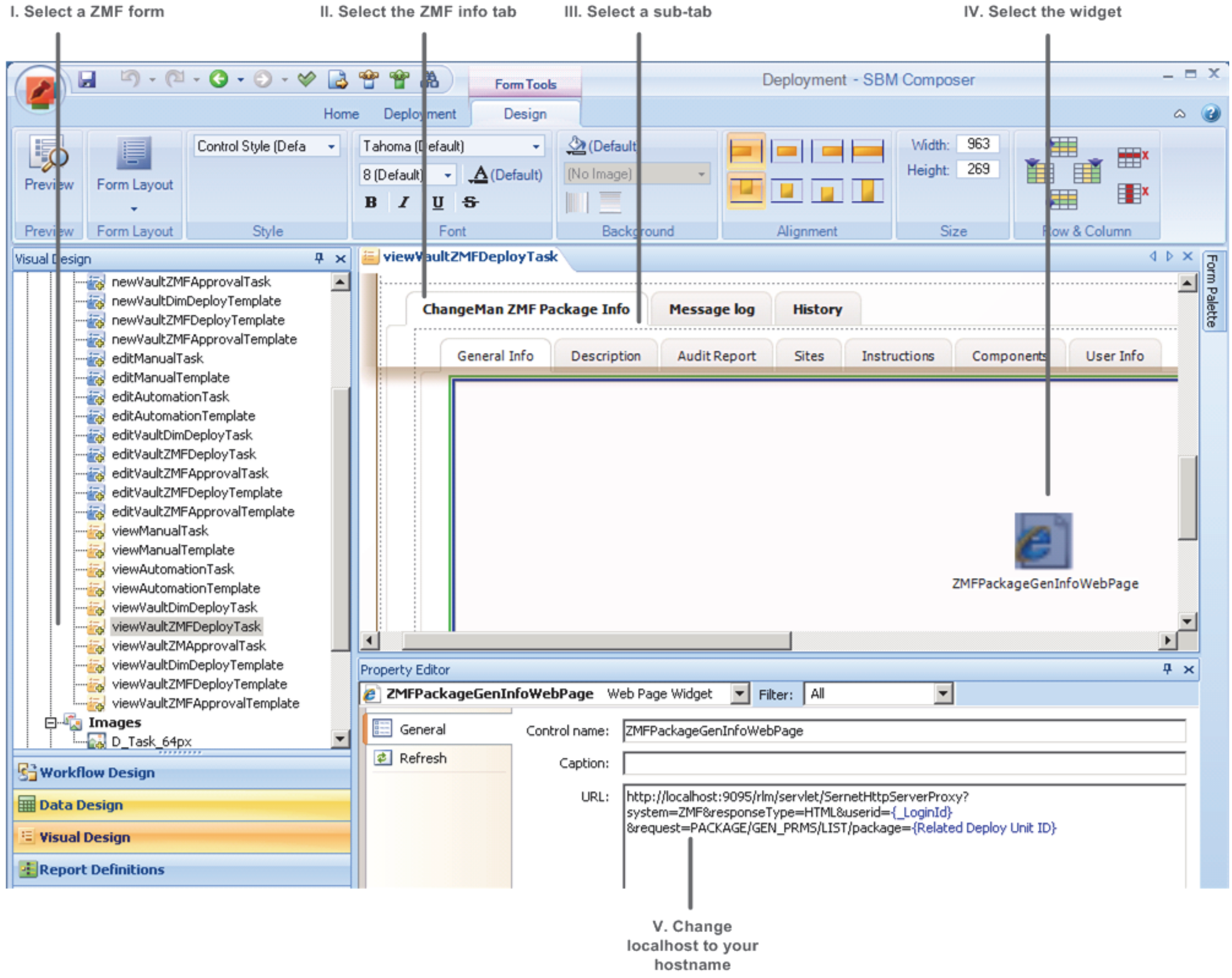
To resolve warnings for unresolved references in the snapshots, you must promote each of them again. Follow the steps in "Promoting the Snapshots" on page 40, except you won't need to create and choose endpoints this time.

Configuring the ChangeMan ZMF Web Page Widgets

If you are using ChangeMan ZMF for z/OS mainframe release management, you must configure your Serena Release Manager hostname in the ZMF deployment task forms that display ZMF change package information. Otherwise, the ZMF information will not appear when you view ZMF deployment tasks. Change the forms in the SBM Composer as shown in the following figure and procedure.

⚠ **CAUTION!** The following procedure assumes that you have just promoted each of the process apps to the repository. If you have checked out components of the process apps and made changes locally since promoting the process apps, please check in your changes before continuing to ensure you don't overwrite any of your changes.

ZMF Widget Form Updates



I. Select a ZMF form

II. Select the ZMF info tab

III. Select a sub-tab

IV. Select the widget

V. Change localhost to your hostname

Property Editor

ZMFPackageGenInfoWebPage Web Page Widget Filter: All

Control name: ZMFPackageGenInfoWebPage

Caption:

URL: http://localhost:9095/rlm/servlet/SernetHttpServerProxy?system=ZMF&responseType=HTML&userid={_LoginId}&request=PACKAGE/GEN_PRMS/LIST/package={Related Deploy Unit ID}

To update the hostname in the ZMF form widgets:

1 In SBM Composer, open the **Deployment** process app from the **Repository**. Choose **Overwrite** if prompted.

2 In **Visual Design**, check out and update the `viewVaultZMFDeployTask` and `viewVaultZMFApprovalTask` forms as follows:

   a Select the form and then select the ZMF Info tab, such as `ChangeManZMFPackageInfo`.

   b For each of the sub-tabs, General, Description, AuditReport, Sites, Instructions, Components, and User Info, select the Web page widget, such as `ZMFPackageGenInfoWebPage`.

   c In the Property Editor, in the URL for the widget, change `localhost` to your server hostname.

3 Save and check in the forms.

TIP If you are not using ChangeMan ZMF in your implementation, consider hiding the ZMF-specific form elements using SBM Composer.

Related Topics

- Chapter 4, "ZMF Communication Configuration Overview" on page 60

SBM 10.1.1.1 Only: Configuring SSO in RESTgrid Widgets

If you are using SBM 10.1.1.1, you must select the **Use SSO authentication** check box for each RESTgrid widget to enable the security token to be included in the URL. If you are using SBM 10.1, this is not required.

NOTE The behavior of the SSO authentication in relation to RESTgrid widgets differs between SBM 10.1 and SBM 10.1.1.1.

For SBM 10.1.1.1, if the **Use SSO authentication** check box is selected, SBM adds the SSO security token to the HTTP header in the HTTP RESTgrid request during the Web services call. If it is deselected, the security token isn't added to HTTP header.

For SBM 10.1, the SSO security token is added to the HTTP header if SSO is enabled for SBM regardless of whether the **Use SSO authentication** check box is selected.

The form controls in which you must update the check box are shown in the following table. This list is based on the default implementation of Serena Release Manager and may not be an exhaustive list in your implementation, so it is good practice to look at each form to see if it uses RESTgrid widgets.

Process App	Forms	Controls
Release Package	addDCRProjects	listDCRProjects
	addDepUnit	listDimCMBaseline, listZMFPackages
	addDevChgRequest	listRequests
	addDimCM_Projects	listDimCMProjects
	addZMF_Projects	listZMFProjects
	createRelPackage, createRPfromAR	listDCRProjects, listDimCMProjects, listZMFProjects
Application Release	addBusinessChangeRequest	RESTGridWidget
Release Train	addRFC	listRfcRequests

Process App	Forms	Controls
Deployment	newAutomationTask, newAutomationTemplate, editAutomationTask, editAutomationTemplate	gridApps, gridEnv, gridProc
	newVaultDimDeployTask, editVaultDimDeployTask	gridDepUnits, gridDepStages, gridDepAreas
	newVaultZMFDeployTask, editVaultZMFDeployTask	gridDepUnits, gridDeploymentSites, gridPromotionAreas
	newVaultZMFApprovalTask, editVaultZMFApprovalTask	gridDepUnits, gridApprovers

Related Topics

- Chapter 9, "Matches Not Found for Selections" on page 166
- Chapter 8, "Configuring Release Manager to Use a Different Port" on page 143

Publishing and Deploying the Process Apps

After promoting the snapshots twice in SBM Application Repository and completing any form configuration necessary, you must then publish and deploy the process apps from SBM Composer. This is required to publish any changes and upgrade the structure of the RESTGrid widgets.

⚠ **CAUTION!** The following procedure assumes that you have checked in any changes to the process apps. Please check in if needed before continuing to ensure you don't overwrite any of your changes.

To publish and deploy the process apps in SBM:

1. In SBM Composer, do the following:
   a. Select **Open** from the Composer menu and open a process app from the repository.
   b. Click **Publish**.
   c. Close the process app; check in to the repository when prompted.
   d. Repeat for each process app.
2. In the SBM Application Repository, do the following:
   a. Click **Process Apps** in the navigation pane and then select a process app.
   b. Click **Deploy**.
   c. Map endpoints as needed as you did when you promoted the snapshots.
   d. Repeat for each process app.

Documentation References

- Complete documentation on promoting and deploying process apps in SBM Application Repository is in the *Serena Business Manager Application Repository Guide* in "Promoting Process Apps" and "Deploying Process Apps".

- Complete documentation on publishing process apps in SBM Composer is in the *Serena Business Manager Composer Guide* in "Working with Process Apps".

Configuring Required Objects in Serena Release Control

Before you can use Serena Release Manager, you must configure required objects in Serena Release Control as follows:

1. Create an administrative user.
2. Set all privileges for the administrative user to the Serena Release Control objects, such as projects, reports, and auxiliary tables. For example, `rlmadmin`.
3. Enable roles for Serena Release Control projects and verify that Serena Release Control is activated.
4. Configure the Dashboard page as needed.

NOTE Terminology and user interface names and elements may differ from release to release of SBM. Make sure you are using the SBM documentation that matches your version of SBM. Examples in this document are based on SBM 10.1.

Related Topics

- "Creating an Administrative User" on page 46
- "Configuring the Administrative User Privileges" on page 47
- "Enabling Serena Release Control Project Roles" on page 48
- "Configuring the Dashboard Page" on page 49

Creating an Administrative User

The Serena Release Manager administrative user is used to execute many Serena Release Manager background activities and is used for communication with integrating products.

If you don’t already have an administrative user that you want to use with Serena Release Manager, create one now using SBM Application Administrator.

To create the administrative user in SBM:

1. Navigate to SBM Application Administrator and its **Administrator portal** as follows:
   a. Login to the SBM User Workspace as an administrative user who has privileges to update users.
   b. Click the **Administrator** icon in the SBM User Workspace.
2. In the **Administrator portal**, click the **Users** icon.

3 Create the user according to SBM Application Administrator documentation.

TIP A quick way to create an administrative user is to select an existing administrative user, such as `admin`, and copy that user to a new user name, such as `rlmadmin`. Edit `rlmadmin` to give the additional unique privileges needed for Serena Release Manager.

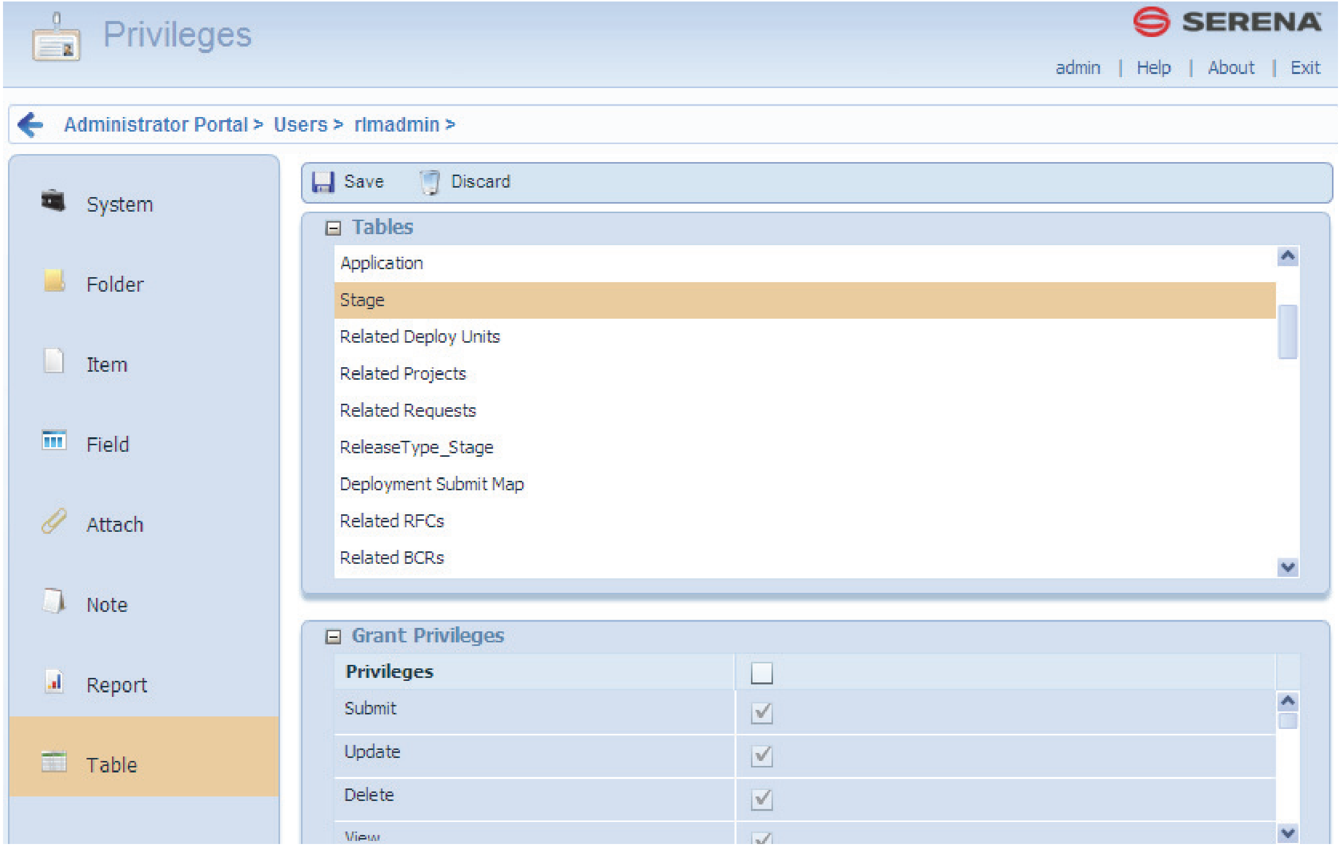
NOTE A matching administrative user ID must be set up in Dimensions CM, because single sign-on is required for the communication to work. A similar TSO user ID is required if you are implementing ChangeMan ZMF with Serena Release Manager.

Configuring the Administrative User Privileges

In SBM Application 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.



Privileges	□
Submit	☑
Update	☑
Delete	☑
View	☑

Documentation References

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

Enabling Serena Release Control Project Roles

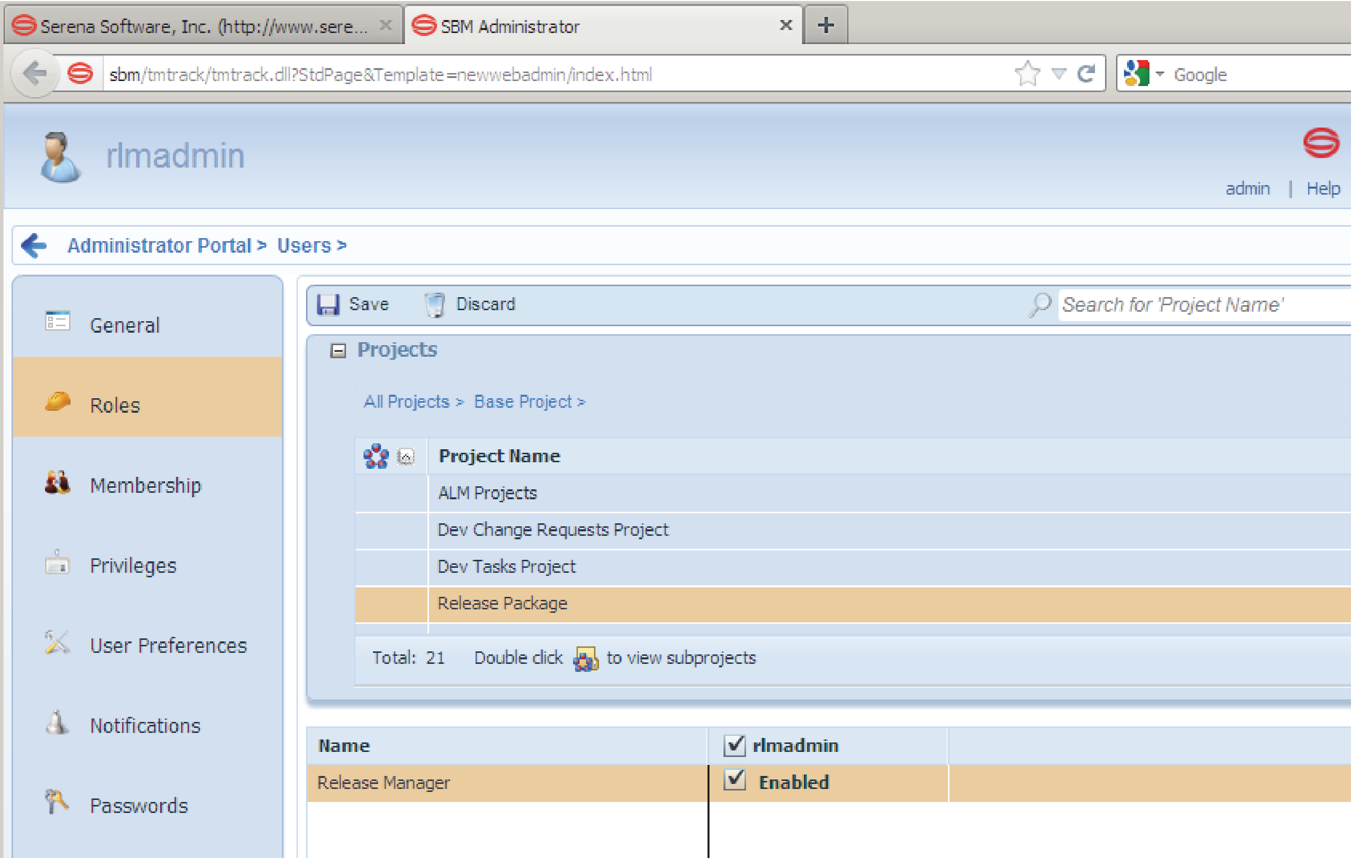
The administrative user must be enabled for all the roles that may have ownership within each of the Serena Release Control projects.

To enable the role for each Serena Release Control project:

1. In SBM Application Administrator, edit the administrative user. For example, `rlmadmin`.
2. In the navigation pane, select **Roles**.
3. In the Project tree, expand the sub-projects.
4. For each project and role that is not enabled for this user, select the user name beside the role which you want to enable for this user. The user name and the **Enabled** selection boxes are shown. Select **Enabled** if it is not already selected.
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.


Serena Software, Inc. (http://www.sere...
SBM Administrator
sbm/tmtrack/tmtrack.dll?StdPage&Template=newwebadmin/index.html
Google
rlmadmin
admin | Help
Administrator Portal > Users >
General
Roles
Membership
Privileges
User Preferences
Notifications
Passwords
Save
Discard
Search for 'Project Name'
Projects
All Projects > Base Project >

Project Name
ALM Projects
Dev Change Requests Project
Dev Tasks Project
Release Package

Total: 21 Double click to view subprojects

Name	☑ rlmadmin
Release Manager	☑ Enabled

Documentation References

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

Configuring the Dashboard Page

To configure your Dashboard page to show the multi-view dashboard report, configure the dashboard as follows.

1. In SBM Application Administrator, from the **Administrator portal** click the **Users** icon.
2. In the list of users, select the **Login ID** for the user you want to configure and click **Details**.
3. In the navigation pane, click **User Preferences**.

   The **Content** page appears.
4. In the **Home Page** section:
   - **a** Deselect **Show Launch Page**.
   - **b** In the **Applications** field, select **ReleaseTrain**.
   - **c** In the **Home Page Report** field, select **Base Project: Dashboard**.
5. Click **Save**.

TIP Clear your browser cache if Serena Release Control has been run from your browser on this machine before to ensure that your user interface is displaying the most current shell elements.

Documentation References

Complete documentation on managing user preferences in SBM is in the *Serena Business Manager Application Administrator Guide* in "User Preference Settings".

Configuring Connections using the Release Manager Configurator

The Serena Release Manager Configurator provides a graphical interface in which you configure the connections to clients that participate in your Serena Release Manager solution. You can run the Serena Release Manager Configurator anytime after initial configuration to re-configure your implementation.

If you run the Serena Release Manager Configurator, the client connection properties files are configured through your entries in the Serena Release Manager Configurator forms. Properties files with names that include `-connection.properties` are configured through the configurator. The rest of the files must be manually configured.

Related Topics

- "Configuring Access to the Release Manager Configurator" on page 50
- "Invoking Serena Release Manager Configurator" on page 50
- "Entering Information in Serena Release Manager Configurator" on page 51

Configuring Access to the Release Manager Configurator

Before anyone can log into the Serena Release Manager Configurator, you must add user IDs to the authorized list in the `rlm.properties` file. Typically this list would include only your Serena Release Manager administrators or users with similar roles.

To update the authorized list of users:

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 `rlm.properties` file.

3 Set the properties for the connection as follows:

```
rlm.config.service.authdUsers=<list of user IDs>
```

where `<list of user IDs>` is a list of Serena Release Manager (SBM) user IDs delimited by commas. For example:

```
rlm.config.service.authdUsers=admin,rlmadmin
```

4 Restart the Serena common Tomcat service.

Invoking Serena Release Manager Configurator

You should have received a Serena Release Manager URL, username, and password from your Serena Release Manager administrator.

To invoke and log in to Serena Release Manager Configurator:

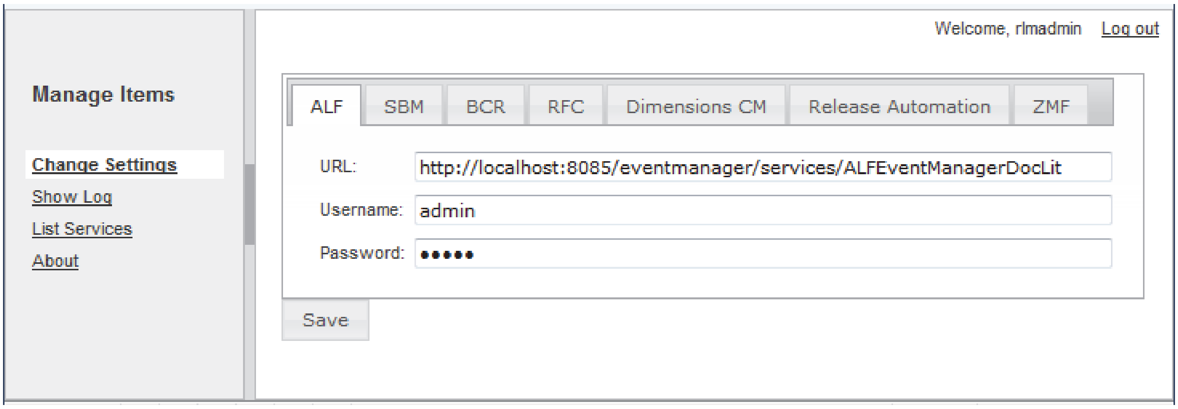
1 Enter the Serena Release Manager Configurator URL you received from your Serena Release Manager administrator. It will look similar to the following:

```
http://rlmhost:9095/rlm
```

The login dialog box appears unless you are already logged in on a shared single sign-on client.

2 Enter your username and password and then click Log In.

Upon successful login, your Serena Release Manager Configurator appears as shown in the following figure.



URL: http://localhost:8085/eventmanager/services/ALFEventManagerDocLit

Username: admin

Password: ●●●●●

If your login is not successful, please contact your Serena Release Manager administrator.

Entering Information in Serena Release Manager Configurator

You can use the Serena Release Manager Configurator to configure the client connections.

To add or update the client connection properties:

1. In the Serena Release Manager Configurator, select the **Change Settings** page.
2. Enter the connection information in the forms for each of the following tabs:
   - ALF: Application Lifecycle Foundation for communication between systems
   - SBM: Serena Business Manager for the core system as well as deployment change request (DCR) or other provider
   - BCR: Business Change Request provider (default SBM Incidents)
   - RFC: Request for Change provider (default SSM, Serena Service Manager)
   - Dimensions CM: Release vault, deployment unit (DU) and optionally DCR provider
   - Release Automation: powered by Nolio, for installing and configuring into target environments
   - ZMF: ChangeMan ZMF: Release vault and DU provider

   The information is saved in separate files in the Serena Release Manager common Tomcat Web server `classes` folder.
3. Restart the Serena Common JBOSS and IIS Admin Service services.
4. Restart the Serena common Tomcat service.

Related Topics

- ALF Client Connection Information
- "SBM Application Connection Information" on page 52
- "BCR Provider Connection Information" on page 52
- "RFC Provider Connection Information" on page 53
- "Dimensions CM Client Connection Information" on page 53
- "Serena Release Automation Client Connection Information" on page 54
- "ZMF Client Connection Information" on page 54

ALF Client Connection Information

Specify the connection to Application Lifecycle Framework (ALF), which is used by Serena Release Automation to communicate with Serena Release Manager.

> **NOTE** Dimensions CM and ChangeMan ZMF have ALF connection information stored in other files as they use different mechanisms to communicate through ALF.

To update the ALF client connection information:

1 On the **ALF** tab, make sure the variables are set to the correct connection information for your installation of the SBM event manager as follows:

Parameter	Value
`ALF_EVENTMANAGERURL`	URL to the SBM server where the ALF event manager Web services are installed, in the form of: `http://<hostname>:<port>/eventmanager/services/ALFEventManagerDocLit`
`AE_USERID`	User ID with access to the appropriate SBM projects and tables controlled by Serena Release Manager. This user ID must exist in SBM.
`AE_PASSWORD`	Password for the SBM user ID.

Example

`alf-client-connection.properties`

```
ALF_EVENTMANAGERURL = http://sbmhost:8085/eventmanager/services/ALFEventManagerDocLit
AE_USERID = rlmadmin

AE_PASSWORD = rlmadmin_test
```

SBM Application Connection Information

Specify the instance of SBM if you are using SBM as a provider for Development Change Requests (DCRs). These may reside in an SBM system other than the SBM system where Serena Release Manager resides.

To update the SBM connection information:

1 On the **SBM** tab, make sure the variables are set to the correct connection information for your installation of SBM as shown in the following example.

Example

Set the SBM provider connection for DCRs. This example gives Serena Release Manager the URL to connect to the SBM application server on the local host at port 80.

`sbm-client-connection.properties`

```
SBM_APPWS_URL = http://localhost:80/gsoap/gsoap_ssl.dll?sbmappservices72
```

BCR Provider Connection Information

Specify the instance of SBM if you are using SBM as a provider for Business Change Requests (BCRs). These may reside in an SBM system other than the SBM system where Serena Release Manager resides.

To update the BCR provider connection information:

1 On the **BCR** tab, make sure the variables are set to the correct connection information for your installation of SBM that provides BCRs as shown in the following example.

Example

Set the provider connection for BCRs. This example sets the values needed to retrieve BCRs from the SBM application services running at on port 80 of host `svr-dev-sbmv01`.

`bcr-connection.properties`

```
bcr.ws.app.url=http://svr-dev-sbmv01:80/gsoap/gsoap_ssl.dll?sbmappservices72
bcr.user=rlmadmin
bcr.password=rlmadmin_test
```

RFC Provider Connection Information

Specify the instance of SBM if you are using SSM as a provider for Requests for Change (RFCs). These may reside in an SBM system other than the SBM system where Serena Release Manager resides. Other systems may also be used as the RFC provider depending on your implementation details.

NOTE The default implementation of SSM expects Serena Release Manager to be installed in the same instance of SBM as SSM.

To update the RFC provider connection information:

1 On the **RFC** tab, make sure the variables are set to the correct connection information for your installation of SBM that provides RFCs as shown in the following example.

Example

Set the SSM provider details for RFCs. This example sets the values needed to retrieve RFCs from the SSM, with the SBM application services running at on port 80 of host `svr-dev-ssmv01`.

`itsm-connection.properties`

```
itsm.ws.app.url=http://svr-dev-ssmv01:80/gsoap/gsoap_ssl.dll?sbmappservices72
itsm.user=rlmadmin
itsm.password=rlmadmin_test
```

Dimensions CM Client Connection Information

Specify the connection to Dimensions CM if it is used as a provider for your Development Change Requests (DCRs) or Deployment Units (DUs).

To update the Dimensions CM client connection information:

1 On the **Dimensions** tab, make sure the variables are set to the correct connection information for your installation of Dimensions CM as follows:

Parameter	Value
DIM_WS_URL	URL to the server where the Dimensions CM Web services are installed, in the form of: `http://<hostname>:<port>/dmwebservices2/services/dmwebservices/`
DIM_DBNAME	The Dimensions CM base database name to which you want to connect.

Parameter	Value
DIM_DBCONN	The Dimensions CM network instance to which you want to connect.
DIM_SERVER	Hostname where the Dimensions CM server is running.

Example

dm-client-connection.properties

```
DIM_WS_URL = http://dimcm_host:8080/dmwebservices2/services/dmwebservices/
DIM_DBNAME = cm_typical
DIM_DBCONN = Dim12
DIM_SERVER = dimcm_host
```

Serena Release Automation Client Connection Information

Specify the connection to Serena Release Automation if it is used to install and configure your deployed files.

To update the Serena Release Automation (Nolio) client connection information:

1 On the **Release Automation** tab, make sure the variables are set to the correct connection information for your installation of Serena Release Automation as follows:

Parameter	Value
nolio.ws.openapi.uri	The URI that points to Web services on the Serena Release Automation execution server, in the form of: http://<srvrhostname>:<srvrport>/datamanagement/ws/OpenAPIService?wsdl
nolio.username	The Serena Release Automation execution server user name.
nolio.password	The Serena Release Automation execution server password.

Example

nolio-client-connection.properties

```
nolio.ws.openapi.uri=http://stl-dev-varmd5:8080/datamanagement/ws/OpenAPIService
nolio.username=superuser
nolio.password=suser
```

NOTE The nolio-client-connection.properties file contains additional system settings that are not set in Release Manager Configurator. These should be changed only with advice from Serena experts.

ZMF Client Connection Information

Specify the connection to ChangeMan ZMF if it is used as a provider for your Deployment Units (DUs).

To update the ChangeMan ZMF connection information:

1 On the **ZMF** tab, make sure the variables are set to the correct connection information for your installation of ZMF as follows:

Parameter	Value
ZMF_WS_URL	URL to the server, or LPAR, where the ALMZMF Web services are installed, in the form of: `http://<hostname>:<port>/almzmf/services/ZMFPackageServices/`
ZMF_SERVER_HOSTADDRESS	Hostname or IP address for the z/OS mainframe server where the ZMF started task, or ZMF application, is running.
ZMF_SERVER_HOSTPORT	Port number for the z/OS mainframe server.
ZMF_SERVER_PROXYID	Mandatory proxy user ID used to log into the z/OS mainframe server on behalf of a SBM user to ZMF. See "Configuring a Proxy User ID" on page 62.
ZMF_SERVER_PROXY_PASSWORD	Password for the proxy user ID.
SERNET_HTTPSERVER	URL to the z/OS mainframe SERNET HTTP server instance, in the form of: `http://<hostname>:<port>` This is required to populate the ZMF UI widgets in Release Manager. See "Configuring the SERNET HTTP Server" on page 62.

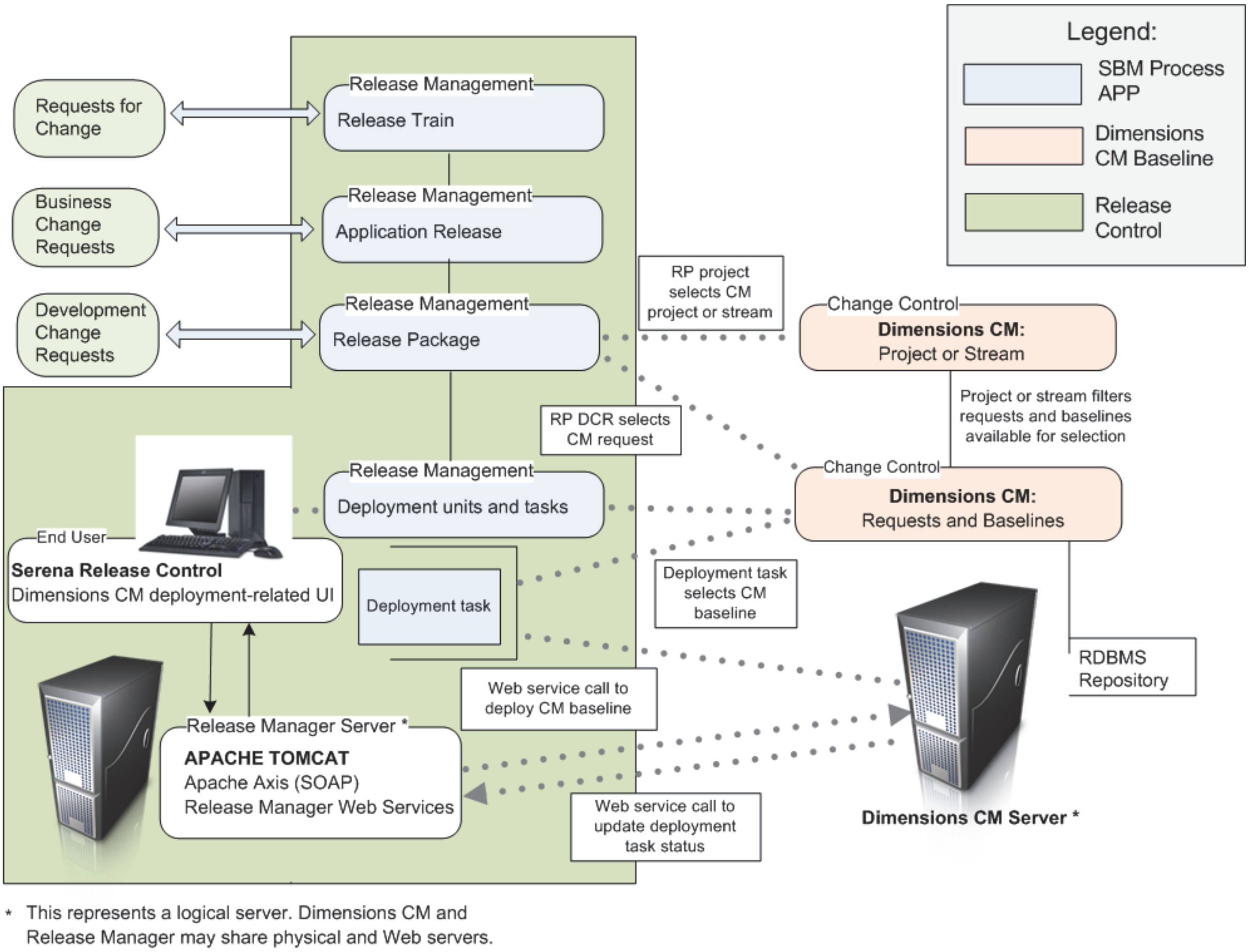
Example

`zmf-client-connection.properties`

```
ZMF_WS_URL = http://localhost:9095/almzmf/services/ZMFPackageServices/
ZMF_SERVER_HOSTADDRESS    = zmf_host
ZMF_SERVER_HOSTPORT       = 5035
ZMF_SERVER_PROXYID        = RLMMAN
ZMF_SERVER_PROXY_PASSWORD = RLMPWD1
SERNET_HTTPSERVER = http://zmf_host:5083
```

Dimensions CM Communication Configuration Overview

You must configure Dimensions CM communication on the Dimensions CM server and on the Serena Release Manager server to activate the integration. The architecture that supports the Dimensions CM integration is shown in the following figure:


Legend:
SBM Process APP
Dimensions CM Baseline
Release Control
Requests for Change
Business Change Requests
Development Change Requests
Release Management
Release Train
Release Management
Application Release
Release Management
Release Package
Release Management
Deployment units and tasks
RP project selects CM project or stream
Change Control
Dimensions CM: Project or Stream
Project or stream filters requests and baselines available for selection
RP DCR selects CM request
Change Control
Dimensions CM: Requests and Baselines
End User
Serena Release Control
Dimensions CM deployment-related UI
Deployment task
Deployment task selects CM baseline
RDBMS Repository
Web service call to deploy CM baseline
Release Manager Server *
APACHE TOMCAT
Apache Axis (SOAP)
Release Manager Web Services
Web service call to update deployment task status
Dimensions CM Server *

* This represents a logical server. Dimensions CM and Release Manager may share physical and Web servers.

Related Topics

- Configuring Communication on the Dimensions CM Server
- "Configuring Dimensions CM Communication in Release Manager" on page 58

Configuring Communication on the Dimensions CM Server

You must configure Dimensions CM communication on the Dimensions CM server, the Windows/UNIX systems release vault, so that Serena Release Control can access the correct server and instance of Dimensions CM and sign on to Dimensions CM. You must do the following:

- Specify ALF event configuration information in the `dm.cfg` file.
- Specify selection criteria for the Dimensions CM events and objects by updating the ALF event configuration file, `ALF_EVENTS_CONFIG.XML`. Specify your Dimensions CM database name, project name, baseline type, and deploy event.

For details, see the following:

- "Specifying Dimensions CM ALF Event Configuration Information" on page 57
- "Specifying Selection Criteria for Dimensions CM Events and Objects" on page 58

Specifying Dimensions CM ALF Event Configuration Information

The connection of SBM with Dimensions CM is implemented using ALF events. You must update the `dm.cfg` file on the Dimensions CM server with the ALF event configuration information, which includes the endpoints and sign-on credentials SBM uses for the connection, the location of the ALF XML file that tells ALF which Dimensions CM information to look for, and which product instance to use.

To update the ALF events configuration for Dimensions CM:

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://<sbmserver>:<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 SBM server and port number and the ALF Event Manager for that SBM 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://sbmhost:8085/eventmanager/services/ALFEventManager
DM_ALF_USER rlmadmin
DM_ALF_PASSWORD rlmadmin_pswd
DM_ALF_EVENT_CONFIG %DM_DFS%alf_events_config.xml
DM_ALF_PRODUCT_INSTANCE DimensionsUnderReleaseManagementControl
```

Specifying Selection Criteria for Dimensions CM Events and Objects

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 filter the information to be sent to Serena Release Manager.

If you are using Dimension CM as your DCR or DU provider, you must also configure the Dimensions development change request and deployment unit provider information as documented in Chapter 6, "Provider Configuration" on page 93.

To specify selection criteria for the Dimensions CM events and objects:

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 Dimensions CM Communication in Release Manager

You must configure Dimensions CM properties on the Serena Release Manager server so that Serena Release Manager can connect to and communicate with Dimensions CM. To do this, you must update the Dimensions CM Web client and connection properties files with the correct connection information as follows:

- Specify the connection information. You should have already done this through the Serena Release Manager Configurator Dimensions CM page.
- Specify Dimensions CM client-specific information in the common Tomcat Web server `webapps\rlm\WEB-INF\classes` folder `dm-client.properties` file.

To update the Dimensions CM client-specific 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 `dm-client.properties` file.
3. Make sure the variables are set to the correct client-specific information for your installation of Dimensions CM as follows:

Parameter	Value
`JOB_STATE_SUCCESS`	State that indicates success.
`JOB_STATE_FAILURE`	State that indicates failure.

4. Restart the Serena common Tomcat service.

Example

`dm-client.properties`

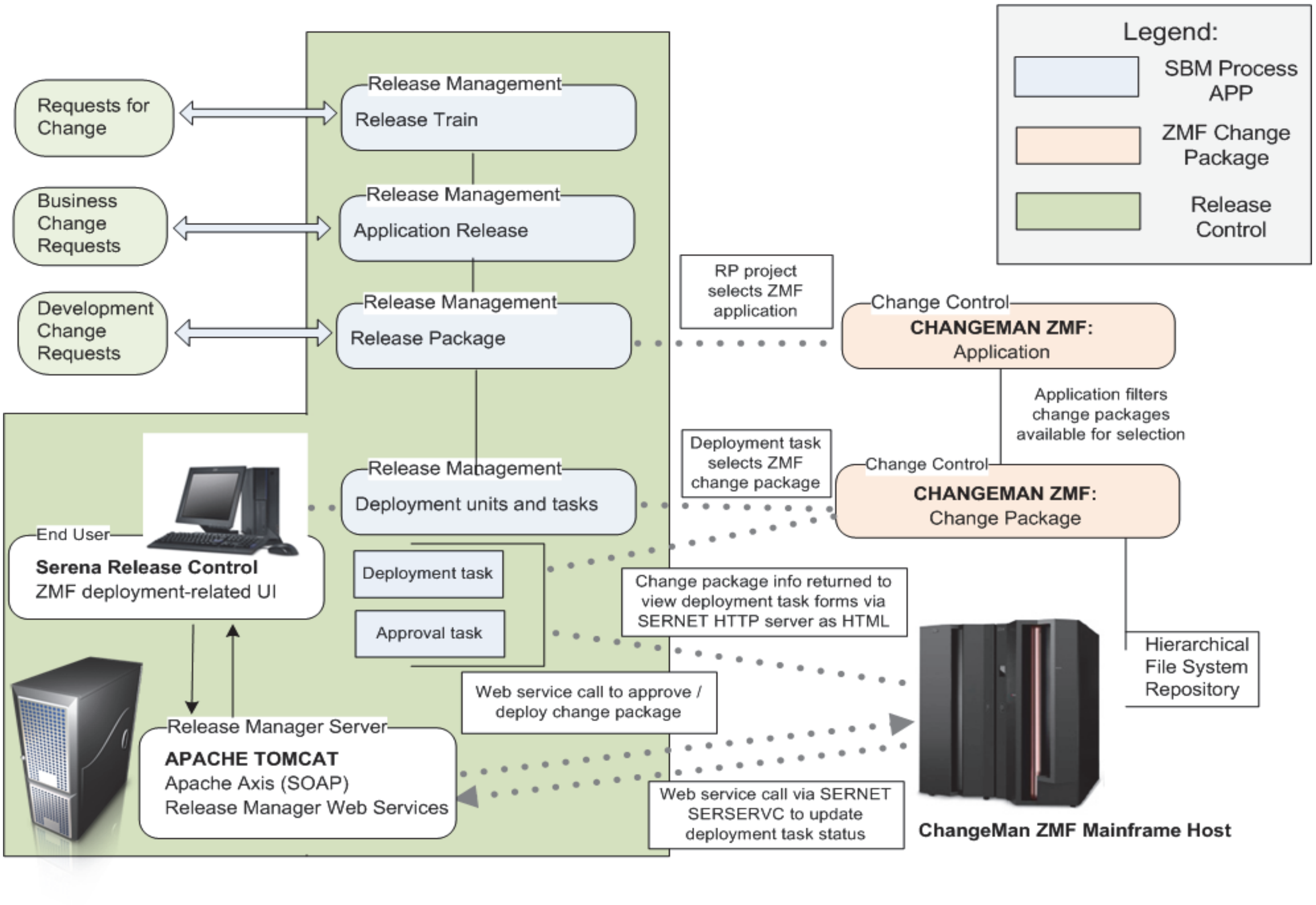
```
JOB_STATE_SUCCESS = Succeeded
JOB_STATE_FAILED = Failed
```

Related Topics

- "Configuring Connections using the Release Manager Configurator" on page 49

ZMF Communication Configuration Overview

You must configure ChangeMan ZMF communication on the z/OS mainframe and on the Serena Release Manager server to activate the integration. The architecture that supports the ZMF integration is shown in the following figure:


Legend:
SBM Process APP
ZMF Change Package
Release Control
Requests for Change
Business Change Requests
Development Change Requests
Release Management
Release Train
Release Management
Application Release
Release Management
Release Package
Release Management
Deployment units and tasks
Deployment task
Approval task
RP project selects ZMF application
Change Control
CHANGEMAN ZMF:
Application
Application filters change packages available for selection
Deployment task selects ZMF change package
Change Control
CHANGEMAN ZMF:
Change Package
End User
Serena Release Control
ZMF deployment-related UI
Change package info returned to view deployment task forms via SERNET HTTP server as HTML
Hierarchical File System Repository
Web service call to approve / deploy change package
Release Manager Server
APACHE TOMCAT
Apache Axis (SOAP)
Release Manager Web Services
Web service call via SERNET SERSERVC to update deployment task status
ChangeMan ZMF Mainframe Host

Related Topics

- Configuring the ChangeMan ZMF Web Page Widgets
- Configuring ZMF Communication on the Mainframe
- "Configuring ZMF Communication in Release Manager" on page 63

Configuring ZMF Communication on the Mainframe

You must configure ChangeMan ZMF communication on the z/OS mainframe as follows:

- Configure the NTFYURL; this is the URL Serena Release Manager uses to send information to SERNET through the server.
- Configure the SERNET HTTP server; this is the server Serena Release Manager uses to populate the UI widgets with ZMF information.

- Configure a proxy user ID for each mainframe host, or LPAR, that Serena Release Manager uses to log in to ChangeMan ZMF.
- Configure TSO user IDs that match the SBM user IDs.
- Configure approvers for each promotion level to be used with Serena Release Manager.

For details, see the following:

- "Configuring the Notification URL" on page 61
- "Configuring the SERNET HTTP Server" on page 62
- "Configuring a Proxy User ID" on page 62
- "Configuring TSO User IDs and Permissions" on page 63
- "Configuring Approvers" on page 63
- "ZMF Communication Configuration Overview" on page 60

Configuring the Notification URL

You must give ChangeMan ZMF a way to tell Serena Release Manager when ChangeMan ZMF has completed a requested task. Since information from ZMF is sent through ALF and SERNET, you must configure the notification URL parameter for the SERNET started task on z/OS.

The NTFYURL parameter is a keyword option used with the SERNET started task. This parameter is required for notifying Serena Release Manager when an ALF event is emitted from ChangeMan ZMF.

There are three different ways of passing the NTFYURL parameter to SERNET.

Whichever method you choose to pass this parameter, it must be specified as follows:

```
NTFYURL=’hostname:port/almzmfalf/services/ZMFALFEventRouter’
```

where `hostname` is the server name where Serena Release Manager Web Services are installed and `port` is the port number for that server.

This parameter is case-sensitive; the non-variable text must be entered exactly as shown. Be sure to include the quotes around the variable string.

Example

```
NTFYURL='rlm_host:9095/almzmfalf/services/ZMFALFEventRouter'
```

CAUTION! If your site is a DP site, you must specify the same hostname and port for the DP site and the P site. If not, the P site will continue to wake up looking for work and will fill up the JESMSGLG (JES message log).

Documentation References

- Complete documentation on passing parameters to SERNET is in the *Passing Parameters to SERNET* in the *Serena ChangeMan ZMF Installation Guide*.

Configuring the SERNET HTTP Server

The ChangeMan ZMF UI widget for Serena Release Manager is populated using a native mainframe HTTP server, `SERSERVC`, that runs under SERNET. See Appendix B, "ZMF: SERNET HTTP Server Setup" on page 189 for installation, operation, and runtime considerations for `SERSERVC`.

IMPORTANT!

- For the UI widgets to populate the forms with information from ZMF, the ZMF forms must be updated so that the widgets point to the correct hostname. See "Configuring the ChangeMan ZMF Web Page Widgets" on page 42.
- The URL to this server is specified in the `zmf-client-connection.properties` file, which can be updated using the Release Manager Configurator. See "Configuring Connections using the Release Manager Configurator" on page 49.

Documentation References

- Documentation on passing parameters to SERNET is in the *Passing Parameters to SERNET* in the *Serena ChangeMan ZMF Installation Guide*.

Configuring a Proxy User ID

A proxy user ID, or trusted user ID, is required for each ChangeMan ZMF host server, or LPAR. You specify these in the `zmf.properties` configuration file when you configure ZMF communication on the Serena Release Manager server.

The purpose of the trusted user ID is to allow users to automatically access ChangeMan ZMF through Serena Release Manager without logging on. The trusted ChangeMan ZMF user ID connects to the host server on behalf of the user.

Consider an example where a user wants to freeze a release unit. The orchestration invoked for the Freeze function requires access to the ChangeMan ZMF host server. The user's TSO user ID is on his SBM contact record and is associated with the trusted user ID; however, there is no password stored in the user's contact record. The trusted user ID (which does have a password) logs on to the ChangeMan ZMF host server on behalf of the user. The trusted user ID impersonates the user, but does not have access to other resources (such as performing ChangeMan ZMF functions). The authority levels of the user are in effect for the transaction.

The trusted user ID can be any SAF-defined user ID. No specific attributes are required. It is not necessary that this user ID be allowed to access TSO. This user ID must be given READ (or higher) access to the "trusted resource". The trusted resource is a SAF resource, by default SERENA.SERNET.AUTHUSR in the FACILITY class. The resource and class are user-modifiable by changing the names in the SERLCSEC CSECT, which is delivered as source code with ChangeMan ZMF. This CSECT is used for customizing a variety of security-related functions.

NOTE It is not necessary to alter SERLCSEC to support Serena Release Manager in the default fashion, as the latest version is already coded for the above resource name and class. Be sure to use the latest version of this CSECT. If you have previously modified it, you will need to re-apply your customizations.

IMPORTANT! The Serena Release Manager *trusted resource* is not related to the RACF user ID TRUSTED attribute.

Configuring TSO User IDs and Permissions

All Serena Release Manager users should have a TSO user ID with the appropriate permissions. A user may not have needed a TSO user ID prior to using Serena Release Manager; however, when using Serena Release Manager, the user may initiate a transition in a workflow that orchestrates an action in ChangeMan ZMF (such as a promotion of a change package) which requires that his user ID be passed to ChangeMan ZMF.

These user IDs must have access to every resource required by ChangeMan ZMF functions that Serena Release Manager uses.

NOTE Serena Release Manager extracts the SBM user ID from the SSO certificate and uses it when invoking the ZMF Web services. The `loginAsUserID` sent to ZMF through the ZMF XML services is the SBM user ID.

Configuring Approvers

You must configure approvers in ZMF as needed to initiate the installation and baselining of change packages.

For example, if you configure just one approver, once the approval is given by the approver, the change package will go into ‘APR” status and the installation will be initiated. The Serena Release Manager ZMF approval deployment task will go into “In Progress” state and will wait for the ZMF package to be installed and baselined. After the ZMF package goes into “BAS” status, the Serena Release Manager ZMF approval deployment task will go into “Complete” state.

An example of how a release engineer might specify approval deployment tasks for the default release stages of Serena Release Manager is given in the *Serena Release Manager Getting Started Guide* in "Creating Approval Deployment Tasks".

NOTE In ChangeMan ZMF, specify approvers on the Planned Approvals panel in option A.A.5. Approvers must have approval authority.

Configuring ZMF Communication in Release Manager

You must configure ChangeMan ZMF properties on the Serena Release Manager server so that Serena Release Manager can connect to and communicate with ChangeMan ZMF. To do this, you must do the following:

- Specify the connection information. You should have already done this through the Serena Release Manager Configurator ZMF page.

- Specify ALF event manager information for ChangeMan ZMF in the Serena Release Manager common Tomcat Web server `webapps\almzmfalf\WEB-INF\conf` folder `alfzmf_resource.properties` file.
- Specify other ZMF client-specific information in the Serena Release Manager common Tomcat Web server `classes` folder `zmf-client.properties` file.

For details, see the following:

- "Configuring Connections using the Release Manager Configurator" on page 49
- "Specifying ALF Event Manager Connection Information for ZMF" on page 64
- "Specifying Client-Specific Information for ChangeMan ZMF" on page 65
- "ZMF Communication Configuration Overview" on page 60

Specifying ALF Event Manager Connection Information for ZMF

So that Serena Release Manager can receive information from ChangeMan ZMF through the ALF event manager, you must update the ALF event properties file used by the ZMF integration with sign-on credentials SBM uses for the connection.

NTFYURL sends information to the ALF event service URL, which then sends the events to the ALF event manager. The user ID and password of the SBM user that initiated the event must have access to all the SBM Serena Release Manager objects and must also be a valid user ID in ChangeMan ZMF.

To update the ZMF ALF connection information:

1. Navigate to the Serena Release Manager common Tomcat Web server `webapps\almzmfalf\WEB-INF\conf` folder. For example:

   ```
   C:\Program Files\Serena\common\tomcat\6.0\webapps\almzmfalf\WEB-INF\conf
   ```

2. Open the `zmfalf_resource.properties` file.
3. Set the `AE_USERID` and `AE_PASSWORD` variables with the user ID and password of your Serena Release Manager administrative user.
4. Set the properties for the connection to the ALF event manager as follows:

Parameter	Value
`ALF_EVENTMANAGERURL`	URL to the SBM server where the ALF event manager Web services are installed, in the form of: `http://<hostname>:<port>/eventmanager/services/ALFEventManager`
`AE_USERID`	User ID with access to the appropriate SBM projects and tables and ZMF applications controlled by Serena Release Manager. This user ID must exist in both SBM and ZMF.
`AE_PASSWORD`	Password for the SBM user ID.
`AE_VERSION`	Version of SBM. This is for documentary purposes only.

5. Restart the Serena Common JBOSS and IIS Admin Service services.

Example

`zmfalf_resource.properties`

```
# Property resource bundle file for Axi2 ZMF Service
# Used to configure Axis2 ZMF Service system properties.

ALF_EVENTMANAGERURL = http://sbmhost:8085/eventmanager/services/ALFEventManager

# The AE userid must have access to the appropriate SBM projects/tables and it must also
     have access to ZMF applications controlled by RLM.
AE_USERID = rlmadmin
AE_PASSWORD = rlmadmin_test
AE_VERSION = SBM 2009 R3
```

Specifying Client-Specific Information for ChangeMan ZMF

So that Serena Release Manager knows which status to expect for successful and failed responses from ChangeMan ZMF, you must specify this information in the appropriate properties file.

To specify the ZMF response status 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 `zmf-client.properties` file.

3. Set the properties for the connection as follows:

Parameter	Value
`JOB_STATE_SUCCESS`	State that indicates success.
`JOB_STATE_FAILURE`	State that indicates failure.

4. Restart the Serena common Tomcat service.

Example

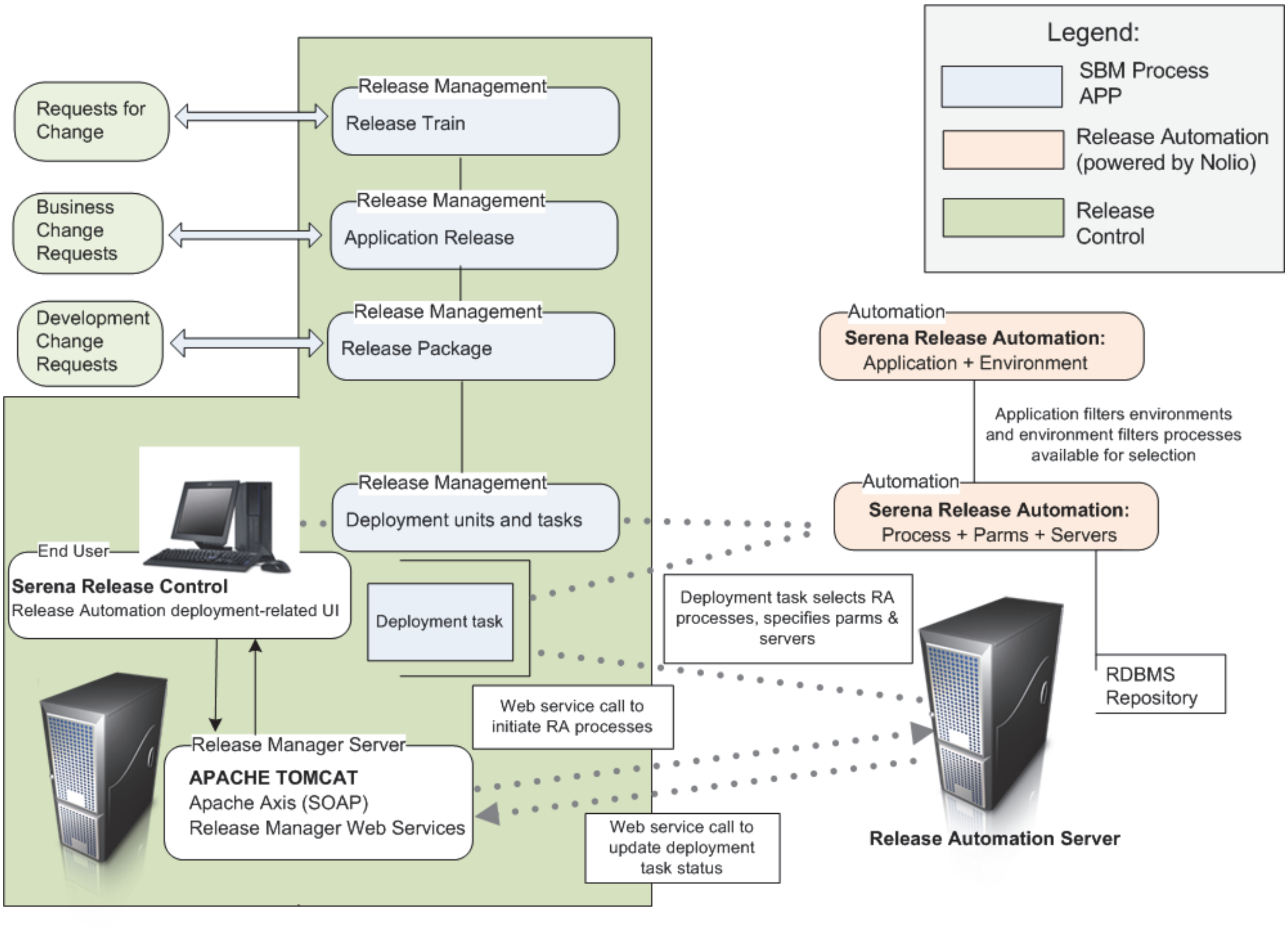
This example sets the values needed to connect to ChangeMan ZMF.

`zmf-client.properties`

```
# Specify ZMF promotion success and failed state.
JOB_STATE_SUCCESS = Completed
JOB_STATE_FAILED  = Failed
```

Serena Release Automation Communication Configuration Overview

You must configure Serena Release Automation communication on the Serena Release Automation server and on the Serena Release Manager server to activate the integration. The architecture that supports the Serena Release Automation integration is shown in the following figure:


Legend:
SBM Process APP
Release Automation (powered by Nolio)
Release Control
Requests for Change
Business Change Requests
Development Change Requests
Release Management
Release Train
Release Management
Application Release
Release Management
Release Package
Release Management
Deployment units and tasks
End User
Serena Release Control
Release Automation deployment-related UI
Deployment task
Release Manager Server
APACHE TOMCAT
Apache Axis (SOAP)
Release Manager Web Services
Web service call to initiate RA processes
Web service call to update deployment task status
Deployment task selects RA processes, specifies parms & servers
Automation
Serena Release Automation:
Application + Environment
Application filters environments and environment filters processes available for selection
Automation
Serena Release Automation:
Process + Parms + Servers
RDBMS Repository
Release Automation Server

Related Topics

- Configuring Communication on the Release Automation Server
- "Configuring Release Automation Communication in Release Manager" on page 69

Configuring Communication on the Release Automation Server

So that Serena Release Manager can get the information from Serena Release Automation about Serena Release Automation applications, processes, events, and server lists, you must configure the communication on the Serena Release Automation server as follows:

- Specify the Serena Release Automation server to notify when an event occurs in the Serena Release Automation `rest.integration.properties` file. If the file does not exist, create it.
- Update the Serena Release Automation environment notifications for each application to tell Serena Release Automation the events about which to notify Serena Release Manager.

For details, see the following:

- "Specifying the Serena Release Automation Server to Notify" on page 67
- "Telling Release Automation Which Event Notifications to Send" on page 67

Specifying the Serena Release Automation Server to Notify

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

To specify the Serena Release Automation server:

1. On the Serena Release Automation server, navigate to the Serena Release Automation installation directory. For example:

   ```
   C:\Program Files\Serena\Serena Release Automation\conf
   ```

2. Open the Serena Release Automation `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:

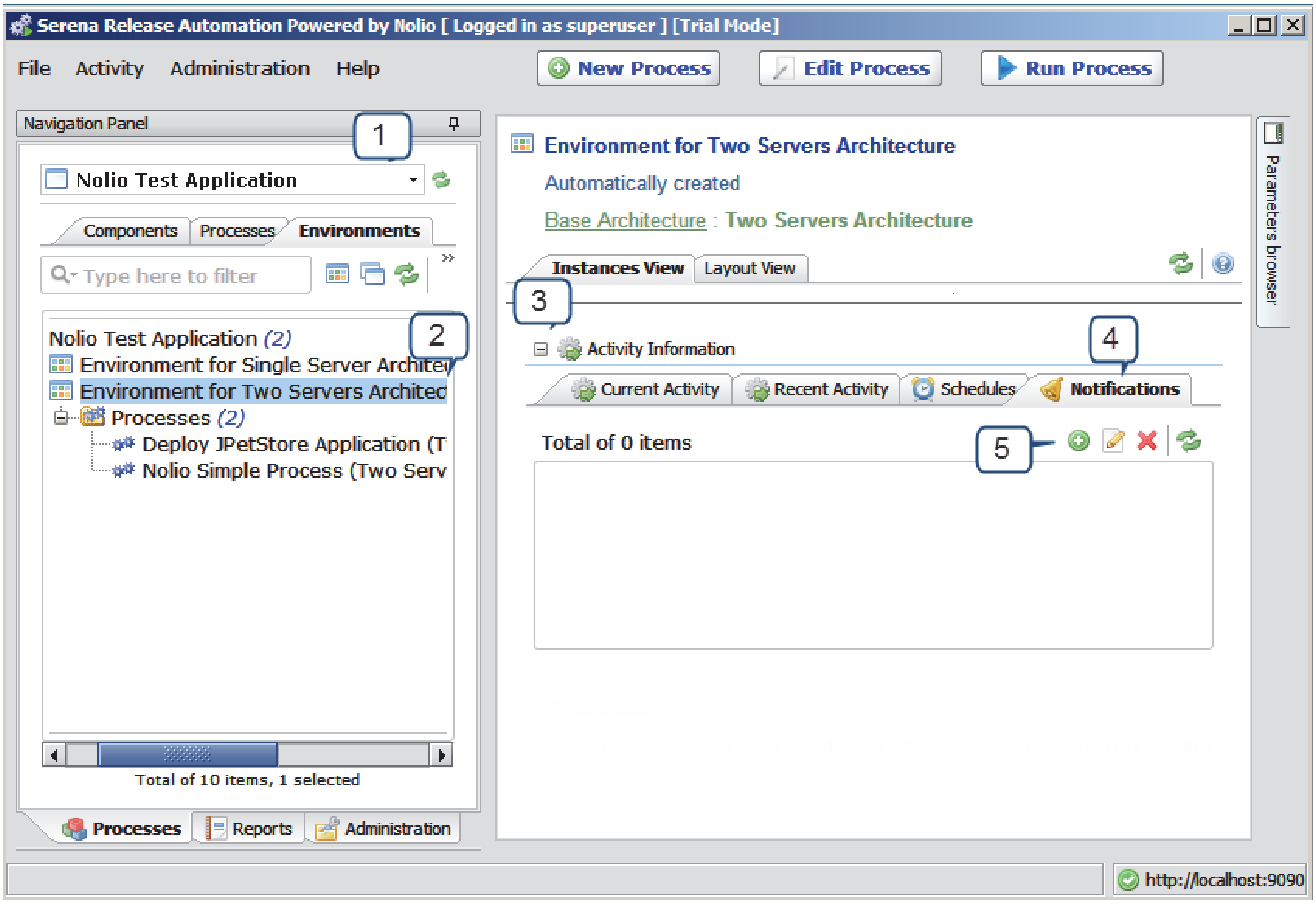
   ```
   target.url=http://rlmhost:9095/rlm/servlet/nolionotification
   ```

4. Restart the Nolio Server and Nolio Agent services.

Telling Release Automation Which Event Notifications to Send

You must update the Serena Release Automation environment notifications for each application to tell Serena Release Automation the events about which to notify Serena Release Manager.

Serena Release Automation is shown in the following figure.



To configure Serena Release Automation environment notification:

1 From Serena Release Automation, for each application, select the application.

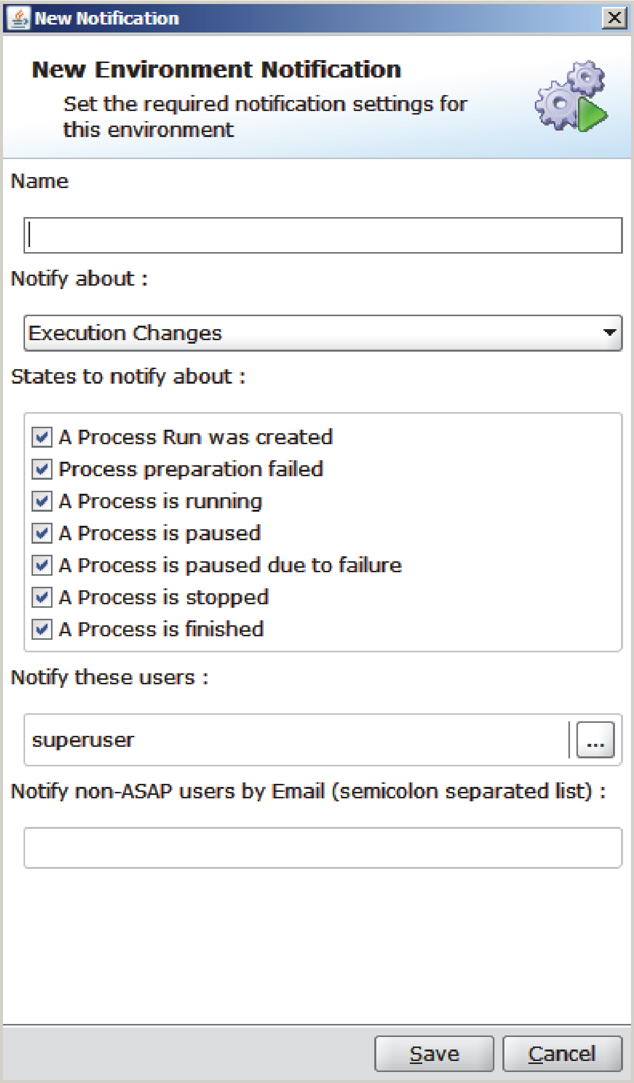
2 For each environment, select the environment.

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.



New Notification

New Environment Notification

Set the required notification settings for this environment

Name

Notify about : Execution Changes

States to notify about :

- ☑ A Process Run was created
- ☑ Process preparation failed
- ☑ A Process is running
- ☑ A Process is paused
- ☑ A Process is paused due to failure
- ☑ A Process is stopped
- ☑ A Process is finished

Notify these users : superuser

Notify non-ASAP users by Email (semicolon separated list) :

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

7 Click **Save**.

Configuring Release Automation Communication in Release Manager

You must configure Serena Release Automation properties on the Serena Release Manager server so that Serena Release Manager can connect to and communicate with Serena Release Automation. To do this, you must do the following:

- Update the Serena Release Automation ALF sign-on credentials. You should have already done this through the Serena Release Manager Configurator ALF page.
- Specify the Serena Release Automation client-specific information in the Serena Release Manager common Tomcat Web server `classes` folder `nolio-client.properties` file.

- Specify the Serena Release Automation client query information in the Serena Release Manager common Tomcat Web server `classes` folder `nolio-client-queries.properties.properties` file.

Related Topics

- "Configuring Connections using the Release Manager Configurator" on page 49
- Specifying Client-Specific Information for Release Automation
- Specifying Serena Release Automation Queries

Specifying Client-Specific Information for Release Automation

You must specify the client-specific information for Serena Release Automation to set time out and wait values and states that indicate success or failure.

To specify the Serena Release Automation client-specific 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-client.properties` file.

3 Set the properties for the client as follows:

Parameter	Value
`nolio.job.runprocess.timeout`	Indicates how long to wait until a timeout message is received.
`nolio.job.runprocess.wait`	Indicates whether to run the Release Automation process in wait mode. Values are true and false.
`nolio.job.state.success`	List of Serena Release Automation job states that indicate success, delimited by commas.
`nolio.job.states.failure`	List of Serena Release Automation job states that indicate failure, delimited by commas.

Example

This example sets the values needed to retrieve successful and failed job notifications from Serena Release Automation.

`nolio-client.properties`

```
nolio.job.runprocess.timeout=0
nolio.job.runprocess.wait=false

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,FLOW_STOPPED
```

Specifying Serena Release Automation Queries

You may specify SQL queries for the information you want returned from Serena Release Automation.

To specify the Serena Release Automation 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-client-queries.properties` file.

3. Set the properties for the queries as follows:

Parameter	Value
`query.get.applications`	SQL queries to return the set of applications, environments, processes, and servers you want for the automation deployment tasks. For the default queries, see the example.
`query.get.environments`	
`query.get.processes`	
`query.get.servers`	
`query.find.application`	
`query.get.job.status`	
`message.application.not.found`	Message to send if applications aren’t found.
`message.environment.not.found`	Message to send if environments aren’t found.

Example

This example shows the default SQL queries for Serena Release Automation.

`nolio-client-queries.properties`

```
query.get.applications = select app_name  from applications where id!=1
query.get.environments = select name  from environments where id!=1 and applicationId =
     ?
query.get.processes    = select process_name from process_container pc, process_in_env
     pe where pc.id = pe.process_id and pe.environment_id = ? order by process_name
query.get.servers      = select s.server_name, st.server_type_name from servers s,
     server_types st, server_type_instance sti where s.id = sti.mapped_server and st.id
     = sti.server_type and sti.environment_id = ?

query.find.application = select id from applications where app_name = ?
query.find.environment = select id from environments where name = ? and applicationId =
     ?

query.get.job.status = select step_title,step_state from step_events where id in (select
     MAX(id) from step_events where job_id=?)

message.application.not.found = Could not find specified application: {0}
message.environment.not.found = Could not find specified environment: {0}
```

Chapter 5

Configuration and Administration of the Integrating Objects

This section tells you how to configure and administer objects 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 must complete the following configuration before the people who participate in release management in your organization begin using Serena Release Manager.

Configuring Objects in Serena Release Control

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

Related Topics

- "Accessing the Standard SBM User Interface" on page 74
- "Adding Your Application Names in Serena Release Control" on page 74
- "Adding Your Server Names in Serena Release Control" on page 75
- "Managing Release Control Users" on page 76
- "Managing Release Control Reports" on page 78
- "Managing Release Control Notifications" on page 81
- "Configuring Objects in Serena Business Manager" on page 90
- "Configuring Objects in Serena Service Manager" on page 91

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 in Serena Release Control?

You can change the following application information:

- Add application name and description.
- Update 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. In SBM Application Administrator, click **Auxiliary Data**.
2. In the **Table** field, select **Application**.
3. Click **New** and enter the application name and description.
4. You may select from existing deployment process templates here or add them from the user interface.
5. Click **OK** to save.

Documentation References

Complete documentation on adding data to SBM auxiliary tables is in the *Serena Business Manager Application Administrator Guide* in "About Auxiliary Data".

Adding Your Server Names in Serena Release Control

Before you start using environments in Serena Release Control, you must pre-populate Serena Release Control with the names of your servers that you plan to manage over time using Serena Release Manager. This enables you to associate those servers with environments that you manage in Serena Release Control.

An environment represents one or more servers associated to a specific stage a release train or release package is going through. For example, you may have one or more UAT environments available that are used in parallel for different purposes, where different applications are installed on each for different testing scenarios. You may also have more than one INT environment available, more than one pre-PROD environment available, and so forth.

You can create environments in Serena Release Control and manage their availability through the associated workflow actions. You can report on the availability and schedule release trains and release packages accordingly.

In the default implementation of Serena Release Manager, there is not yet an active connection to the Serena Release Control objects, but you may choose to extend and customize this functionality. For example, you could customize the system to use the environment server configuration to store parameters in specific field values, which could then be passed to Serena Release Automation.

What Can You Change in Serena Release Control?

You can change the following server information:

- Add server name and description.

What is the Impact?

The server names appear in the **Create Environment** dialog box in the **Servers** selection field. If you change the names or add names, this impacts the list of names the users see when they select servers for environments.

How Do You Change It?

You add and change server information in the **Server** auxiliary table.

To change the Server table entries:

1. In SBM Application Administrator, click **Auxiliary Data**.
2. In the **Table** field, select **Server**.
3. Click **New** and enter the values into the following fields:
   - Name
   - Description
   - IP Address
   - Hostname
4. Click **OK** to save.

Documentation References

Complete documentation on adding data to SBM auxiliary tables is in the *Serena Business Manager Application Administrator Guide* in "About Auxiliary Data".

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 in Release Manager?

User and group changes are done in native SBM.

What Can You Change in SBM?

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

If you add a role in SBM, If you change or add roles, you must also:

- Enable roles in projects for any workflows in which this role may be assigned ownership.
- Change privileges of roles that are assigned to states.
- You may need to change or add related 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 "Users and Roles" in the *Dimensions CM Process Modeling User's Guide*.

How Do You Change It?

You should 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 Application Administrator.
- You can grant privileges not related to roles, such as administrative privileges, in SBM Application Administrator.

Documentation References

- Complete documentation on managing roles in SBM is in the *Serena Business Manager SBM Composer Guide* in "Creating Roles".
- Complete documentation on managing users in SBM is in the *Serena Business Manager Application Administrator Guide* in "Managing Users".

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.

Role Name	Description
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.
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 Serena Release Automation. 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

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

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 8, "Configuring the User Interface" on page 116.

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.

Objects	Reports	Description
Release Trains	Dashboard	A multi-view report for the **Dashboard** page. See Chapter 8, "Configuring the Serena Release Control Dashboard" on page 116.
	Inactive RTrains	All inactive release trains.
	Inactive Release Trains	All release trains that are in inactive status.
	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 8, "Configuring the Calendars" on page 118.
	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 8, "Configuring the Inbox" on page 119.
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 8, "Configuring the Inbox" on page 119.
	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.
	Inactive Application Releases	Application releases that are in inactive status.
Release Packages	Active Release Packages (JSON)	All release packages that are in active status used for the **Activity** page. See Chapter 8, "Configuring the Activity Page" on page 121.
	Inactive Release Packages	All release packages that are in inactive status.
	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 8, "Configuring the Inbox" on page 119.
	Release Packages in Application Release	Release packages in the selected application release.
	Release packages list by owner	Release packages list by owner.
	Unassigned Release Packages	Release packages that are not yet assigned to application releases.

Objects	Reports	Description
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 8, "Configuring the Inbox" on page 119.
	Template Tasks	Tasks in the selected deployment process template.
Environment	Environment in Release Packages	All environments in the selected release package.
	Environments All	All environments.
	Environments owned by current user	Environments owned by the current user.
	Environments Commissioned	All environments in the commissioned state.
	Inactive Environments	All inactive environments.
Others	All Release Type Stages	All stages sorted by release type and sequence.
	All Stages	All stages defined in Serena Release Control.
	Assigned Business Change Requests	Business change requests associated with the selected application release.
	Assigned Deployment Units	Deployment units associated with the selected release package.
	Assigned Development Change Requests	Development change requests associated with the selected release package.
	Assigned Requests for Change	Requests for change associated with the selected release train.
	DCR Projects Assigned to a Release Package	Development change requests associated with the selected release package.
	Dimensions CM Projects Assigned to a Release Package	Dimensions CM projects associated with the selected release package.
	Release Train Scope Change	All RFCs added to or removed from the selected release train since the release train was approved.
	Stages in Release Type	All stages for the selected release type.
	ZMF Projects Assigned to a Release Package	ZMF projects, or applications, associated with the selected release package.

What Can You Change in Release Manager?

You can manage reports in the **Reports** tab of Serena Release Manager, but for full feature access, it is recommended that you make configuration changes using native SBM.

What Can You Change in SBM?

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.
- If you rename a report that is used for the UI, the UI element affected may no longer work. You must save the report in place using the same report reference name to prevent this.

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 Manager 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".

Managing Release Control Notifications

Notifications help keep release management stakeholders informed of release status information. If subscribed, e-mail notifications are sent to Serena Release Control users to alert them of actions requiring their attention and to provide important release status information.

Serena Release Control provides a default set of notifications, and you can configure these as needed to support your organization. The 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
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 Automation Task fails for Release Engineer or Manager
	D - Any Vault Task fails for Release Engineer or Manager
	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 an In Progress Automation Task
	D - I become the owner of an In Progress Manual Task
	D - I become the owner of an In Progress Vault Task
	D - I become the owner of any Deployment

Entity	Notifications
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
Development Change Request (DCR): SBM Incident	I - Any Incident changes owner
	I - Any Incident changes state
	I - Any Incident changes to inactive
	I- Any Incident I submitted changed state
	I - Any Incident I submitted changed to inactive
	I- Any Incident is submitted
	I - I become the owner of any Incident
Business Change Request (BCR): SBM Issue	I - Any Issue changes owner
	I - Any Issue changes state
	I - Any Issue changes to inactive
	I- Any Issue I submitted changed state
	I - Any Issue I submitted changed to inactive
	I- Any Issue is submitted
	I - I become the owner of any Issue
Development Change Request (DCR): Dimensions CM Request	RV - Any Change Request changes owner
	RV - Any Change Request changes state
	RV - Any Change Request changes to inactive
	RV - Any Change Request I submitted changed state
	RV - Any Change Request I submitted changed to inactive
	RV - Any Change Request is submitted
	RV - I become the owner of any Change Request

Entity	Notifications
Request for Change (RFC): SSM CAR	CAR - Any Change Request changes owner
	CAR - Any Change Request changes state
	CAR - Any Change Request changes to inactive
	CAR - Any Change Request I submitted changed state
	CAR - Any Change Request I submitted changed to inactive
	CAR - Any Change Request is submitted
	CAR - I become the owner of any Change Request
Environment	E - Any Environment changes owner
	E - Any Environment changes state
	E - Any Environment changes to inactive
	E - Any Environment I submitted changed state
	E - Any Environment I submitted changed to inactive
	E - Any Environment is submitted
	E - I become the owner of any Environment

What Can You Change in Release Manager?

Notification changes are done in native SBM.

What Can You Change in SBM?

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 Objects in Dimensions CM

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

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

- "Configuring the Dimensions CM Global Stage Lifecycle" on page 85
- "Managing Dimensions CM Users" on page 85
- "Configuring Dimensions CM Projects and Streams" on page 87
- "Available Selection of Requests and Baselines" on page 88

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 Manager 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 Manager 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 Manager:

- An administrative user that is specified in the system configuration files and that Serena Release Manager 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 Manager through Dimensions CM.
  - Access to all request information for applications that will be tracked from Serena Release Manager through Dimensions CM as the Development Change Request provider.
  - Access to all baseline information for applications that will be deployed from Serena Release Manager through Dimensions CM as the Deployment Unit provider.
  - Ability to deploy baselines for applications that will be deployed from Serena Release Manager through Dimensions CM.
- Any users that have roles in both Dimensions CM and Serena Release Manager, such as Serena Release Control power users or Serena Release Manager administrators.

What Can You Change in Release Manager?

Dimensions CM object changes are done in Dimensions CM.

What Can You Change in Dimensions CM?

- 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 Manager and Dimensions CM to work.

What is the Impact?

- When a user is signed on 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 Manager communication to Dimensions CM, the Serena Release Manager integration to Dimensions CM 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 "Users and Roles".

Configuring Dimensions CM Projects and Streams

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

Serena Release Manager 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 Manager integrates with SBM and Dimensions CM projects.

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

- 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 in Release Manager?

Dimensions CM object changes are done in Dimensions CM.

What Can You Change in Dimensions CM?

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".

Available Selection of Requests and Baselines

If Dimensions CM is used as one of your development change request providers, 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 one of your deployment unit providers, 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 6, "Provider Configuration" on page 93.

Configuring Objects in Serena Release Automation

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

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

- "Configuring Serena Release Automation Users" on page 88
- "Configuring Serena Release Automation Processes and Servers" on page 89

Configuring Serena Release Automation Users

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

There are two main types of Serena Release Automation users that interact with Serena Release Manager:

- An administrative user that is specified in the system configuration files and that Serena Release Manager uses to sign on to Serena Release Automation through Web services and remote database access. This user must have privileges for the following in Serena Release Automation:
  - 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 Serena Release Automation and Serena Release Manager, such as Serena Release Manager power users or Serena Release Manager administrators.

What Can You Change?

- You can change user and role information in Serena Release Automation as needed for the users’ roles in Serena Release Automation.

- 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.

What is the Impact?

- When you create an automation deployment task in Serena Release Control, the super user credentials specified in the configuration files are 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, the automation deployment tasks will fail.

How Do You Change It?

Serena Release Automation administrators should configure users in Serena Release Automation according to the Serena Release Automation documentation.

Documentation References

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

Configuring Serena Release Automation Processes and Servers

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. This includes:

- Applications
- Environments
- Processes
- Servers

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

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

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

What Can You Change?

- Serena Release Automation administrators can change objects in Serena Release Automation according to the Serena Release Automation documentation.
- You can pass application parameters to invoke desired operations on target servers defined in Serena Release Automation.

NOTE You cannot pass server parameters in the default implementation; you can pass only application parameters.

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?

Serena Release Automation administrators should configure application, environment, process, and server information according to the Serena Release Automation documentation.

Documentation References

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

Configuring Objects in Serena Business Manager

In addition to the objects you configure in SBM to support Serena Release Control, Serena Release Manager may rely on other SBM information.

If you use SBM process apps, such as Issue Defect Management, Incident Management, or Change Request Management, to provide requests for change (RFCs), business change requests (BCRs), or development change requests (DCRs), you must configure the required information in SBM. That information includes:

- Projects
- Items, such as issues or incidents

What Can You Change?

You can add or change the above information in the SBM process apps you are using, but not in Serena Release Manager. The only things that change in the SBM process apps from Serena Release Manager should be a result of automations built into the release train workflow in relation to associated RFCs, application releases in relation to associated BCRs, or release packages in relation to DCRs.

What is the Impact?

- SBM items, such as approved incidents, appear in the Application Release dialog box. If you change or add items, this may impact the list of items the users see when they select BCRs for an application release.
- Project names for DCRs appear in the project selection table in the Release Package dialog box. If you change the names or add names, this may impact the list of names the users see when they select projects for a release package.
- SBM items, such as approved issues, appear in the Release Package dialog box. If you change or add items, this may impact the list of items the users see when they select DCRs for a release package.

How Do You Change It?

SBM administrators should change information in SBM according to the SBM documentation.

Documentation References

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

Related Topics

- "Configuring Objects in Serena Release Control" on page 74
- "Configuring Objects in Serena Service Manager" on page 91
- Chapter 6, "Provider Configuration" on page 93

Configuring Objects in Serena Service Manager

If you use Serena Service Manager (SSM) to provide requests for change (RFCs), you must configure the required information in SSM. That information includes:

- Projects
- Items, such as change requests

What Can You Change?

You can add or change the above information in SSM, but not in Serena Release Manager. The only things that change in SSM from Serena Release Manager should be a result of automations built into the release train workflow in relation to associated RFCs.

What is the Impact?

- SSM items, such as change requests, appear in the Release Train dialog box. If you change or add items, this may impact the list of items the users see when they select RFCs for an application release.

How Do You Change It?

SSM administrators should change information in SSM according to the SSM and SBM documentation.

Documentation References

- Complete documentation on submitting and actioning SSM change requests is in the *Serena Service Manager ITIL Guide* in "Change Management".
- Complete documentation on configuring SBM projects is in the *Serena Business Manager Application Administrator Guide* in "About Projects".
- Complete documentation on submitting items into SBM projects is in the *Serena Business Manager System User's Guide* in "Working with Primary Items".

Related Topics

- "Configuring Objects in Serena Business Manager" on page 90

- "Configuring Objects in Serena Release Control" on page 74
- Chapter 6, "Provider Configuration" on page 93

Configuring Objects in ChangeMan ZMF

Serena Release Manager uses several objects that your ChangeMan ZMF administrators configure as part of the ongoing use of ChangeMan ZMF in addition to some that must be configured specifically to support Serena Release Manager.

Objects that you will ordinarily already have set up as part of your normal administration and use of ChangeMan ZMF are as follows:

- Applications (Projects)
- Sites (Environments)
- Change Packages
- Approver lists
- Promotion levels
- Audit return code rules

What Can You Change?

You can change any of the above information in ChangeMan ZMF, but not in Serena Release Manager. The only things that change in ZMF should be a result of ZMF vault and approval deployment tasks initiated by deploying a release package in Serena Release Manager.

What is the Impact?

If you update information in ZMF for the change packages for which you have initiated the installation from Serena Release Manager, you may impact the result in Serena Release Manager.

How Do You Change It?

ChangeMan ZMF administrators should change ZMF objects in ZMF according to the ZMF documentation.

Documentation References

- Complete documentation on configuring ZMF objects is in the *Serena ChangeMan ZMF Administrator Guide*.

Chapter 6

Provider Configuration

This section tells you how to configure existing provider connections.

See the following sections for details.

Provider Configuration Overview

Serena Release Manager providers are products that integrate, or interface, with Serena Release Manager to provide access to information that you want to include as part of your release management solution.

The default implementation includes connections to the following default providers:

- Request for Change (RFC) provider connection for Serena Business Manager (SBM)
- Request for Change (RFC) provider connection for Serena Service Manager (SSM)
- Business Change Request (BCR) provider connection for SBM
- Business Change Request (BCR) provider connection for SSM
- Development Change Request (DCR) provider connection for SBM
- Development Change Request (DCR) provider connection for Dimensions CM
- Deployment Unit (DU) provider connection for Dimensions CM
- Deployment Unit (DU) provider connection for ChangeMan ZMF

NOTE This section tells how to configure connections to providers for which underlying integration layers have been implemented using the Serena Release Manager provider mechanism. For information on how to implement a new provider using this mechanism, see Chapter 8, "Adding Provider Connections" on page 138.

For details on configuring access to existing providers, see the following:

- "Configuring Access to Requests for Change" on page 95
- "Configuring Access to Business Change Requests" on page 96
- "Configuring Access to Development Change Requests" on page 97
- "Configuring Access to Deployment Units" on page 100

Configuring Access to Requests for Change

The default Request for Change (RFC) provider connection provides an integration between an RFC in Serena Release Manager and a related change request in Serena Service Manager.

This section tells how to configure RFC access from Serena Service Manager. For information on customizing your Serena Release Manager implementation to use providers for RFCs other than Serena Service Manager, see "Adding Provider Connections" on page 138.

Designate RFC provider information as follows:

1. Designating the Details for Each RFC Provider
2. "Telling Release Manager Which Providers to Use" on page 105

Designating the Details for Each RFC Provider

You should give the connection details for each RFC 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 RFC 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 properties files. For example:

   - `itsm.properties`

3. The details are unique for each properties file, and variables and parameters are defined in the implementation for the provider. See the example following this procedure.

4. After updating the provider properties files, restart the Serena common Tomcat service.

Designating RFC Provider Details for Serena Service Manager (SSM)

Example

Set the SSM provider details for RFCs. This example sets the values needed to retrieve requests for change from the SSM sample database.

```
itsm.properties
```

```
# rfc provider definitions
itsm.provider.name=Itsm
itsm.provider.description=ITSM Request Provider for Change system

itsm.table.tableName=TSM_CHANGEREQUEST
itsm.transition.update=CHANGE_MANAGEMENT.UPDATE1
itsm.transition.assignedRlm=CHANGE_MANAGEMENT.ASSIGNED_VIA_RLM
itsm.transition.implementedRlm=CHANGE_MANAGEMENT.IMPLEMENTED_VIA_RLM
itsm.transition.assignedRlm.type=Execute
itsm.transition.implementedRlm.type=Close

# rfc item fields
itsm.table.field.issueId=ISSUEID
itsm.table.field.state=STATE
itsm.table.field.relatedReleaseTrainId=LINKED_RELEASE
itsm.table.field.related=LINKED_TO_RELEASE

# rfc item states
itsm.defaultState=Approved Changes, Approved
```

NOTE If you are using SSM as your RFC provider, see the following related documentation:

- Chapter 8, "Customizing the SSM Integration" on page 151
- In the *Serena Service Manager User’s Guide,* "Serena Release Manager Integration".

Configuring Access to Business Change Requests

A Business Change Request (BCR) provider is an integration between Serena Release Manager and a change tracking system. The default implementation includes provider connections for SBM so that you can also associate Serena Release Manager BCRs with SBM issues.

This section tells how to configure BCR access from SBM. For information on customizing your Serena Release Manager implementation to use providers for BCRs other than SBM , see "Adding Provider Connections" on page 138.

Designate BCR provider information as follows:

1 Designating the Details for Each BCR Provider

2 "Telling Release Manager Which Providers to Use" on page 105

Designating the Details for Each BCR Provider

You should give the connection details for each BCR 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 BCR 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 properties files. For example:
   - `bcr.properties`
3. The details are unique for each properties file, and variables and parameters are defined in the implementation for the provider. See the example following this procedure.
4. After updating the provider properties files, restart the Serena common Tomcat service.

Designating BCR Provider Details for Serena Business Manager (SBM)

Example

Set the SBM provider details for BCRs. This example sets the values needed to retrieve business change requests from the SBM sample database.

`bcr.properties`

```
# bcr provider definitions
bcr.provider.name=Business Change Request  system
bcr.provider.description=Business Change Request system

bcr.table.tableName=TSM_CHANGEREQUEST
bcr.table.field.issueId=ISSUEID
bcr.table.field.state=STATE

bcr.defaultState=Approved Changes, Approved
```

Configuring Access to Development Change Requests

A Development Change Request (DCR) provider is an integration between Serena Release Manager 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.

This section tells how to configure DCR access from SBM and Dimensions CM. For information on customizing your Serena Release Manager implementation to use providers for DCRs other than SBM and Dimensions CM, see "Adding Provider Connections" on page 138.

Designate DCR provider information as follows:

1. Designating the Details for Each DCR Provider
2. "Telling Release Manager Which Providers to Use" on page 105

Designating the Details for Each DCR Provider

You should give the connection details for each DCR 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 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 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 the example following this procedure.

4. After updating the provider properties files, restart the Serena common Tomcat service.

Related Topics

- Designating DCR Provider Details for Dimensions CM Requests
- "Designating DCR Provider Details for SBM Issues" on page 99
- "Designating DCR Provider Details for SBM Incidents" on page 99

Designating DCR Provider Details for Dimensions CM Requests

Example

Set the Dimensions CM provider details for DCRs. This example sets the values needed to retrieve requests 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
.
.
.
# filter requests by statuses
FILTER_REQUEST_BY_STATUSES = IN QA,IN PROGRESS,UNDER WORK,IN TEST
.
.
.
```

The text following the keys, `requests.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 "Configuring Communication on the Dimensions CM Server" on page 56.

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

Designating DCR Provider Details for SBM Issues

Example

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  = DCR_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.

Designating DCR Provider Details for SBM Incidents

Example

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 Manager 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 incidents to associate with DCRs in a release package.

Configuring Access to Deployment Units

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

This section tells how to configure DU access from Dimensions CM and ChangeMan ZMF. For information on customizing your Serena Release Manager implementation to use providers for DUs other than Dimensions CM, see "Adding Provider Connections" on page 138.

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

1. "Designating the Details for Each DU Provider" on page 101
2. "Telling Release Manager Which Providers to Use" on page 105

Designating the Details for Each DU Provider

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

The properties include provider name, which points to an associated file with connection information, and filters by status, so that users see only relevant information about the deployment units in Serena Release Manager.

To designate details for each 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 properties files. For example:

   - `dm_qlarius.properties`
   - `zmf_packages.properties`

3. The details are unique for each properties file, and variables and parameters are defined in the implementation for the provider. See Chapter 8, "Adding Provider Connections" on page 138 for details if you plan to use a provider other than Dimensions CM or ChangeMan ZMF for DUs.

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

4. After updating the provider properties files, restart the Serena common Tomcat service.

Related Topics

- Designating Dimensions CM Deployment Unit Selection Criteria
- "Designating ChangeMan ZMF Deployment Unit Selection Criteria" on page 102

Designating Dimensions CM Deployment Unit Selection Criteria

Set the Dimensions CM provider details for deployment units, or baselines, using the example `dm_qlarius.properties` file or a custom Dimensions CM provider properties file.

Example

Set the Dimensions CM provider details for DUs. This example sets the values needed to retrieve 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, and is also used in the related UI report search filter. In this example, `DIM_QLARIUS` 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 "Configuring Communication on the Dimensions CM Server" on page 56.

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

Designating ChangeMan ZMF Deployment Unit Selection Criteria

If you plan to use ChangeMan ZMF to deploy change packages as your deployment units, you must specify the selection criteria for the change packages. Set the ChangeMan ZMF provider details for deployment units, or change packages, using the example `zmf_packages.properties` file or a custom ZMF provider properties file.

Set the properties for the ZMF filters as follows:

Parameter	**Value**
`deploy.units.provider.name`	Deployment unit provider name. Default value = `ZMF_PACKAGES`
`deploy.units.provider.description`	Deployment unit provider description. Default value = `ChangeMan Deployment Unit Provider for ZMF packages`
`FILTER_ZMF_PACKAGES_IN_DEVELOPMENT_STATUS`	Change packages that are in frozen status are typically selected as deployment units that are ready to install, but you may include change packages in development status if your processes require that. Setting this to Y will list all packages that are in FRZ and DEV status. Setting this value to N will list all packages that are in FRZ status. Default value = N

Parameter	Value
FILTER_ZMF_PACKAGES_BY_WORKREQNO	Setting this to Y will list only packages that have an empty or null work request number value. Default value = Y **NOTE** If you want to update the ZMF change package work request number value with the associated Serena Release Manager deployment unit's SBM item ID, you should set this value to Y. See RELATE_DU_ITEMID_TO_ZMF_PACKAGE_WORKREQNO.
FILTER_PLANNED_PERMANENT_ZMF_PACKAGES	Include change packages with package type planned permanent. (Values Y or N) Default value = Y
FILTER_PLANNED_TEMPORARY_ZMF_PACKAGES	Include change packages with package type planned temporary. (Values Y or N) Default value = Y
FILTER_UNPLANNED_PERMANENT_ZMF_PACKAGES	Include change packages with package type unplanned permanent. (Values Y or N) Default value = Y
FILTER_UNPLANNED_TEMPORARY_ZMF_PACKAGES	Include change packages with package type unplanned temporary. (Values Y or N) Default value = Y
FILTER_SIMPLE_ZMF_PACKAGES	Include change packages with the level of simple. (Values Y or N) Default value = Y
FILTER_PARTICIPATING_ZMF_PACKAGES	Include change packages with the level of participating. (Values Y or N) Default value = Y
FILTER_ZMF_PACKAGES_BY_PROMOTION_LEVEL	Include change packages with this promotion level and above. The last promotion level must be *greater than or equal to* the promotion level filter. For example, if you have the following promotion levels in ZMF, setting this value to 10 returns change packages in these promotion levels: 10 – INT Integration Test 20 – UAT User Acceptance Test 30 – PAT Production Acceptance Test Default value = 10
FILTER_ZMF_PACKAGES_BY_AUDIT_LEVEL	Include audit return code. The audit return code must be *less than or equal to* the audit level filter. Default value = 04

Parameter	Value
`RELATE_DU_ITEMID_TO_ZMF_PACKAGE_WORKREQNO`	Choose whether to fill in the ZMF change package work request number with the development change request value from Serena Release Manager. (Values Y or N) Default value = Y `RELATE_DU_ITEMID_TO_ZMF_PACKAGE_WORKREQNO`
`RELATE_RP_PROD_DEPLOY_DATE_TO_ZMF_PACKAGE_INSTALL_DATE`	Choose whether to fill in the ZMF change package installation date with the release package stage end date from Serena Release Manager. (Values Y or N) Default value = Y

Example

This example sets the values needed to retrieve deployment units (change packages) from a ChangeMan ZMF sample system.

`zmf_packages.properties`

```
# Deployment unit provider name and description.
deploy.units.provider.name = ZMF_PACKAGES
deploy.units.provider.description = ChangeMan Deployment Unit Provider for ZMF packages

# Include ZMF change packages in FRZ and DEV statuses (Y/N). Set this to Y to list packages that are in FRZ
     and DEV statuses. Set this value to N to list only packages that are in FRZ status.
FILTER_ZMF_PACKAGES_IN_DEVELOPMENT_STATUS = N

# Retrieve only ZMF change packages that have an empty or null work request numbers (Y/N). Use in conjunction
     with RELATE_DU_ITEMID_TO_ZMF_PACKAGE_WORKREQNO.
FILTER_ZMF_PACKAGES_BY_WORKREQNO = Y

# Include ZMF change packages with designated package types (Y/N).
FILTER_PLANNED_PERMANENT_ZMF_PACKAGES = Y
FILTER_PLANNED_TEMPORARY_ZMF_PACKAGES = Y
FILTER_UNPLANNED_PERMANENT_ZMF_PACKAGES = Y
FILTER_UNPLANNED_TEMPORARY_ZMF_PACKAGES = Y

# Include ZMF change packages with designated package levels (Y/N).
FILTER_SIMPLE_ZMF_PACKAGES = Y
FILTER_PARTICIPATING_ZMF_PACKAGES = Y

# Include ZMF change packages with this promotion level and above. The last promotion level must be greater
     than or equal to the designated promotion level value.
FILTER_ZMF_PACKAGES_BY_PROMOTION_LEVEL = 10

# Include ZMF change packages with this audit return code or below. The audit return code must be less than
     or equal to the designated audit level value.
FILTER_ZMF_PACKAGES_BY_AUDIT_LEVEL     = 04

# Fill in the ZMF change package work request number with the Release Manager deployment unit\u2019s SBM item
     ID.
RELATE_DU_ITEMID_TO_ZMF_PACKAGE_WORKREQNO = Y

# Fill in the ZMF change package installation date with the release package stage end date from Release
     Manager.
RELATE_RP_PROD_DEPLOY_DATE_TO_ZMF_PACKAGE_INSTALL_DATE = Y
```

The text following the keys, `deploy.units.provider.name`, is documentary, and is also used in the related UI report search filter. In this example, `ZMF_PACKAGES` simply describes the kind of data being retrieved. The actual connection to the ChangeMan ZMF system is defined in the `zmf.properties` file. See Chapter 4, "Configuring ZMF Communication in Release Manager" on page 63.

This example tells ChangeMan ZMF to return only deployment units, or change packages, that are in FRZ status, have a blank work request number, are in promotion level 10 or above, and have passed audit with a return code of 04 or less. The relationship filters are set so that the work request number and installation dates will be updated in ChangeMan ZMF based on information stored in Serena Release Manager.

You should set your selection criteria based on your organization's release management practices related to ChangeMan ZMF.

Telling Release Manager Which Providers to Use

After 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 provider properties file.

You can select one or more providers for each of the types of objects.

To specify the 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 appropriate provider keys variable to the 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 DU Provider" on page 101.

Example

This example shows providers selected for each of the object types, RFCs, BCRs, DCRs (requests), and DUs. The bold text indicates the specific provider key for deployment units.

`providers.properties`

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

# deploy units provider keys
deploy.units.providers.keys=dm_qlarius, zmf_packages

# rfc provider keys
rfc.providers.keys=itsm

# bcr provider keys
bcr.providers.keys=bcr
```

In the preceding example,

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

tells Serena Release Manager to use the `dm_qlarius.properties` file and the `zmf_packages.properties` file for DUs

```
rfc.providers.keys=itsm
```

tells Serena Release Manager to use the `itsm.properties` file for RFCs

```
bcr.providers.keys=bcr
```

tells Serena Release Manager to use the `bcr.properties` file for BCRs

```
requests.providers.keys = sbm_issues
```

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

Related Topics

- "Configuring Access to Requests for Change" on page 95
- "Configuring Access to Business Change Requests" on page 96
- "Configuring Access to Development Change Requests" on page 97
- "Configuring Access to Deployment Units" on page 100

Chapter 7

Serena Release Manager Upgrade

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

Upgrading from Serena Release Manager v2.0 to v2.1

If you need to upgrade Serena Release Manager from and to earlier versions, please see the documentation for your version of Serena Release Manager for recommended steps.

IMPORTANT! If you upgrade SBM to a version that is not supported by your version of Serena Release Manager, Serena Release Manager will no longer function properly. Please see the supported platforms for your version of Serena Release Manager as instructed in Chapter 1, "Software Compatibility Requirements" on page 17.

You can upgrade Serena Release Manager from version 2.0 to version 2.1 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	Actions
1: Backup	Back up data and product files. (Optional) It is a good practice to snapshot your system or back up existing files and data before beginning the upgrade. **1** At minimum, you should back up the data as follows: **a** Copy the files from the Serena Common Web services `rlm\WEB-INF\classes` folder, for example `C:\Program Files\Serena\common\tomcat\6.0\webapps\rlm\WEB-INF\classes`, to a temporary folder. **b** Backup your SBM database. **2** For an easier restoration if necessary, back up the product directories: **a** Copy the Serena Common Web services `webapps` folder, for example `C:\Program Files\Serena\common\tomcat\6.0\webapps`, to a temporary folder. **b** Copy the `Serena\Solutions\Release Control` folder, for example `C:\Program Files\Serena\Solutions\Release Control`, to a temporary folder.
2: Copy the Upgrade Package	Copy the upgrade package to the Release Control folder. **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 `RC3.3-Win32-Upgrade.zip`, to a temporary folder. **3** Extract the upgrade package to the `Serena\Solutions\Release Control` folder. **4** The following files should now appear under the `Release Control` folder: ■ `com.serena.rlm.sbm.shell.zip` ■ `solution` file: for example, `RLM_Solution_Pack-2.1.032.sln` ■ `war` files • `rlm.war` • `almzmf.war` • `almzmfalf.war` • `almzmfws.war`

Step	Actions

3: Configure Web Services	Configure the Web services files in 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 following folders under the `webapps` folder:
	`rlm`
	`zmf`
	`zmfalf`
	`zmfws`
	4 Delete the following `war` files under the `webapps` folder, if present:
	`rlm.war`
	`zmf.war`
	`zmfalf.war`
	`zmfws.war`
	5 From the `Release Control` folder where you extracted them in a preceding step, copy all of the `war` files to the Serena Common Tomcat `webapps` folder. For example:
	`C:\Program Files\Serena\common\tomcat\6.0\webapps`
	6 Restart the Serena Common Tomcat service.
	This war file contents are automatically extracted to new directories in that location. You should now see the following directories under `webapps`:
	`almzmf`
	`almzmfalf`
	`almzmfws`
	`rlm`

4: Copy the Solution File	Copy the solution file that contains the Serena Release Manager 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-2.1.032.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`

Step	**Actions**
5: Install the Shell Template	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. For any conflicts, select the option to replace with the newer versions. 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 SBM System Administrator, select **File | Put Files Into Database**. Confirm when prompted. This puts the UI shell files into the SBM database.
6: Import the Solution	Import the Release Manager solution. **1** Log into the SBM Application Repository as an SBM administrative user. **2** Navigate to the **Solutions** tab and import the solution. For example: `RLM_Solution_Pack 2.1.0.32` See Chapter 4, "Importing the Serena Release Control Solution" on page 37.
7: Promote the Snapshots	Promote the snapshots. **1** Navigate to the **Process App Snapshots** tab. **2** In SBM Application Repository, promote the snapshots. When promoting the snapshots, make sure to select the endpoints as needed, and make sure the endpoints are authenticated with Security Token. ▪ Release Train ▪ Application Release ▪ Release Package ▪ RLM_AUX ▪ Deployment ▪ Environment ▪ ReleaseTemplate **IMPORTANT!** Make sure to check for any warnings after you have promoted the snapshots. For information on analyzing warnings or errors, see Chapter 9, "Snapshot Promotion Errors" on page 162. For detailed instructions see Chapter 4, "Promoting the Snapshots" on page 40.

Step	Actions
8: Deploy the Process Apps	Deploy the process apps. **1** After the Serena Release Manager process apps are promoted to the correct environment, you must deploy the process apps from within SBM Composer. This validates the target endpoints prior to deployment to ensure that your environment is correctly configured. To redeploy the process apps: **a** In SBM Composer, publish each of the Serena Release Manager process apps. **b** After successfully publishing each process app, deploy each of the process apps. See Chapter 4, "Publishing and Deploying the Process Apps" on page 45. **NOTE** Follow the guidelines in the SBM documentation to deploy the Serena Release Manager process apps.
9: Reconfigure the Port	If you are changing the port on which the Serena Common Tomcat runs, reconfigure Serena Release Manager to use the new port number. See Chapter 8, "Configuring Release Manager to Use a Different Port" on page 143.
10: Set Privileges	Set privileges for the administrative user to the Serena Release Control objects, such as projects, reports, and tables. See Chapter 4, "Configuring the Administrative User Privileges" on page 47.
11: Enable Roles	Enable roles for Serena Release Control projects and verify that Serena Release Manager is activated. **1** Ensure roles are enabled for all Serena Release Control projects. See Chapter 4, "Enabling Serena Release Control Project Roles" on page 48. **2** Verify that Serena Release Manager is activated by entering the URL in your Web browser. For example: `http://rlmhost/tmtrack/tmtrack.dll?shell=rlm`

12: Configure the Dashboard	Configure the Dashboard page if needed and clear the Web browser and template caches.
	1 If your Dashboard page does not appear with the multi-view dashboard report, configure the dashboard. See Chapter 4, "Configuring the Dashboard Page" on page 49.
	2 Clear and refresh your Web browser cache to ensure that saved shell elements are no longer saved in your Web browser.
	a To clear the cache, choose the option in your browser to delete history and select cache from the options given.
	b To refresh the template cache, enter the following URL in your browser:
	`http://<hostname>/tmtrack/`
	`tmtrack.dll?AdminPage&command=ClearTemplateCache`
	where `hostname` is your Serena Release Manager host server name.

Step	Actions
13: Update the Registry	Update the registry to ensure the proper version of Serena Release Control is registered for future reference. (Optional) **1** 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 3.3.0. **NOTE** This registry location has all the information that you will see in Windows **Add/Remove programs** and **Programs and Features**.
14: Complete the Configuration	Configure the system, application objects, and providers as needed. See Chapter 4, "System Activation and Configuration" on page 33, Chapter 5, "Configuration and Administration of the Integrating Objects" on page 73, and Chapter 6, "Provider Configuration" on page 93. Changes for this release include the following: - The ZMF war files have been renamed as follows: zmf to almzmf zmfws to almzmfws zmfalf to almzmfalf - The properties for the clients and providers have been divided into two properties files: typically connection properties in `<product>-client-connection.properties` and settings specific to the client in `<product>-client.properties`. The connection file values may be set using the Serena Release Manager Configurator or updated manually. **CAUTION!** If you choose to restore values from configuration files you backed up, you should copy only your implementation-specific values into specific parameters to ensure that you don't introduce errors. These files have changed significantly in this release, so restoring by copying and replacing entire files is not recommended.

Chapter 8

Serena Release Manager Customization

This section 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.

⚠ **CAUTION!** Modification of transitions and states in Serena Release Manager 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 Manager are used by the underlying Web services and are referenced by JavaScript, and if these are modified without additional system changes, Serena Release Manager will no longer function.

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

This chapter includes the following topics.

Configuring the User Interface

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

Related Topics

- "Configuring the Serena Release Control Dashboard" on page 116
- "Configuring the Calendars" on page 118
- "Configuring the Inbox" on page 119
- "Configuring the Activity Page" on page 121
- "Configuring Views and Dialog Boxes" on page 121

NOTE This section includes typical user interface configuration. This does not include customization of the UI shell to include new form elements, which may be required when you add or change stages. See "Customizing the User Interface Custom Shell" on page 148.

Configuring the Serena Release Control Dashboard

Upon login to Serena Release Control, the Dashboard page is shown with the results of your SBM home page report. To change the view that is shown on the Dashboard page, you can configure your home page report in SBM.

What Can You Change?

You can change the reports displayed on the Dashboard page as follows:

- Select a different report to display.
- 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.

Selecting the Report for the Dashboard Page

Upon initial install or upgrade of Serena Release Control, your Dashboard page shows whatever page is set as your SBM home page. For example, for the SBM Sample database, the Issue Defect Management Task Page appears for the Dashboard page. You should typically change this to show the Serena Release Control Dashboard multi-view report for the Dashboard page.

To select the Dashboard multi-view report for the Dashboard page:

If your Dashboard page does not appear with the multi-view dashboard report, configure report to show on the Dashboard page 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**. Log out and back in again to bring in the new settings for the user if you are logged into Serena Release Control.

Configuring the Dashboard Multi-view Report

You can change the Serena Release Control Dashboard multi-view report shown on the Dashboard page so that it shows information that is relevant to your role in release management.

NOTE The Serena Release Control Dashboard does not use Serena Dashboard in the default implementation. Serena Dashboard is a highly configurable and multi-product dashboard that you can configure to report on your Serena Release Manager and other orchestrated ALM suites' data. Serena Dashboard is sold separately. For more information, see the Serena Dashboard documentation.

To configure the Dashboard multi-view report:

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.

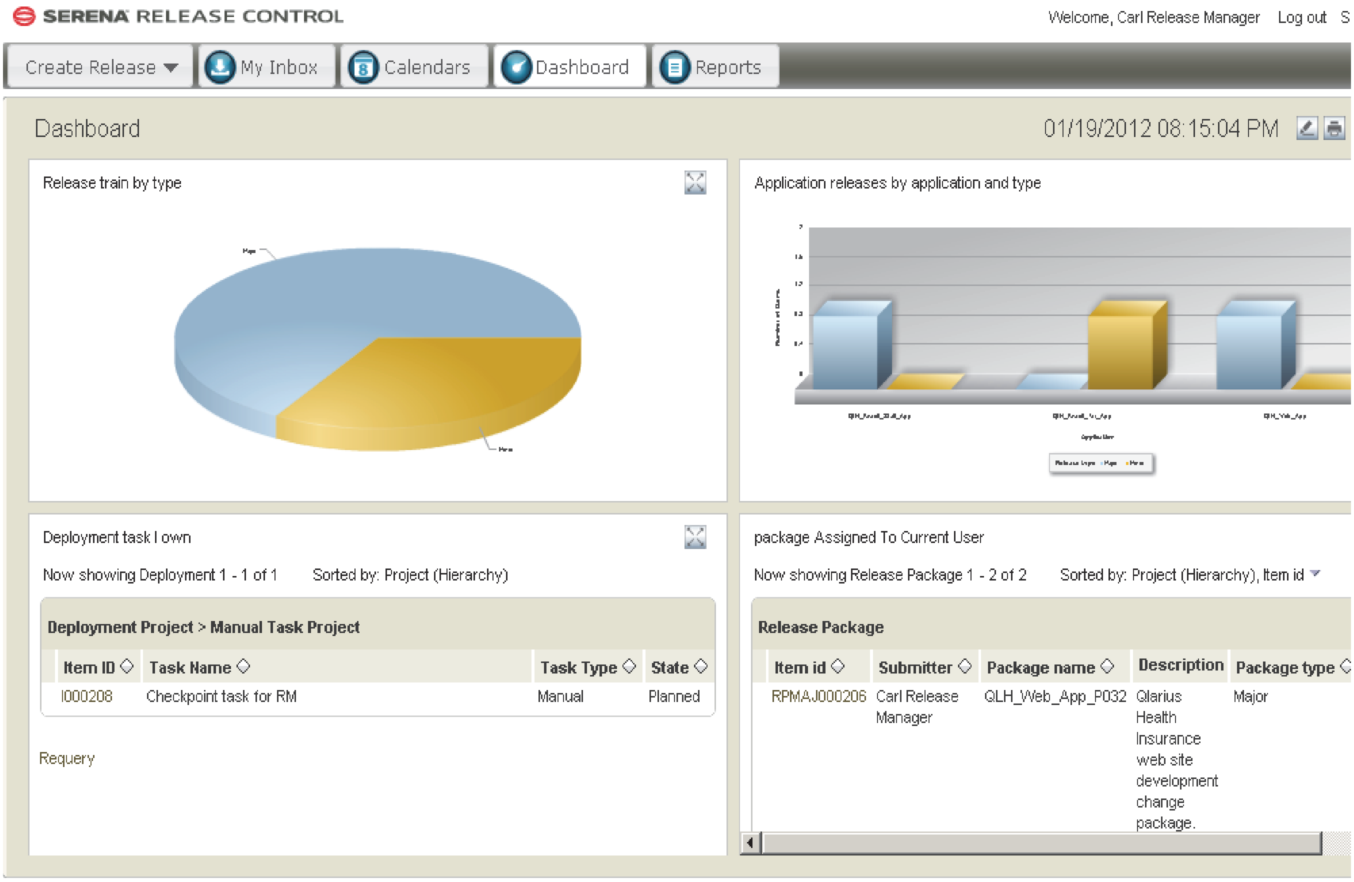
5 Click **Set as Home Page** if the dashboard report is not already set as your home page report.

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**, **Deployment tasks I own**, and **package Assigned to Current User** is shown in the following figure.



Dashboard

01/19/2012 08:15:04 PM

Release train by type

Application releases by application and type

Deployment task I own

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

Deployment Project > Manual Task Project

Item ID ◇	Task Name ◇	Task Type ◇	State ◇
I000208	Checkpoint task for RM	Manual	Planned

Requery

package Assigned To Current User

Now showing Release Package 1 - 2 of 2 Sorted by: Project (Hierarchy), Item id

Release Package

Item id ◇	Submitter ◇	Package name ◇	Description	Package type ◇
RPMAJ000206	Carl Release Manager	QLH_Web_App_P032	Qlarius Health Insurance web site development change package.	Major

Configuring the Calendars

Serena Release Control provides two calendar views, a Gantt view and a regular calendar view. Both view use the **Release Trains in Gantt** report by default.

What Can You Change?

You can change the information displayed on the calendars as follows:

- Change the report that populates the calendars.
- Change the HTML templates for the calendars.

What is the Impact?

If you change the calendar report or templates, the calendar pages change accordingly.

How Do You Change It?

- You can change the report filters to change the information shown on the calendar.
- You can change the HTML templates used by the report to change the appearance of the UI.

To change the Gantt view 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 template is:

   ```
   ganttview.htm
   ```

5. Change the report selections and the corresponding HTML template as needed.

NOTE When the calendar page is displayed in the shell, the HTML template used for the page changes dynamically from `ganttview.htm` to `calendar.htm` based on the selection of the associated button, **Gantt View** or **Calendar** View.

Documentation References

- Documentation on managing reports in SBM is in the *Serena Business Manager User's Guide* in "Working with Reports".
- Documentation on referencing reports in SBM is in the *Serena Business Manager SBM Composer Guide* in "Referencing a Report".

Configuring the Inbox

Serena Release Control uses SBM 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 different names.

CAUTION! You must change the report name by saving in place and retaining the report reference name. 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.
- These reports may be used for multiple views. You must 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, make your changes, and then click **Save**.

4 Overtype the existing name with a new name, making sure that you do not change the report reference name.

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	Report Reference Name
My Action Items	Release Trains	train Assigned To Current User	train.inbox
	Application Releases	application Assigned To Current User	application.inbox
	Release Packages	package Assigned To Current User	package.inbox
	Deployment Tasks	task Assigned To Current User	task.inbox
	Deployment Process Templates	Template Assigned To Current User	templates.owned.by.current.user
	Environments	Environments owned by Current user	environments.owned.by.current.user
Manage All Items	Release Trains	Train All	train.all
	Application Releases	Application All	application.all
	Release Packages	Package All	package.all
	Deployment Tasks	All DT	task.all
	Deployment Process Templates	Templates All	templates.all
	Environments	Environments All	environments.all

Documentation References

- Documentation on managing reports in SBM is in the *Serena Business Manager User's Guide* in "Working with Reports".
- Documentation on referencing reports in SBM is in the *Serena Business Manager SBM Composer Guide* in "Referencing a Report".

Configuring the Activity Page

Serena Release Control uses an SBM report to display release package deployment status information on your **Activity** page.

What Can You Change?

You can change the **Activity** page as follows:

- Change the report.
- Select different data to be displayed in the activity report.
- Change the color-coding from the defaults of red for Failed state and green for Production or Completed state.
- Change the display name for the **Activity** tab.

What is the Impact?

- If you change the activity report, the list of items in the **Activity** page changes according to the new report criteria.
- If you change the name of the report incorrectly, the **Activity** page will be displayed incorrectly.
- If you change the HTML files and don’t resolve all references properly, the **Activity** page will be displayed incorrectly.

How Do You Change It?

- In the SBM User Workspace you can change the report used to populate the **Activity** page.
- In the `activity.html` file you can select different data to be displayed in the activity log by adding new columns to the tabular report.
- In the `activity.html` file you can change the color-coding.
- In the `wrapper.html` file you can change the display name for the tab.

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 objects. 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. Embedded report widgets also provide a way to 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.

⚠ **CAUTION!** You must change the report name by saving in place and retaining the report reference name. 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.htm` layout.
- Most of these reports are used for multiple views. You must 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	Report Reference Name
Release Train	ApplicationReleasesInTrain	application.releases.in.release.train
	Assigned Requests for Change	RCF.assigned
Application Release	PackagesInAppRelease	release.packages.in.application.release
	Assigned Business Change Requests	BCR.assigned

View	Report	Report Reference Name
Release Package	DCR Projects Assigned to a Release Package	DCR.projects.assigned.to.RP
	DU Projects Assigned to a Release Package	dimCM.projects.assigned.to.RP
	ZMF Projects Assigned to a Release Package	ZMF.projects.assigned.to.RP
	Assigned Development Change Requests	DevChRequest.assigned
	Assigned Deployment Units	DepUnits.assigned
	Assigned Tasks	Task.assigned
Deployment Process Template	Template Tasks	task.template

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 Manager, you can customize the workflow states and transitions in SBM that support your organization's release control processes.

Serena Release Manager 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 Manager workflow accordingly.

What Can You Change?

- Add workflow state and transition information
- Change workflow state and transition information

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 125.

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 so that the UI shell displays 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 objects in SBM Composer according to the SBM documentation.
- Change the auxiliary table entries in 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:

- Change release types
- Change stages
- Add release types
- Add stages
- Delete release types
- Delete stages

What is the Impact?

- Release types and stages are interrelated, and if you change one aspect, you must 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 Manager 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 the release package deployment process fails. 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 single selection field, TASK_STATUS, 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 **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 change 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 forms:
   - createReleaseTrain
   - viewReleaseTrain
3. Edit **Attributes visibility** in the Visual Design JavaScripts to specify when to show or hide the new date fields.
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 **Package type** single-selection field value.

c Add a rule for the single-selection field.

d Add a transition for the deploy decision.

e Add the new rule to the deploy decision box.

Example of Adding a Stage

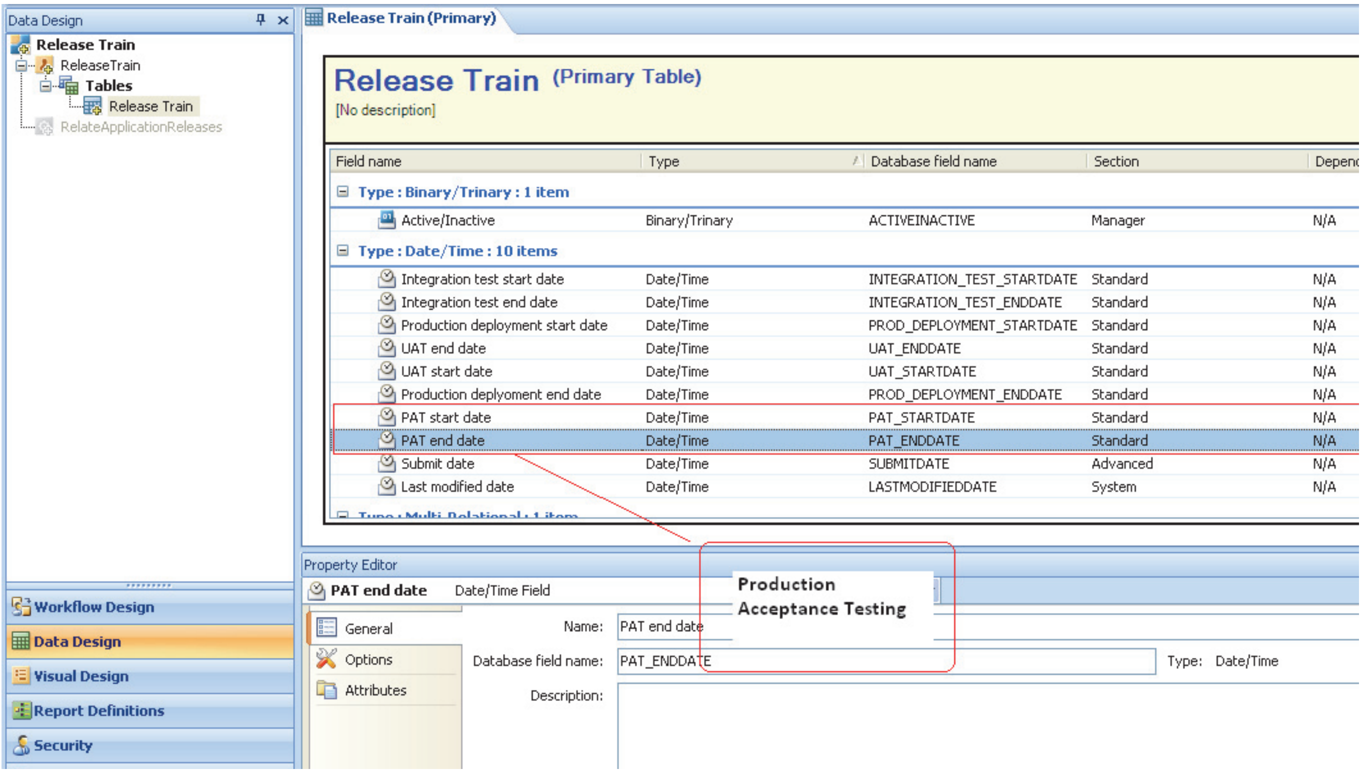
This example shows how to add a new stage that's associated with a new Release Train Start and End Date and a new Release Package stage process. This example adds a stage of PAT, PAtch Test. This stage is defined to be used only when the release type is Patch.

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
⊟ Type : Multi-Relational : 1 item				

Property Editor

PAT end date Date/Time Field

Name: PAT end date

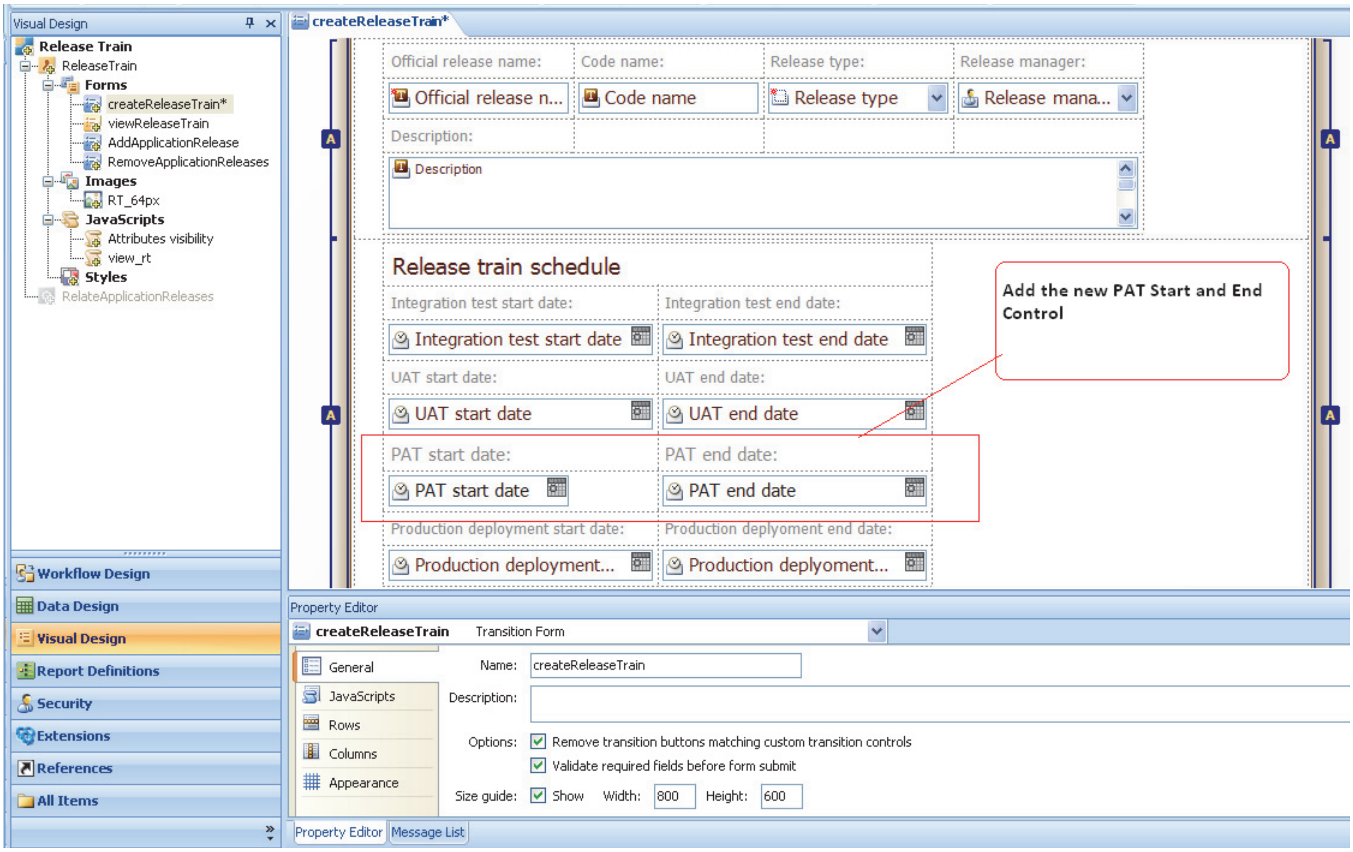
Database field name: PAT_ENDDATE

Type: Date/Time

Description:

Production Acceptance Testing

2 Add the Start and End field controls in the related forms, **createReleaseTrain** and **viewReleaseTrain**. The changes for **createReleaseTrain** are shown in the following figure.



Add the new PAT Start and End Control

createReleaseTrain Transition Form

Name: createReleaseTrain

Description:

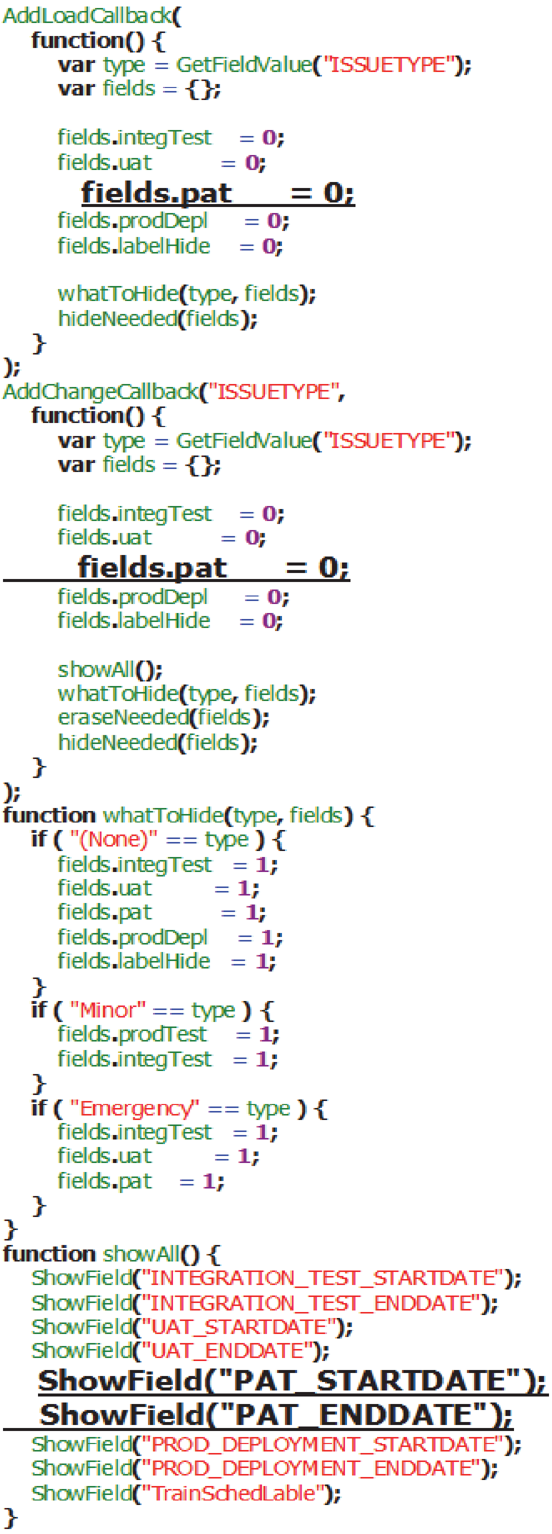
Options: ☑ Remove transition buttons matching custom transition controls

☑ Validate required fields before form submit

Size guide: ☑ Show Width: 800 Height: 600

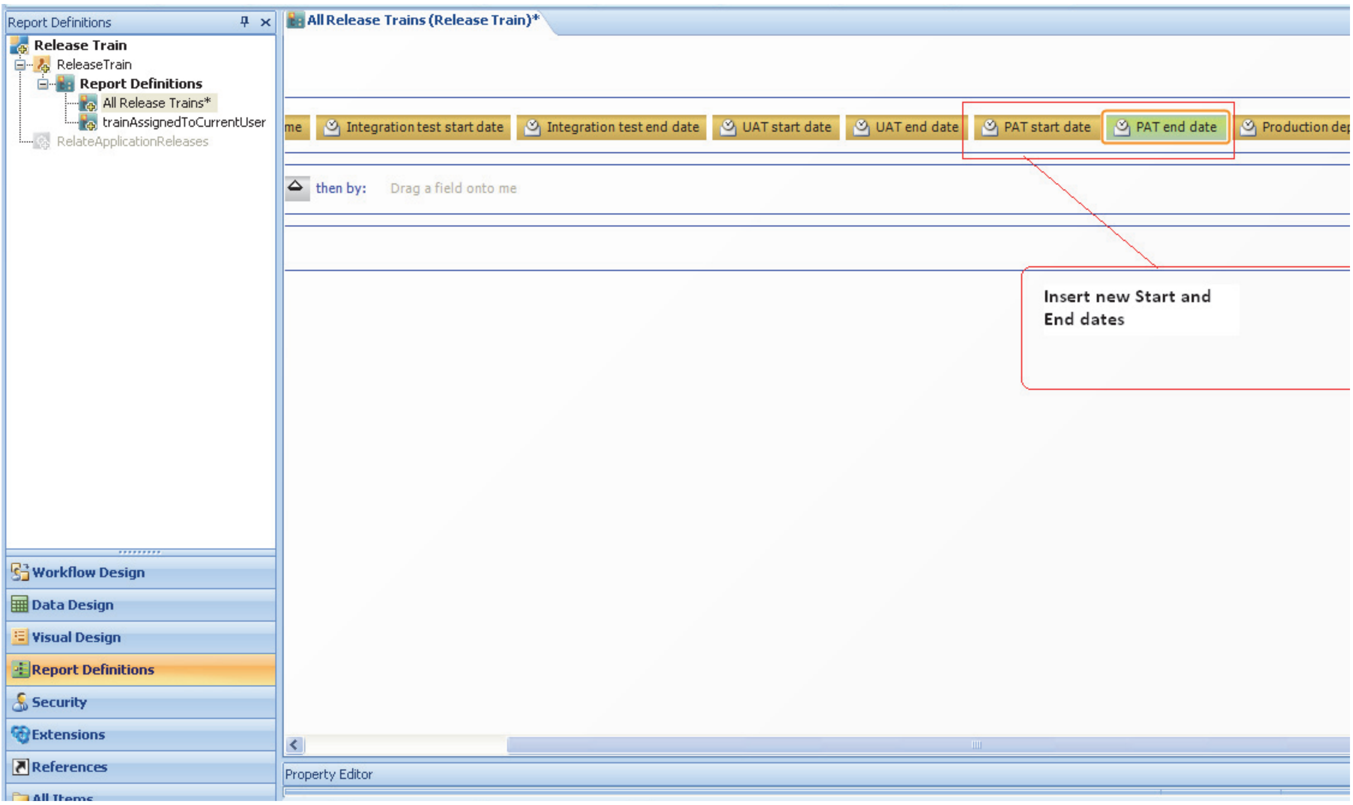
3 Edit **Attributes visibility** in the Visual Design JavaScripts to specify when to show or hide the new date fields.

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

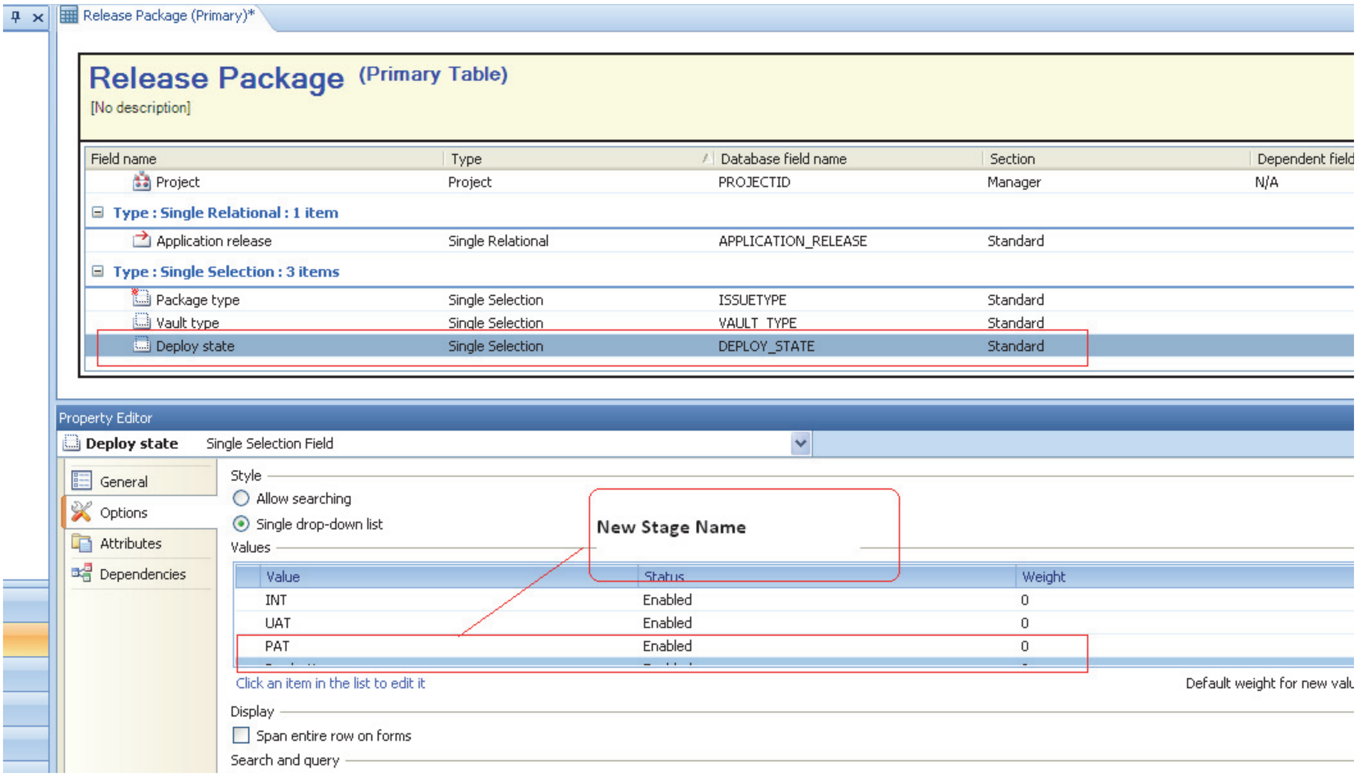


Insert new Start and End dates

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 **DEPLOY_STATE** single selection field as shown in the following figure.



Release Package (Primary)*

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	

Property Editor

Deploy state Single Selection Field

Style

○ Allow searching

● Single drop-down list

Values

New Stage Name

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

Click an item in the list to edit it

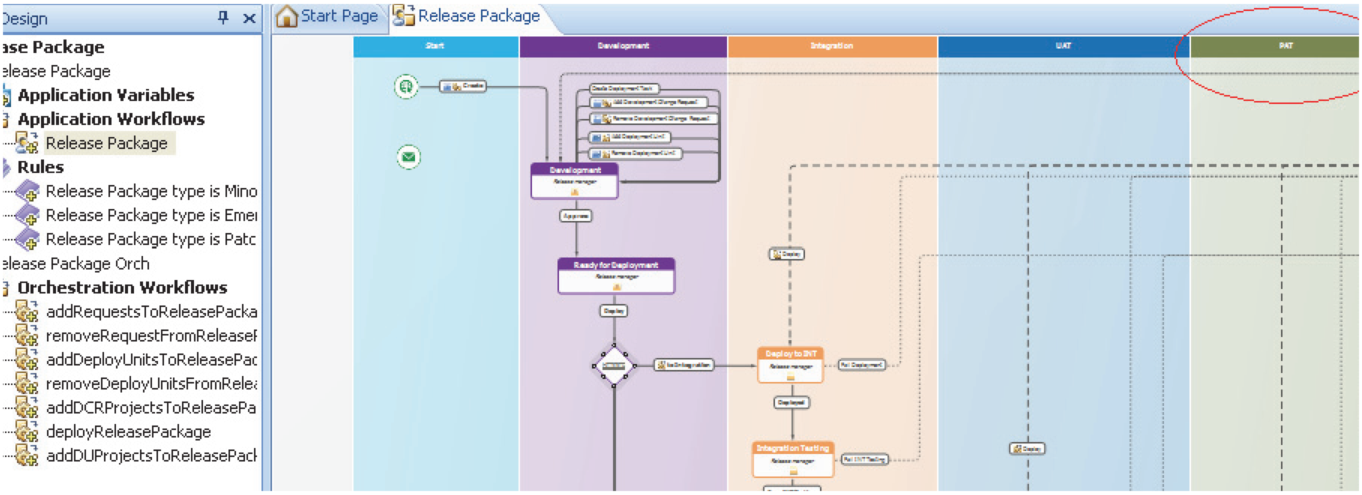
Default weight for new valu

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.

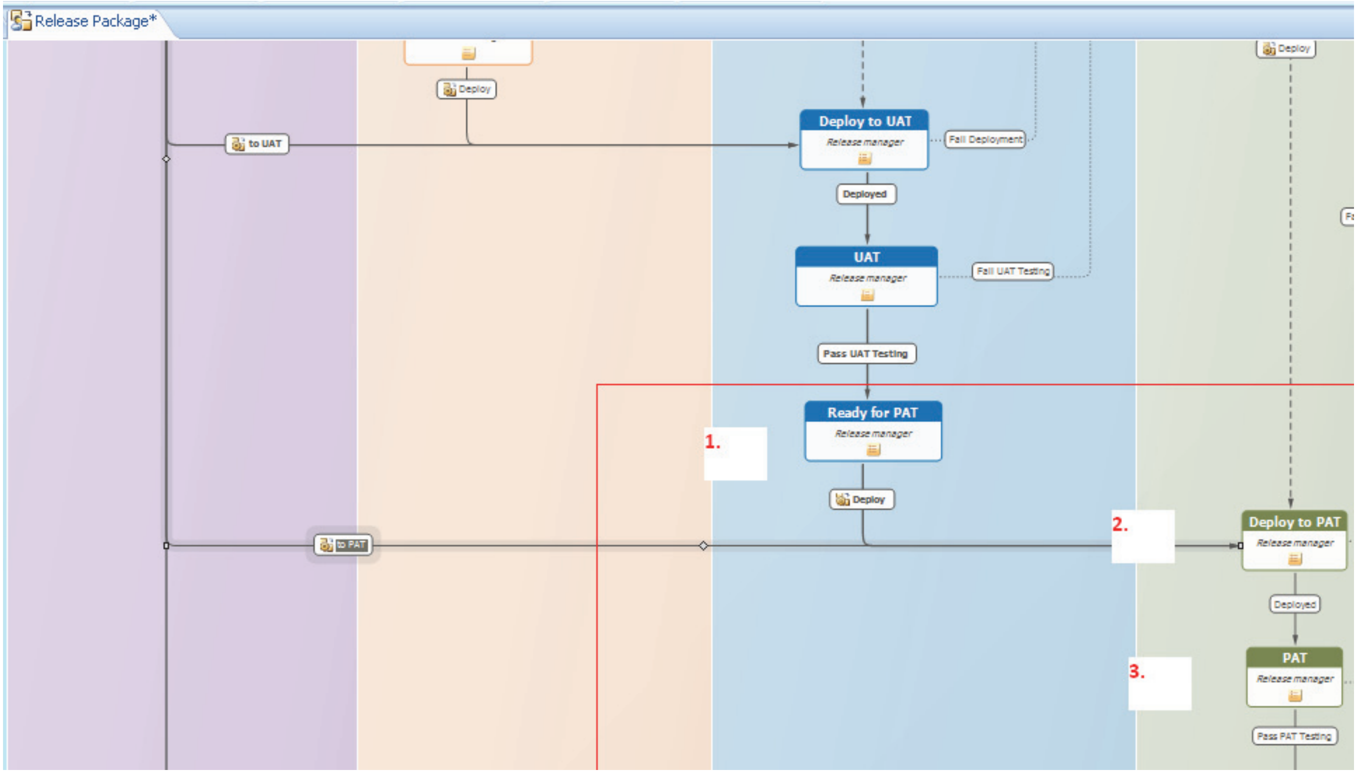


ase Package
elease Package
Application Variables
Application Workflows
Release Package
Rules
Release Package type is Mino
Release Package type is Emer
Release Package type is Patc
elease Package Orch
Orchestration Workflows
addRequestsToReleasePacka
removeRequestFromReleaseF
addDeployUnitsToReleasePac
removeDeployUnitsFromRele
addDCRProjectsToReleasePa
deployReleasePackage
addDUProjectsToReleasePacl

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.

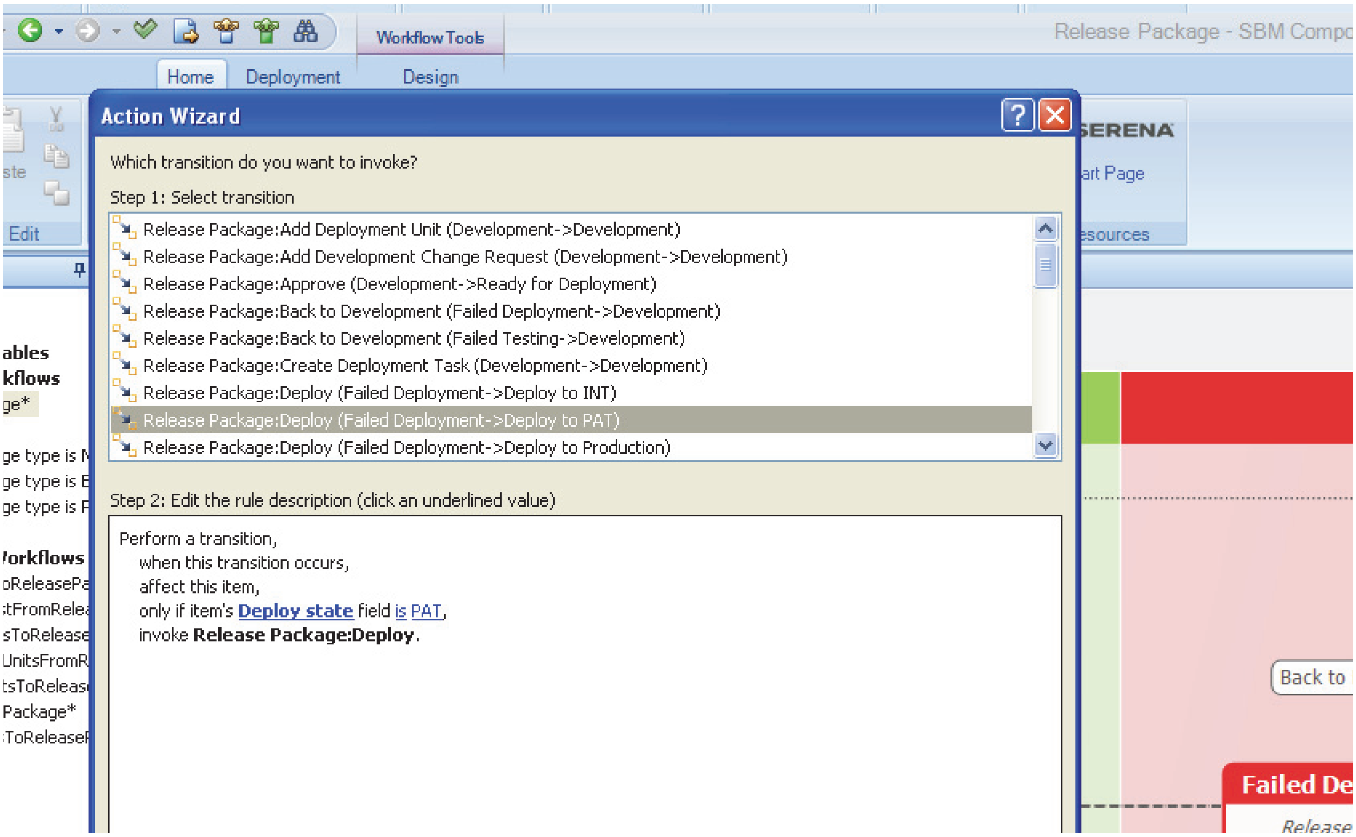
State	Transition	Option	Selection
From state: Failed Deployment To state: Deploy to PAT	Deploy	Options	Quick transition and Hide button on form
		Form	None
		Field Privileges	default values
		Field Overrides	Failed Deploy Transition ▪ Read Only ▪ Set to default: Fail Deployment
		Actions	Invoke deployReleasePackage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

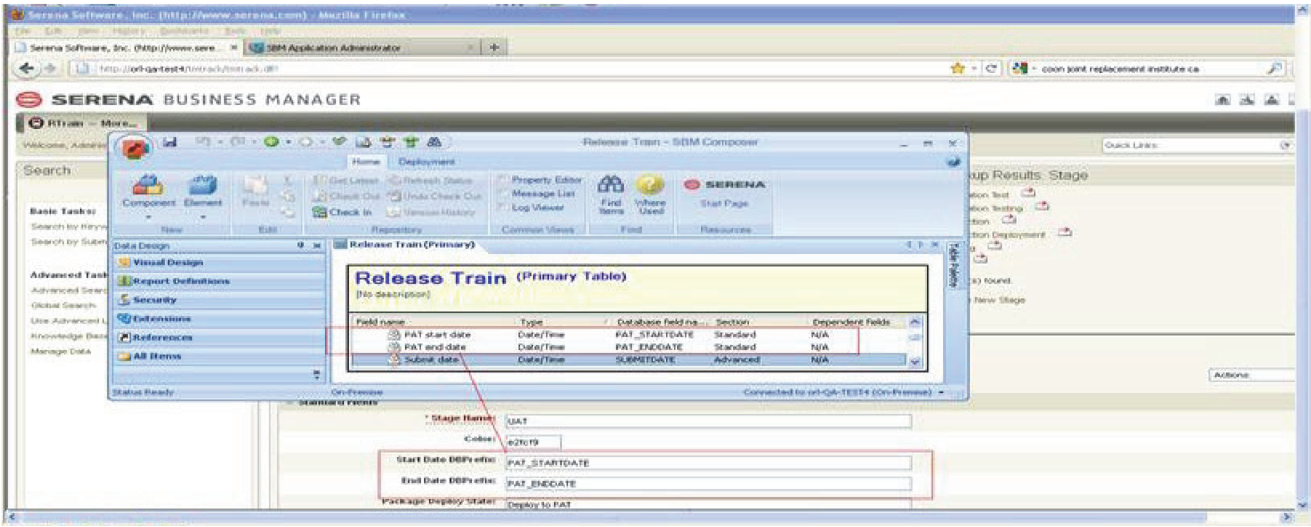
- Release Package:Add Deployment Unit (Development->Development)
- Release Package:Add Development Change Request (Development->Development)
- Release Package:Approve (Development->Ready for Deployment)
- Release Package:Back to Development (Failed Deployment->Development)
- Release Package:Back to Development (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.



Release Train (Primary)

Release Train (Primary Table)
[No description]

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

Standard Fields

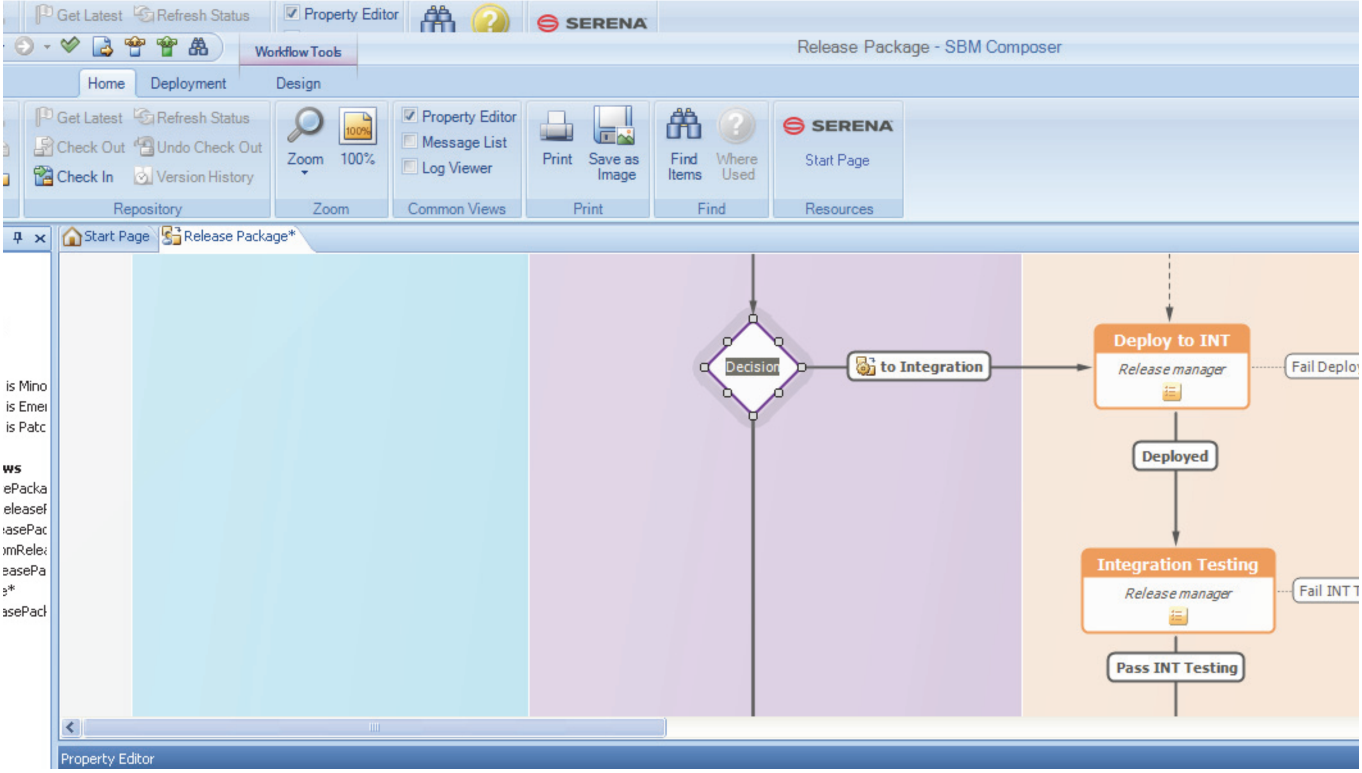
* Stage Name: UAT
Color: e2fc f9
Start Date DBPrefix: PAT_STARTDATE
End Date DBPrefix: PAT_ENDDATE
Package Deploy State: Deploy to PAT

- 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.

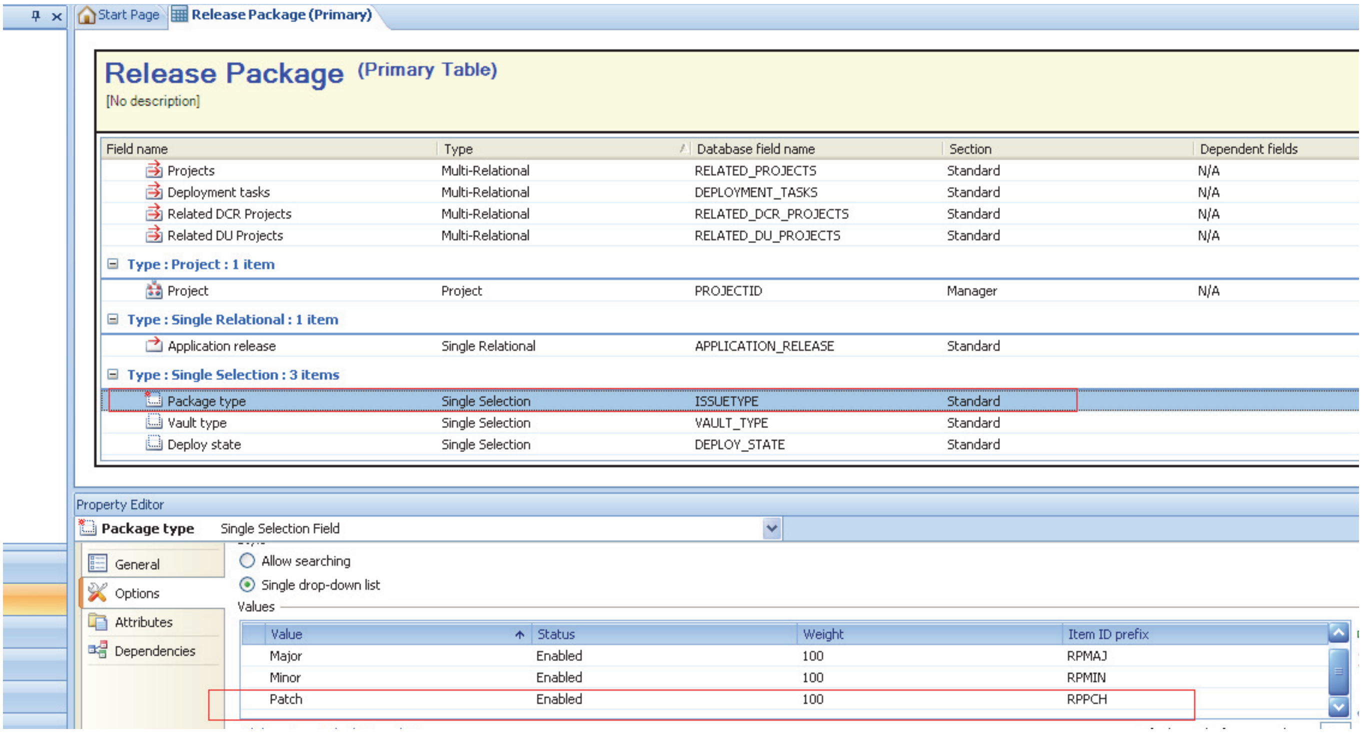
8 Optionally add a deploy decision rule to the Release Package stage process.

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 **Package type** single-selection field value as shown in the following figure.



Start Page | Release Package (Primary)

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	

Property Editor

Package type — Single Selection Field

General | Options | Attributes | Dependencies

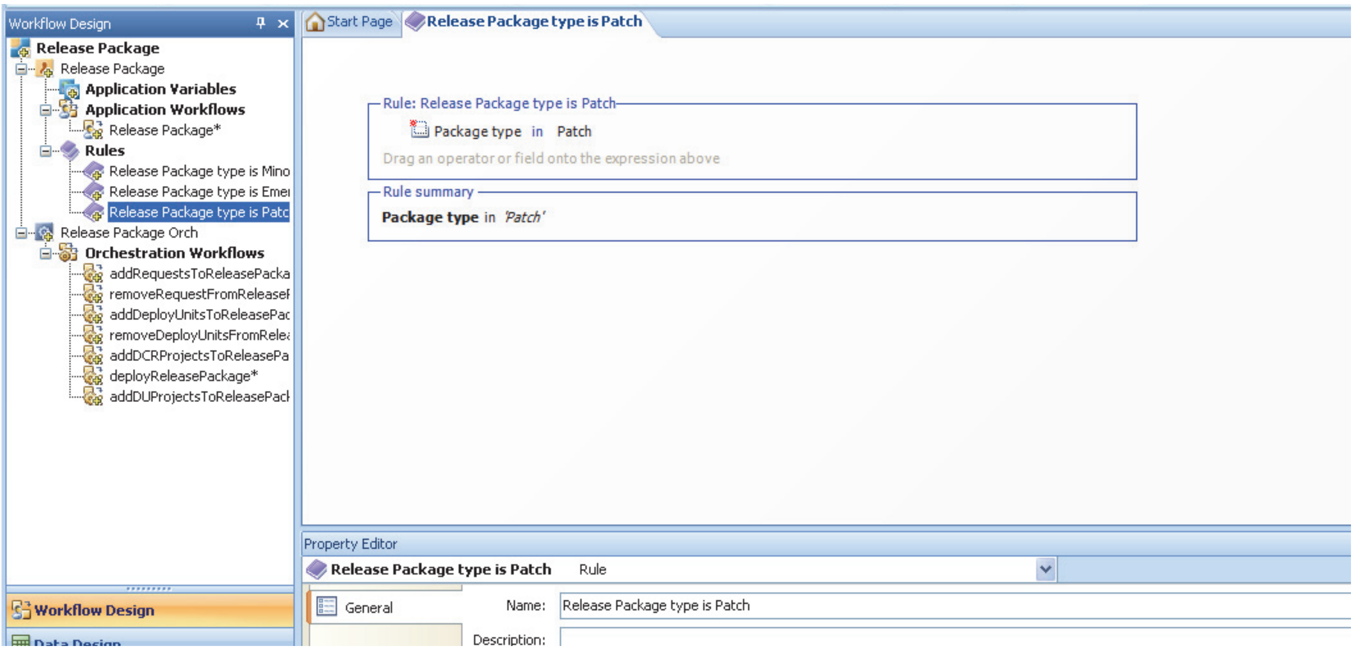
○ Allow searching

● 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 Emei
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

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

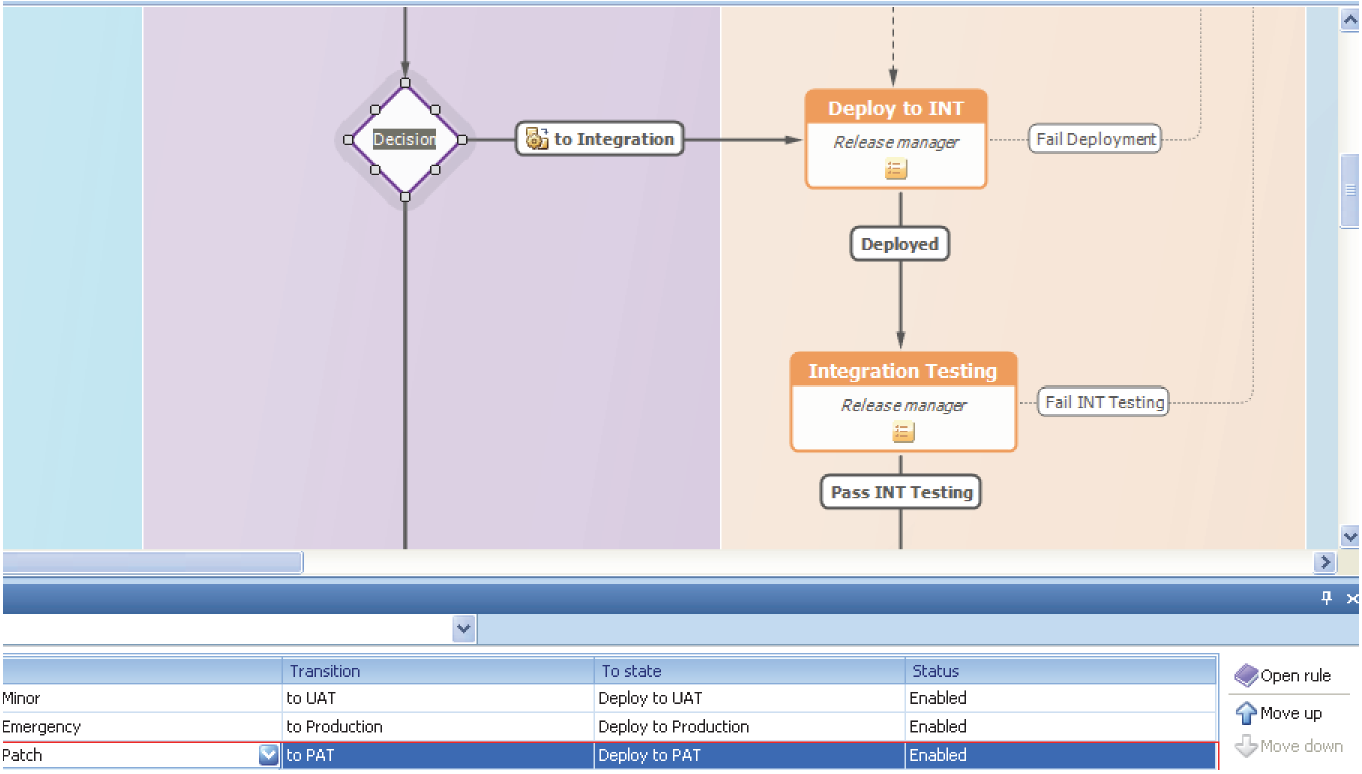
Name: Release Package type is Patch

Description:

d Add a 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 Provider Connections

Serena Release Manager uses provider connections to make it easier to extend the product integrations with the suite. You can extend the integration to use other systems as providers using the Serena Release Manager configurable provider connection method.

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

To see if a provider connection you want 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 providers, see the following sections:

- "Creating a Class for Your Provider" on page 139
- "Creating Properties Files for Your Providers" on page 139
- "Building and Packaging" on page 142

- "Telling Serena Release Manager to Use This Provider" on page 142

NOTE For documentation on configuring existing provider connections, see Chapter 6, "Provider Configuration" on page 93.

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 139.
- Define all instance-specific values for parameters in a properties file. See "Telling Serena Release Manager to Use This Provider" on page 140.

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.

For example, Serena provides the `provider-dm.xml` file for Dimensions CM, a potential provider of DCRs and DUs and `provider-sbm.xml` file for SBM, a potential provider of RFCs, BCRs, and 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 Manager 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, use 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, enabling you to 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 Manager server.

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 Manager to Use This Provider

After you have built and packaged your new provider, you tell Serena Release Manager 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
```

Configuring Release Manager to Use a Different Port

Serena Release Manager 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.

- Serena Release Manager 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.

- Use the default installation and configuration procedures on the Serena Release Manager 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.
- If you want to use a different port for Serena Release Manager, you must change the port number as shown in Checklist for Changing the Port Number.

IMPORTANT! The port change procedures do NOT replace or supersede the required installation and configuration procedures. Although it is possible to do the port changes as you are installing, these procedures assume that you have already completed the installation and system configuration procedures elsewhere in this document before beginning the port change procedures.

Related Topics

- Checklist for Changing the Port Number

Checklist for Changing the Port Number

Procedure	Reference
❑ Install and configure the Serena Common Tomcat Web server and Serena Release Control as usual. If you have already done this, you do not need to do this again!	See Chapter 3, "Installing Serena Release Control" on page 28 and Chapter 4, "System Activation and Configuration" on page 33.
❑ Rerun the Serena Release Control installer at the command line with parameters to change the port the Serena Common Tomcat Web server runs on.	See "Changing the Port on Which the Common Web Server Runs" on page 145.
❑ Configure the endpoints for the snapshots to point to the non-default port, and then promote the snapshots and deploy the process apps.	See Chapter 4, "Promoting the Snapshots" on page 40.
❑ Configure all RESTgrid widgets in the process app forms to point to the non-default port and then redeploy the process apps.	See "Configuring a Non-Default Web Server Port in the Process Apps" on page 145.
❑ Change the port to which the Web Services WSDLs point and re-import the WSDLs.	See "Changing the Web Services to Point to a Different Port" on page 147.
❑ When you are finished, publish and redeploy 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.	See Chapter 4, "Publishing and Deploying the Process Apps" on page 45.

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 Manager 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 change your Web server installation with the specified port.
5. Continue with the next checklist item in "Checklist for Changing the Port Number" on page 144.

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 must change the port numbers in the Serena Release Manager process apps that have RESTgrid widgets. After you change the process apps, you need to redeploy them.

Changing the Port Number in the Forms

You must find the RESTgrid widgets in the forms, and update each occurrence of the default port number, 9095, to the port number for your Web server installation.

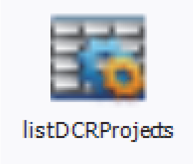
To change and redeploy the process apps:

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. For example, select `createRelPackage`. For a list of forms that you should update, see "Forms with RESTgrid Widgets" on page 146.

   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 RESTgrid widget and select it. For example, you'll see the RESTgrid 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 RESTgrid widget control in each process app. See Forms with RESTgrid Widgets for the list of default controls to change.

9 Continue with the next checklist item in "Checklist for Changing the Port Number" on page 144.

Forms with RESTgrid Widgets

The form controls in which you need to update the Web server port number are shown in the following table. This list is based on the default implementation of Serena Release

Manager and may not be an exhaustive list in your implementation, so it is good practice to look at each form to see if it uses RESTgrid widgets.

Process App	Forms	Controls
Release Package	addDCRProjects	listDCRProjects
	addDepUnit	listDimCMBaseline, listZMFPackages
	addDevChgRequest	listRequests
	addDimCM_Projects	listDimCMProjects
	addZMF_Projects	listZMFProjects
	createRelPackage, createRPfromAR	listDCRProjects, listDimCMProjects, listZMFProjects
Application Release	addBusinessChangeRequest	RESTGridWidget
Release Train	addRFC	listRfcRequests
Deployment	newAutomationTask, newAutomationTemplate, editAutomationTask, editAutomationTemplate	gridApps, gridEnv, gridProc
	newVaultDimDeployTask, editVaultDimDeployTask	gridDepUnits, gridDepStages, gridDepAreas
	newVaultZMFDeployTask, editVaultZMFDeployTask	gridDepUnits, gridDeploymentSites, gridPromotionAreas
	newVaultZMFApprovalTask, editVaultZMFApprovalTask	gridDepUnits, gridApprovers

Changing the Web Services to Point to a Different Port

If you are changing the port that Serena Release Manager runs under, you must change the port to which the Web Services WSDLs point and re-import the WSDLs.

To change and re-import the Web services WSDLs:

1. From SBM Composer, open one of the Serena Release Manager process apps.
2. In the navigation pane, click **Extensions**.
3. Under **Web Services**, select one of the following Serena Release Manager Web services, RLMUtilServices or ReleaseRequestService.
4. In the **Property Editor** beside the **WSDL** field, click **Reimport**.
5. Change the port number to the one you are using.
6. Click **OK** to re-import the updated Web service.
7. Repeat for each of the Serena Release Manager Web services in each of the process apps except RLM_AUX and Environments, which do not use endpoints.

8 Continue with the next checklist item in "Checklist for Changing the Port Number" on page 144.

Customizing the User Interface Custom Shell

The Serena Release Control user interface runs as a custom shell on top of the SBM native interface. The display details are implemented using an HTML file hierarchy, with a wrapper file, `wrapper.html`, defining the common display elements and tabs. The wrapper file references separate files that define the details of each page.

A custom shell for an SBM application is a set of customized templates, styles, scripts and images plus SBM server-side scripts that put these elements together. SBM uses a templating engine to embed tags, for example $TAGNAME(), to obtain environment information dynamically from a process app.

Serena supplies a default shell to use with the SBM-based Serena Release Control UI to provide a user-friendly look and feel. You invoke the shell when you access Serena Release Control by specifying `shell=customShellName` at the end of the URL, where `customShellName` is by default `rlm`. For example:

```
http://server:port/tmtrack/tmtrack.dll?shell=rlm
```

When you install Serena Release Control, there are a set of shell folders in the SBM application folder, for example:

```
C:\Program Files\Serena\SBM
```

Containing custom templates, style sheets, and scripts. These are:

- Templates:

  ```
  …\Application Engine\template\shell\customShellName
  ```

- Style Sheets:

  ```
  …\Application Engine\bin\styles\shell\customShellName\
  ```

- Images:

  ```
  …\Application Engine\bin\images\shell\customShellName\
  ```

- Scripts (if used):

  ```
  …\Application Engine\bin\javascript\shell\customShellName\
  ```

In a default installation of Serena Release Control, `customShellName` is rlm.

To create your own custom shell, you need to create a corresponding set of folders with the name of your custom shell, and invoke it by adding `shell=customShellName` at the end of the URL when accessing the Serena Release Control UI.

It is necessary to create the folder under `template` for your custom shell name, but it is optional for the other folders, as the process application will search in the top-level folders under `\bin` for styles, JavaScript and images. However it is recommended that you use these folders. You will need to provide relative paths for elements other than templates in your pages and styles.

When displaying a customized page, SBM will first look for the page inside:

`…SBM\Application Engine\template\shell\customShellName\`

If the page is not found there, the page from the default location:

`…SBM\Application Engine\template\"`

will be used.

For example, when SSO is disabled, SBM uses `loginform.htm` as the login page. This page is stored in

`…SBM\Application Engine\template\`

To create custom login screen, put your own version of `loginform.htm` into

`…SBM\Application Engine\template\shell\customShellName\`

Log In Page with SSO

If SSO is enabled, which is the default for Serena Release Manager, the SBM application behaves a little differently. The login screen in this case is built dynamically with `login.jsp`. To change it, the default page `ssocatalog.jsvar` should be created in

`...SBM\Application Engine\template\shell\customShellName\"`

For example:

```
$LOGINUI()({
  stylesheet: "styles/shell/dvm/login.css",
  "title": "Serena Release Control",
  introMsg: "",
  extraDiv1: "<div><a href='http://support.serena.com/'
    target='_tab'>Serena Online Support Community</a></div>",
  extraDiv2: "<div class='gray_bar'></div>"
})
```

Parameters you may set up from within `ssocatalog.jsvar` are:

```
stylesheet
title
logoLink
introMsg
useridHelp
passwordHelp
links (javascript array)
extraDiv1
extraDiv2,3,4
```

IMPORTANT! If there is some error in this file (for example, no comma after a parameter) the server will fail to parse it and will display the standard page, so it is recommended to limit the use of this dynamic generation.

Other Pages

It is also necessary to make sure that there is a version of the file `wrapper.htm` in the folder

```
…SBM\Application Engine\template\shell\customShellName\
```

because the SBM application will look for this file when the user logs in, and if it is not found the default page will be displayed.

You can change any other pages by putting templates with the same name in this folder. If SBM cannot find a template in this location, it takes the version in the default location:

```
…SBM\Application Engine\template\"
```

Applying the Changes

To apply any changes you have made, you must put the changes into the SBM database.

Applying Custom Shell Changes

1 Stop the Serena Common JBoss and IIS:

a Open SBM Configurator: Serena | Serena Business Manager | SBM Configurator

b Select the **Manage Services** tab on the left

c Click the **Stop** buttons for Serena Common JBoss and IIS.

2 Update database with the new images and settings:

a Open SBM system Administrator.

b Select **File** | **Put Files Into Database.**

c Click **OK** in the dialog box.

3 Start the Serena Common JBoss and IIS:

Open SBM Configurator, Select **Manage Services**, and click the **Start** buttons for Serena Common JBoss and IIS.

4 Clear your browser cache completely.

5 Refresh the template cache by entering the following URL in your browser:

```
http://<hostname>/tmtrack/
    tmtrack.dll?AdminPage&command=ClearTemplateCache
```

Activating Environment Association to Release Packages

Although you can create environments and report on their state in the default implementation of Serena Release Manager, you cannot associate the environments with release packages.

To activate this additional functionality, you can activate the underlying objects that have already been implemented. The JavaScript code is already included in the default implementation, so no shell changes are required.

To activate the existing environment form and functionality for release packages:

1. In SBM Composer, open the Release Package blueprint.
2. On the **Environments** tab form, add the button controls for **Add Environments** and **Remove Environments**.
3. In Serena Release Manager in the native SBM interface, create a report with the reference name that is referenced in the orchestration workflow.
4. Deploy the updated process app and test your changes.

Customizing the SSM Integration

You can associate SSM Change Management change requests with Serena Release Manager RFCs in the default implementation. Reports are configured to support the UI and provide relevant information about the RFC associations to release trains. Events in Serena Release Manager send information back to SSM and transition the related change requests as the release train moves through its workflow.

CAUTION! If you have the SBM Sample DB installed in the same instance of SBM as SSM, make sure to select SSM Changes from the Change Management process app rather than Change Requests from the SBM Change Request Management process app.

What Can You Change?

You can change the SSM integration as follows:

- Create the reports that SSM and Serena Release Manager use for the integration.
- Change the events within the release train workflow.
- Change the integration in SSM.
- Change the instance of SSM to which Serena Release Manager connects.

What is the Impact?

- If you change reports that impact the UI, the UI changes accordingly.
- If you change the reports or events used by SSM, you must change the related information in SSM to match the updated Serena Release Manager elements.
- If you change the release train workflow, you must assess and test the impacts for each affected transition and state.
- If you change references to Serena Release Manager elements in SSM, you must change the related information in Serena Release Manager to match the updated SSM elements.

How Do You Change It?

- You can create the reports in the SBM user workspace.

- For the Serena Release Manager-side changes, you can change the integration in SBM Composer in the Serena Release Manager RTrain process app.
- For the SSM-side changes, you can change the integration in SBM Composer in the SSM - Change Management process app.

Documentation References

- Documentation on using the default SSM-Serena Release Manager integration from SSM is in *Serena Service Manager ITIL Guide* in "Integration to Serena Release Manager".
- Documentation on using the default SSM-Serena Release Manager integration from Serena Release Manager is in *Serena Release Manager Getting Started Guide* in "Associating RFCs with Release Trains".
- Documentation on configuring workflow events in SBM is in the *Serena Business Manager Composer Guide* in "Applications".
- Documentation on managing reports in SBM is in the *Serena Business Manager User’s Guide* in "Working with Reports".

Related Topics

- "Creating the Reports Used for the SSM Integration" on page 152
- "Changing the Integration in SSM" on page 153
- "Changing the Instance of SSM that Release Manager Uses" on page 154

Creating the Reports Used for the SSM Integration

To activate the integration, you must create the auxiliary table reports in Serena Release Manager that SSM uses for the integration.

To create the Release Manager reports used by SSM:

1. In SBM Composer, check your SSM - Change Management blueprint to get the report reference names used in your implementation.
2. In Serena Release Manager in the native SBM interface, select the RTrain process app.
3. Create listing reports with the reference names from the SSM blueprint.

Example

a. Create a report to select all release trains in the planning state as follows:

Title:	All Planned Release Trains
Reference Name:	AllTrains-Planned
Privilege Category:	Guest
Report Project:	Base Project
Columns to Display:	Official release name
	Production deployment start date
	Production deployment end date

Include items from Sub-projects	(Select)
Use Basic Conditions:	State in Planning

b Create a report to select release trains within a selected date range as follows:

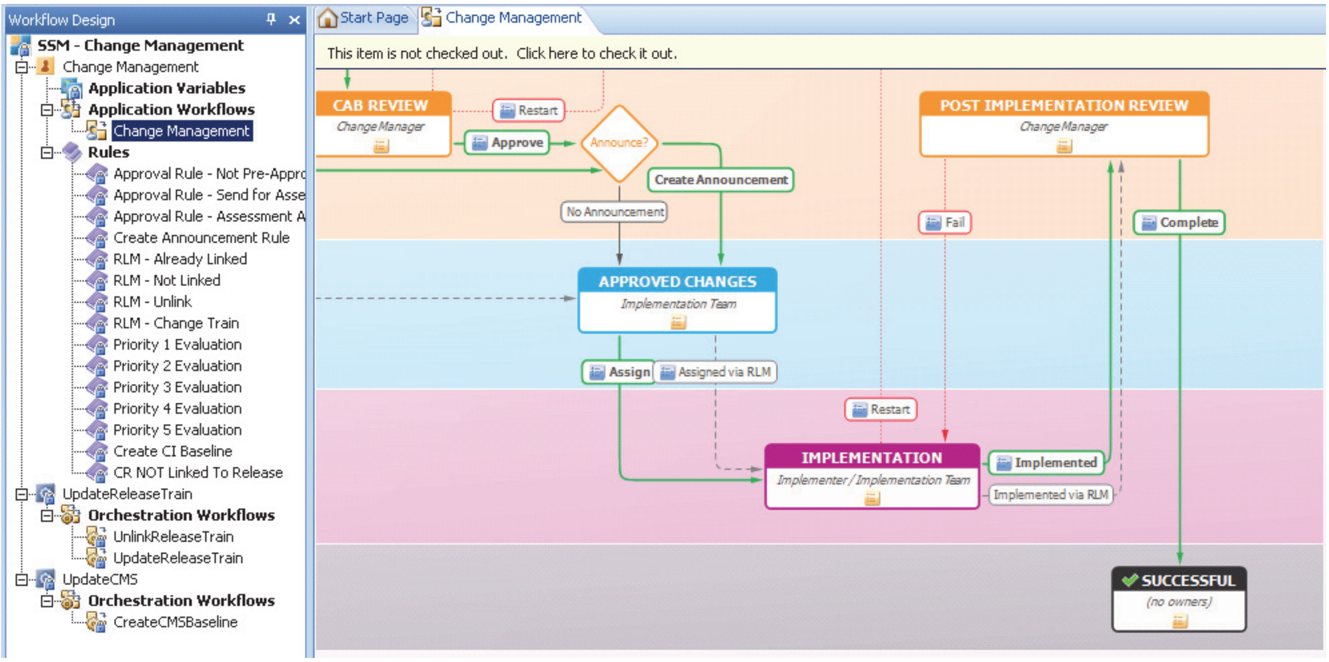
Title:	Suitable Release Trains
Reference Name:	Suitable_Trains
Privilege Category:	Guest
Report Project:	Base Project
Columns to Display:	Official release name Production deployment start date Production deployment end date
Include items from Sub-projects	(Select)
Use Basic Conditions:	State in Planning Production deployment start date = (Query At Runtime) Production deployment end date = (Query At Runtime)

Changing the Integration in SSM

You can change the integration to Serena Release Manager in SSM. Change the SSM integration according to the SBM and SSM documentation, along with your knowledge of the integration on the Serena Release Manager side.

Integration Points in the SSM Workflow

The SSM integration points are in the Change Management workflow as shown in the following figure.



The **Assigned via RLM** and **Implemented via RLM** transitions are implemented in the **Approved Changes** and **Implementation** states respectively. The transitions for linking a release train are available from any state. The full SSM Change Management workflow is shown in the *Serena Service Manager User's Guide*.

Related Topics

- "Configuring the User Interface" on page 116

Changing the Instance of SSM that Release Manager Uses

You can change the instance of SSM to which Serena Release Manager connects through the Serena Release Manager Configurator **RFC** page. In the default implementation, SSM detects that Serena Release Manager is installed if both are in the same instance of SBM.

Related Topics

- Chapter 4, "Configuring Connections using the Release Manager Configurator" on page 49
- Chapter 6, "Configuring Access to Requests for Change" on page 95

Setting Maximum Associations for Release Control Objects

To ensure that you don't overload your system by trying to deploy too many deployment units in one process, you can set maximum limits for the number of hierarchical associations for release trains, application releases, and release packages.

By default, Serena Release Manager sets the maximum number of hierarchical associations to 50 each. For example, application releases per release train, release packages per application release, and deployment tasks per release package.

Depending on the bandwidth of your servers and the demands your deployment processes make on the participating servers, you may choose to raise or lower the maximum limit.

To change the maximum values for the associations:

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 `sbm-client.properties` file.

3. Update the value of one or more of the following entries as needed:

   ```
   # URM_RELEASE_TRAIN
      release.train.maximum.application.releases=50
   #USR_APPLICATION_RELEASE
      application.release.maximum.release.packages=50
   # USR_RELEASE_PACKAGE
      release.package.maximum.deployment.tasks=50
   ```

4. After updating the file, restart the Serena common Tomcat service.

⚠ **CAUTION!** Do not change any other entries in the `sbm-client.properties` file. The other entries are used by Serena Release Manager Web services and any changes could cause Serena Release Manager to stop working correctly.

Chapter 9

Troubleshooting

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

Troubleshooting Overview

When you encounter an issue in Serena Release Manager, 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 displayed from Serena Release Manager Configurator.
- Information stored in the Serena Release Manager product log file, `rlm.log`.
- Troubleshooting information for integrating products, such as SBM, Dimensions CM, ChangeMan ZMF, and Serena Release Automation.

Related Topics

- Information from the Serena Release Control User Interface
- "Information from the Release Manager Configurator" on page 159
- "Information from Log Files" on page 160
- "Symptoms and Solutions" on page 161

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.

Related Topics

- Error Messages
- "Activity Log" on page 159
- "Activity Page" on page 159
- "History" on page 159
- "Information from the Release Manager Configurator" on page 159
- "Information from Log Files" on page 160
- "Symptoms and Solutions" on page 161

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 161 for more assistance in these situations.

Activity Log

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.

Activity Page

You can monitor release package deployment progress in real-time using the **Activity** page. This page lists all active release packages that have been deployed within a designated time period and gives visual indicators of in progress, complete, or failed status.

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.

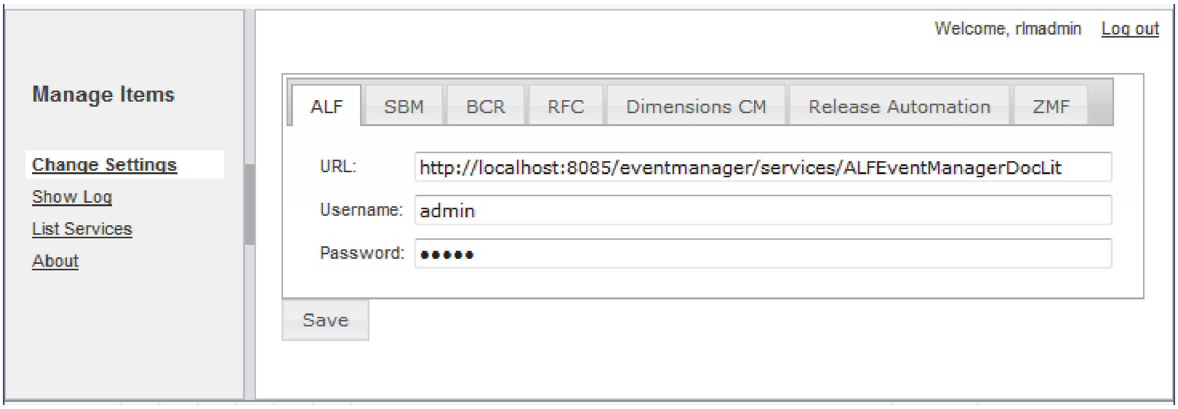
Information from the Release Manager Configurator

You can use the Serena Release Manager Configurator to help determine and resolve connection issues.

From the Serena Release Manager Configurator, click the following links in the navigation pane to display the corresponding pages.

- **Change Settings**: Verify and change connection information as needed.
- **Show Log**: Show and analyze the product log file.
- **List Services**: List the Web services and verify that they are running.
- **About**: View version and build information for Serena Release Control.

The Serena Release Manager Configurator is shown in the following figure.



URL: http://localhost:8085/eventmanager/services/ALFEventManagerDocLit

Username: admin

Password: ●●●●●

Related Topics

- Chapter 4, "Configuring Connections using the Release Manager Configurator" on page 49
- Information from the Serena Release Control User Interface
- "Information from Log Files" on page 160
- "Symptoms and Solutions" on page 161

Information from Log Files

Log files are created by the installer and by the Serena Release Manager product. Both of these files are useful for troubleshooting.

Related Topics

- Product Log File
- Installer Log Files
- Information from the Serena Release Control User Interface
- "Information from the Release Manager Configurator" on page 159
- "Symptoms and Solutions" on page 161

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.

You can view the log contents using the Serena Release Manager Configurator or using any text editor.

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 file:

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
```

Installer Log Files

The installer creates log files that contain the full trace log from the installation process, successful or otherwise.

The default location of the installer log files is the directory set by the variable %TEMP%. You can change directory into this folder. For example:

```
>cd %TEMP%
```

The installer log files are as follows:

`Install_rc_comp.log`

This file appears if the check box **Show the Windows log file** is selected on the final installation dialog box, before you click **Finish**.

`Summary.rtf`

This file shows the content of the **Installation Summary** page of the installation wizard. This is shown at the end of a successful installation, and shows details of the files installed onto the system during that installation run.

You can view the log file contents using any text editor. If there is an issue with the installation you should supply these files to Serena Support.

Symptoms and Solutions

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

- "Installer Errors" on page 162
- "Snapshot Promotion Errors" on page 162
- "Cannot log into Serena Release Control" on page 163
- "User Interface Display Issues" on page 164
- "Create Release Fails with a Check Uniqueness Error" on page 165
- "Matches Not Found for Selections" on page 166

- "No Change Package Data Displayed in ZMF Deployment Tasks" on page 167
- "Release Package Deployment Fails" on page 167
- "Slow Response Time" on page 168

Related Topics

- Information from the Serena Release Control User Interface
- "Information from the Release Manager Configurator" on page 159
- "Information from Log Files" on page 160

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 folder as the `Release Control.exe` file and that the folder has the Common Web services files.

Related Topics

- "Installer Log Files" on page 161

Snapshot Promotion Errors

If the promote of a snapshot fails or gives warning messages, here are some possible reasons and solutions.

The SBM environment endpoints are not mapped properly

If the promotion of a snapshot fails, the message in the log may give an error message similar to one of the following:

```
ERROR -- Can't deploy a process app with orchestration to an environment that doesn't
have target servers defined.

ERROR -- Cannot deploy BPEL definition for process model alf/13db576c-5bec-4115-8ea1-
56b44d7f0ffb/ - <soapenv:Reason xmlns:soapenv="http://www.w3.org/2003/05/soap-
envelope"><soapenv:Text xml:lang="en-US">The endpoint reference (EPR) for the
Operation not found is /eventmanager/services/ALFAdmin?ns=00000 and the WSA Action =
null</soapenv:Text></soapenv:Reason>.

Failed to complete the promotion to server "SBM Event Manager." at 1/18/12 10:19 AM.
```

Look at the SBM environment target servers and verify that they are set up properly.

Some common oversights are:

- Selecting the same server type twice, such as System Event Manager or BPEL Server (JBPM).
- Pointing both server types to the same endpoint URL.

The promotion succeeds but gives warnings

If the promotion of a snapshot succeeds but gives warning messages that concern you, you may or may not need to take further action.

Warnings you can ignore

Warning messages similar to one of the following can be ignored:

```
WARN -- Import: Invalid Data 'b320b63f-f08b-44ed-8803-806acc663278' in node
'LastModifierID'.

WARN -- Import: Invalid Data 'b320b63f-f08b-44ed-8803-806acc663278' in node
'AuthorID'.

WARN -- Import: Invalid Data '19c59fdf-f13d-4b1a-a07c-0ffca1c14a37' in node 'UserID'.
```

These warnings occur because the snapshot was captured on a system that had data populated and this data doesn’t exist on the target system. This is not a problem, as you will create your own data in the target system.

Warnings that require further action

Warning messages similar to one of the following should not be ignored:

```
WARN -- Unable to map a parameter for web service call 'Mashup Event.EventNotice' made
from transition 'Done' because field 'Application' in table 'Deployment' could not be
created.

WARN -- Unable to map a parameter for web service call 'Mashup Event.EventNotice' made
from transition 'Fail' because field 'Release Package' in table 'Deployment' could not
be created.
```

These warnings occur because there are dependencies between snapshots. When there is a two-way dependency, one of the dependent snapshots must be promoted again after the snapshot it depends on is promoted.

The easiest way to resolve the mappings is to promote all snapshots once and then promote any snapshots that had the mapping warnings again.

Related Topics

- Chapter 4, "Creating an Environment for Serena Release Manager" on page 37
- Chapter 4, "Promoting the Snapshots" on page 40

Cannot log into Serena Release Control

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.

Verifying the Web Services Connection

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

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

   ```
   http://localhost:9095
   ```

   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. After 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:9095/rlm
   ```

3. Log into the Serena Release Manager Configurator.

4. In the navigation pane, click **List Services**.

   Verify that the Serena Release Manager Web services are listed and in active status as follows:

   - RLMUtilService
   - ReleaseRequestService
   - DeploymentAutomationService
   - DeployUnitService

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

User Interface Display Issues

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

Related Topics

- You can’t view something you created or added
- The UI shows outdated elements
- Some of the UI elements are missing
- Release trains are not appearing on the Calendar page

You can’t view something you created or added

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.

Examples:

- If you added release trains and they do not appear in the inbox in the **Manage All Items** section, your privileges probably aren’t set properly.

- If you created a deployment process template and do not see the option to add deployment tasks, the owning role may not be enabled for the RLM Aux project.
- If you created 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.

Some of the UI elements are missing

If your UI looks correct, but some elements, such as the spell check icon, do not appear, ensure that you have put the files in the SBM database during the installation, upgrade, or after customizing and redeploying the process apps.

The UI shows outdated elements

If your user interface retains elements from a prior release of RLM after an upgrade, new installation, or customization, you may need to clear and refresh your browser cache. This occurs because certain UI elements are stored in the browser cache for faster refresh time.

Release trains are not appearing on the Calendar page

If your release trains do not appear on the calendar, make sure that the following are true:

- The release trains have been created and you are logged in with a user who has the privileges to view them. For example, can you view the release train from your inbox?
- The release trains have to and from schedule dates for the last stage, such as Production. If not, you can view the release train from your inbox and edit it to add the schedule.
- You have completed the suggestions in the preceding troubleshooting sections.

Related Topics

- Chapter 4, "Configuring Required Objects in Serena Release Control" on page 46
- Chapter 4, "Installing the UI Shell files" on page 35
- Chapter 9, "Information from Log Files" on page 160

Create Release Fails with a Check Uniqueness Error

When you try to create a release train, application release, or release package, the operation may fail with an error similar to the following:

```
Error occurred during web service invocation:

SOAP Fault Code: env:Client

SOAP Fault String: checkUniquness: Blank: The error occurred during the execution of the
orchestration workflow.
```

If you receive this error even though the name and version are unique, this means the check uniqueness call is failing. This is the first Web services call in the orchestration workflow, so this message could simply mean that there is a problem with the Web services.

The Web services are using the wrong port number

The check uniqueness error may indicate that the Serena Release Manager Web services are defined for a different port number than the one on which the Serena Common Tomcat services are currently running. If you are using a non-default port number (not 9095), this message may indicated that you have not changed the port number in all of the Web services WSDLs.

An incorrect version of the Web services is present

Make sure that the correct version of the `rlm.war` file is installed under the Serena common Web services and that no older version of the `rlm.war` file is present in the ..`\Serena\common\tomcat\6.0\webapps` folder. You can use the RLM Configurator to see if the Web services are running.

Related Topics

- Chapter 8, "Configuring Release Manager to Use a Different Port" on page 143
- "Information from the Release Manager Configurator" on page 159

Matches Not Found for Selections

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

We did not find any matches for your request (error 401 or 404)

If Serena Release Manager fails to find any matches for your request, for example when you select a project for a release package or click **Add development change requests**, this indicates that the connection to the provider is not completing successfully. There are several things that can cause this.

User credentials must match in both products

First check the `rlm.log` file for specific error messages.

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.

SSO must be enabled in both products and share an SSO server

Verify that both SBM and Dimensions CM have SSO enabled and share the same SSO server.

SBM 10.1.1.1: SSO must be selected for each RESTgrid widget

If you are using SBM 10.1.1.1, the security token may not be included in the URL the RESTgrid widget uses to pass to the SSO server. You must select the **Use SSO authentication** check box for each RESTgrid widget to enable the security token to be included in the URL. See Chapter 4, "SBM 10.1.1.1 Only: Configuring SSO in RESTgrid Widgets" on page 44 for more details.

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

If Serena Release Manager fails to find any matches for your request when you try to select a process for an automation deployment task, verify the following:

- The desired application, environment, process, and server combination is defined in Serena Release Automation.
- The values used for the connection to Serena Release Automation are correct.

Related Topics

- "Product Log File" on page 160
- Chapter 4, "SBM 10.1.1.1 Only: Configuring SSO in RESTgrid Widgets" on page 44
- Chapter 4, "System Activation and Configuration" on page 33

No Change Package Data Displayed in ZMF Deployment Tasks

If there is no change package data displayed in the ZMF deployment task, look at the message at the bottom left of the page. If it says "Waiting for localhost", it's still loading.

If this is not the problem, recheck your configuration and make sure the HTTP server is running on the port you specified in the HTTP Server setting in the Serena Release Manager Configurator **ZMF** tab.

Related Topics

- Chapter 4, "Configuring Connections using the Release Manager Configurator" on page 49
- Chapter 4, "Configuring the ChangeMan ZMF Web Page Widgets" on page 42
- Chapter 4, "Configuring ZMF Communication on the Mainframe" on page 60
- Appendix B, "ZMF: SERNET HTTP Server Setup" on page 189

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 Page** and **Activity Log** tab for information. If those do not give enough information for you to identify the problem, check the details in the `rlm.log` file.

Verify that the connection requirements are met

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.

Check for an SbmDeployUnitHelper error message

If you receive an error in the product log file similar to the following:

```
Error

java.lang.NullPointerException: null at java.util.StringTokenizer.(Unknown Source)
~[na:1.6.0_12] at java.util.StringTokenizer.(Unknown Source) ~[na:1.6.0_12]
at
com.serena.rlm.sbm.client.internal.util.hlp.SbmDeployUnitHelper.getRelatedByIds(Sb
mDeployUnitHelper.java:175)
```

check the `sbm-client.properties` related deployment unit ID table value to ensure it is entered exactly as follows.

```
deployment.extfield.relatedDeployUnitId=RELATED_DEPLOY_UNITS
```

The value should be plural, not singular. If you have upgraded from a pre-release version of Serena Release Manager 2.1 and did not replace the `sbm-client.properties` file with the latest version, you could encounter this issue.

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 Serena Release Automation.

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

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

Related Topics

- "Troubleshooting Overview" on page 158
- Chapter 4, "System Activation and Configuration" on page 33
- Chapter 5, "Configuration and Administration of the Integrating Objects" on page 73
- Chapter 6, "Provider Configuration" on page 93

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 Manager, all SBM server components, all Dimensions CM 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, Dimensions CM, 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.

Related Topics

- Chapter 3, "Serena Release Manager Installation" on page 25

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 Manager 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 objects and what actions are pending.

This section gives a reference of workflows provided in the default version of Serena Release Manager. The workflows implemented for your organization may be customized so that they differ from these. The most reliable way to analyze your workflows is to open them in SBM Composer.

Workflow Relationships

Serena Release Manager is a hierarchical system, with Release Train as the highest level organizing entity, Application Release next, and so on. Release Packages can optionally participate in the hierarchy or exist standalone. Deployment Tasks are dependent on Release Packages.

Workflow Dependencies

The two main dependency relationships among the workflows are restrictions and automations.

Workflow Restrictions

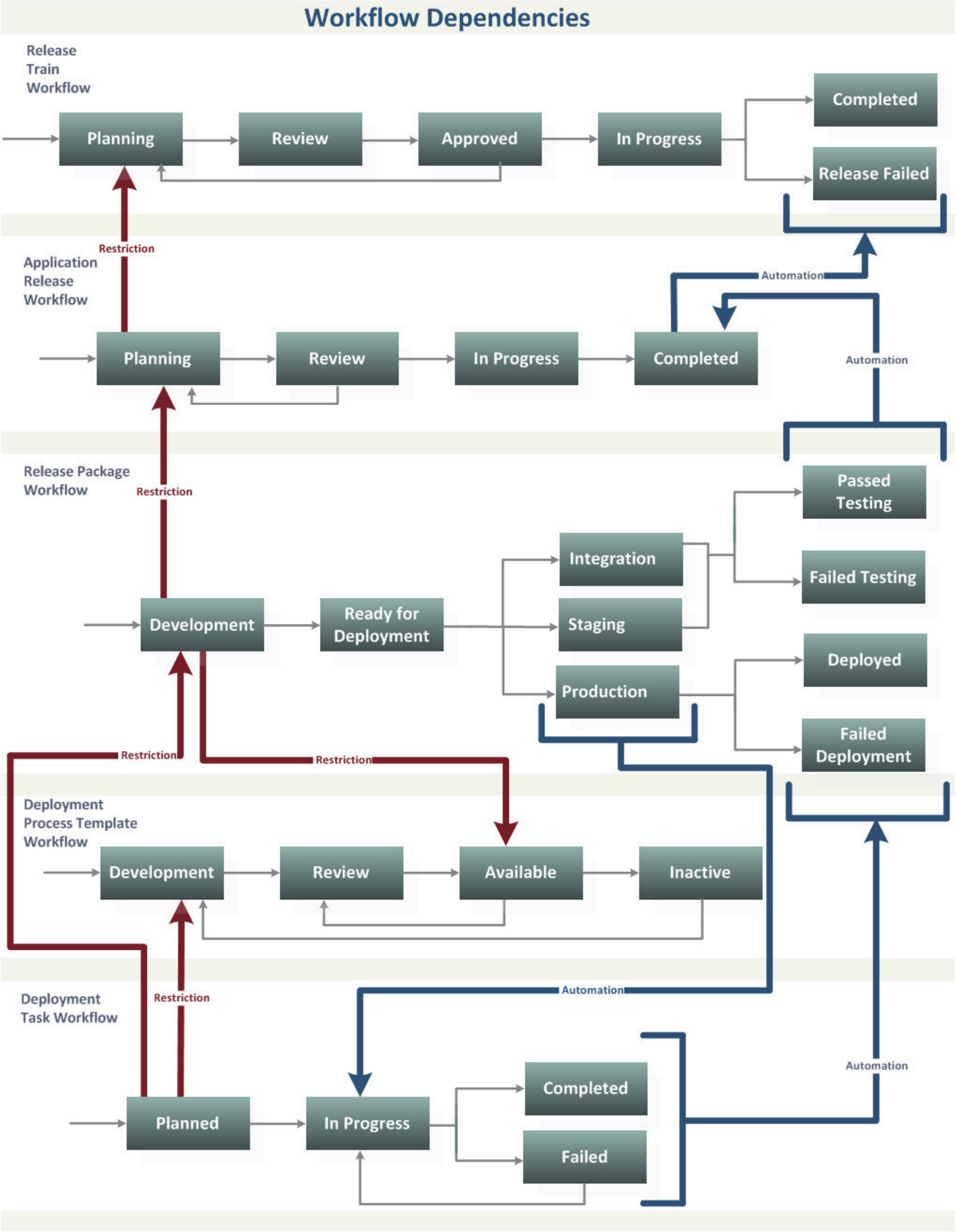
Most of the default workflows allow editing and associating objects to other objects only when items are in the Development or Planning states. For example, you can associate application releases with a release train only when that release train is in the Planning state. You can associate release packages with an application release only when the application release is in the Planning state.

Workflows for Deployment Process Templates are standalone, although deployment process templates populate the deployment tasks for release packages and in that respect have a link to the workflows for both Deployment Tasks and Release Packages. Workflows for environments are completely standalone by default.

Workflow Automations

Many of the workflows are moved to completion states automatically when associated objects are complete. The automation is usually initiated at the lowest level of association and moves up the hierarchy as each level of release activities are completed. For example, after all deployment tasks for a release package reach completion, the release package is marked complete. After all release packages for an application release reach completion, the application release is marked complete. When all application releases for a release train reach completion, the release train is marked complete.

The dependencies among the default workflows are shown in the following figure.

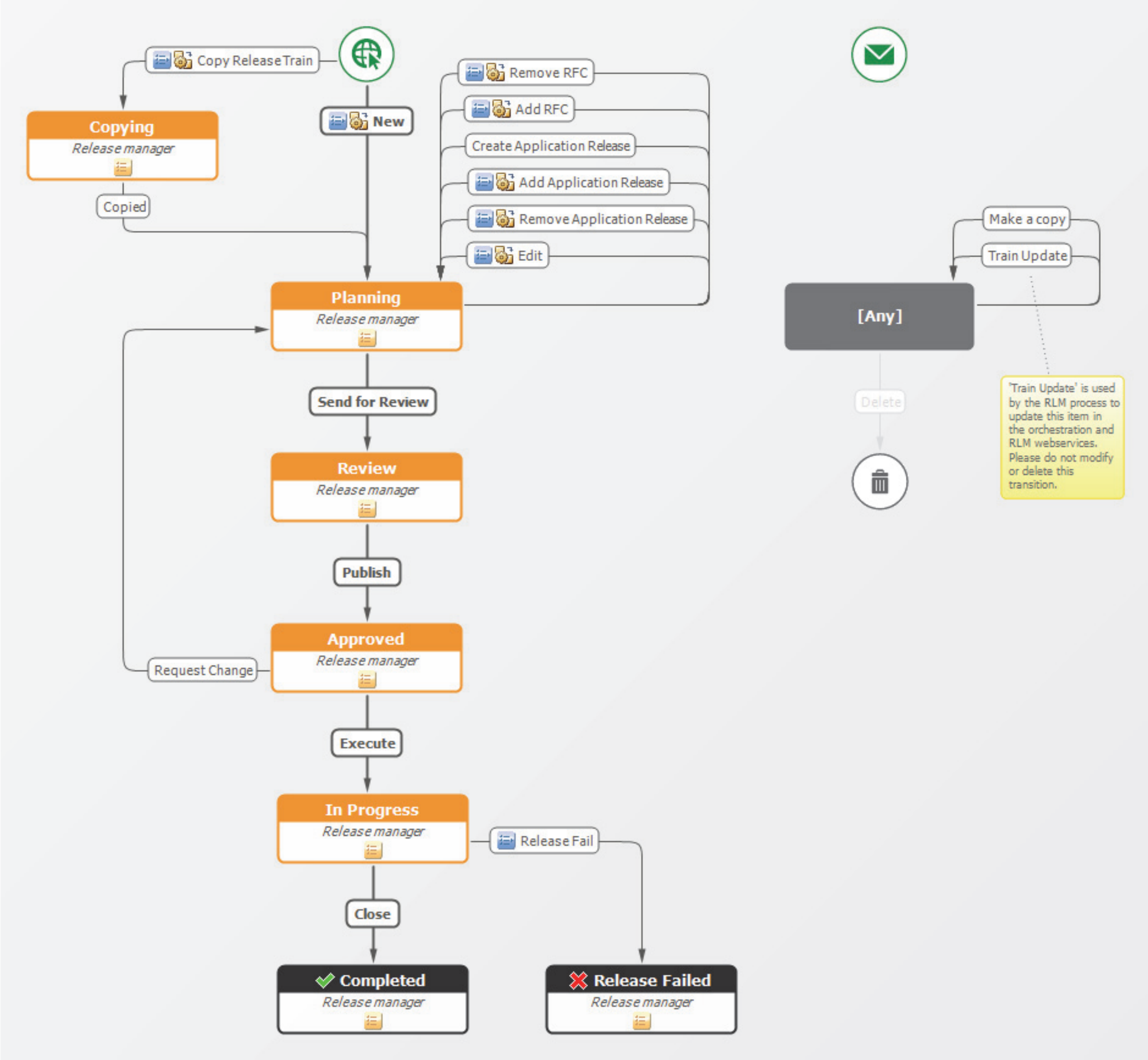


Related Topics

- Release Train Workflow
- "Application Release Workflow" on page 176
- "Release Package Workflow" on page 177
- "Deployment Task Workflows" on page 182
- "Deployment Process Template Workflow" on page 187
- "Environment Workflow" on page 188

Release Train Workflow

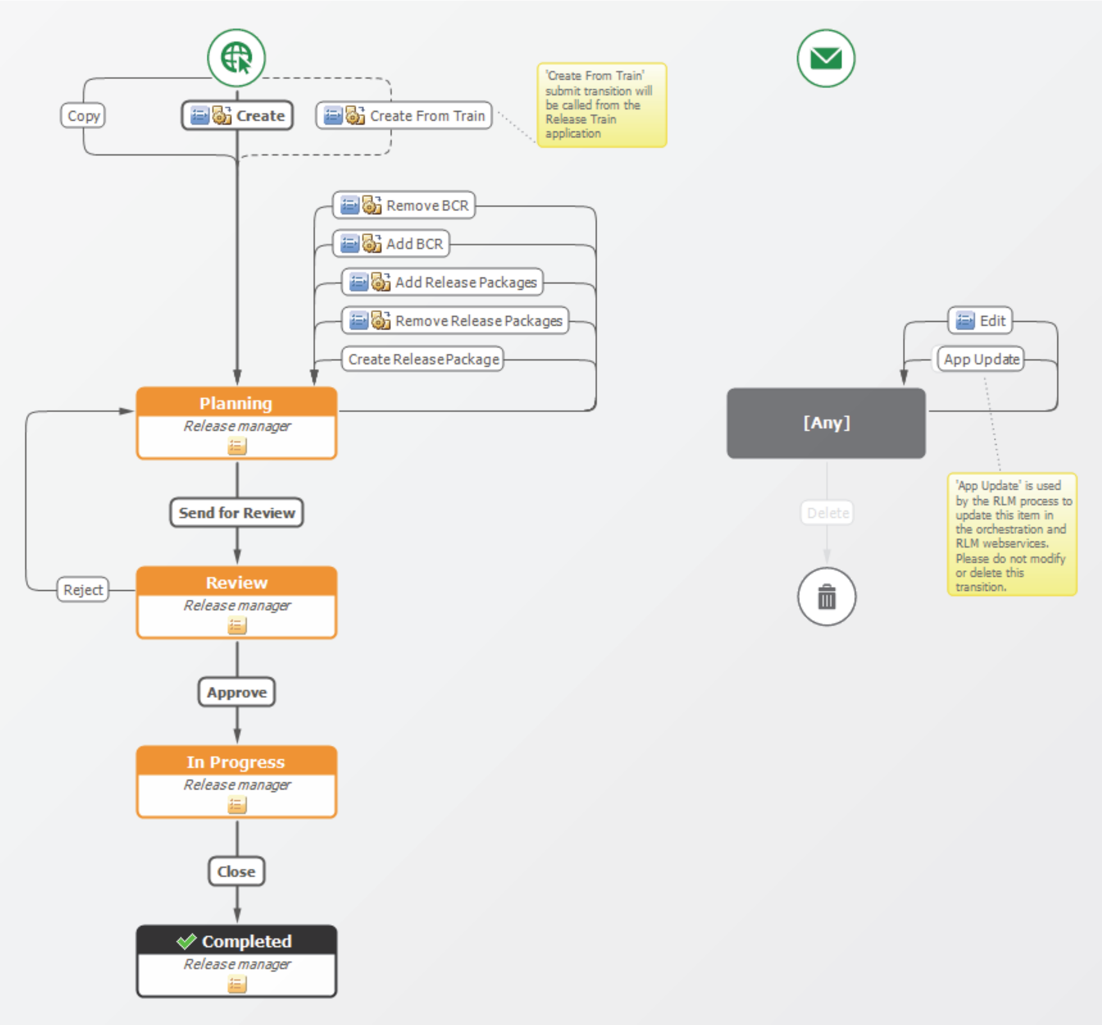
The Release Train workflow in the default version of Serena Release Manager is shown in the following figure.



'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 Manager 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.

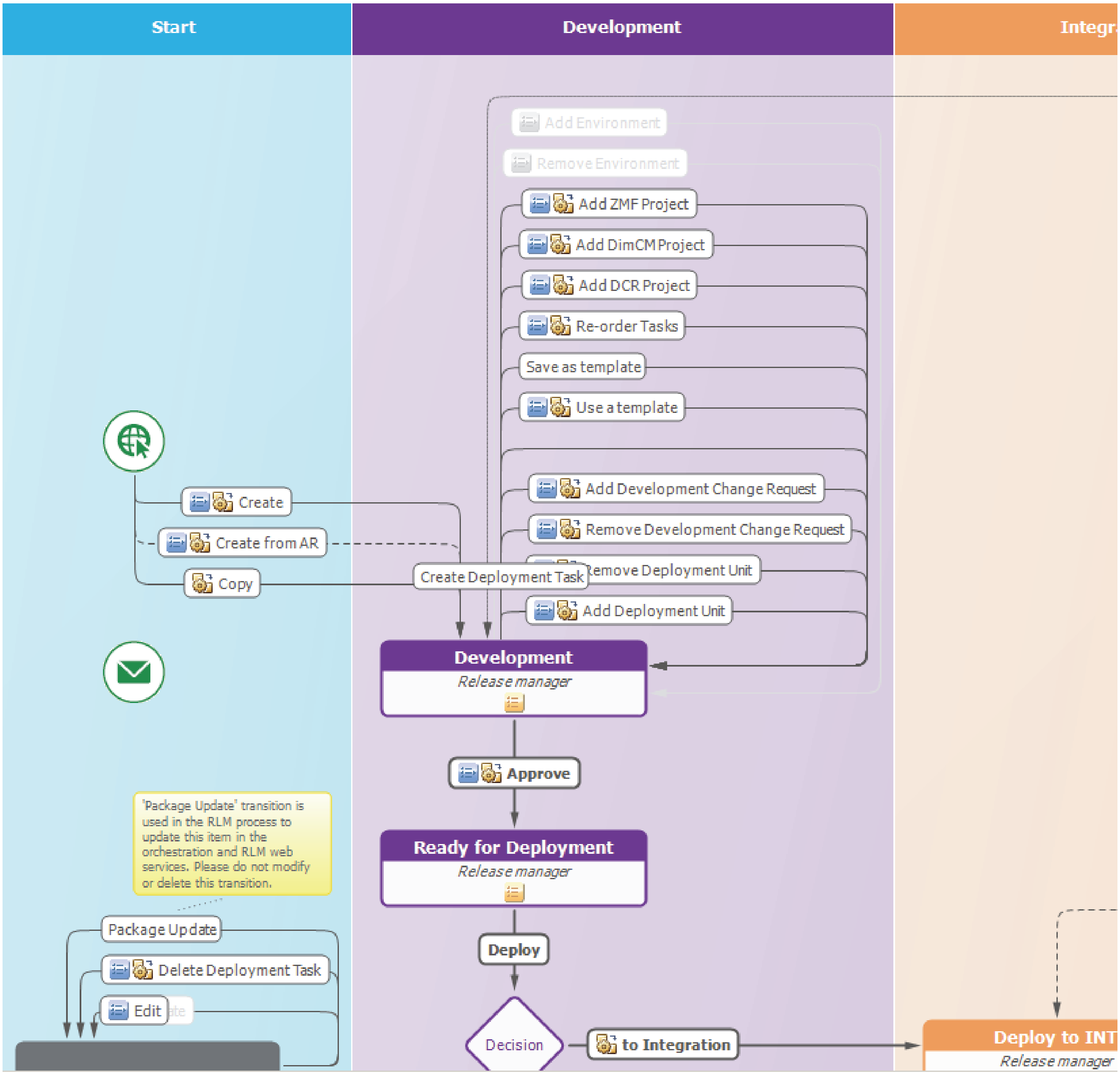
Release Package Workflow

The Release Package workflow in the default version of Serena Release Manager 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.

Related Topics

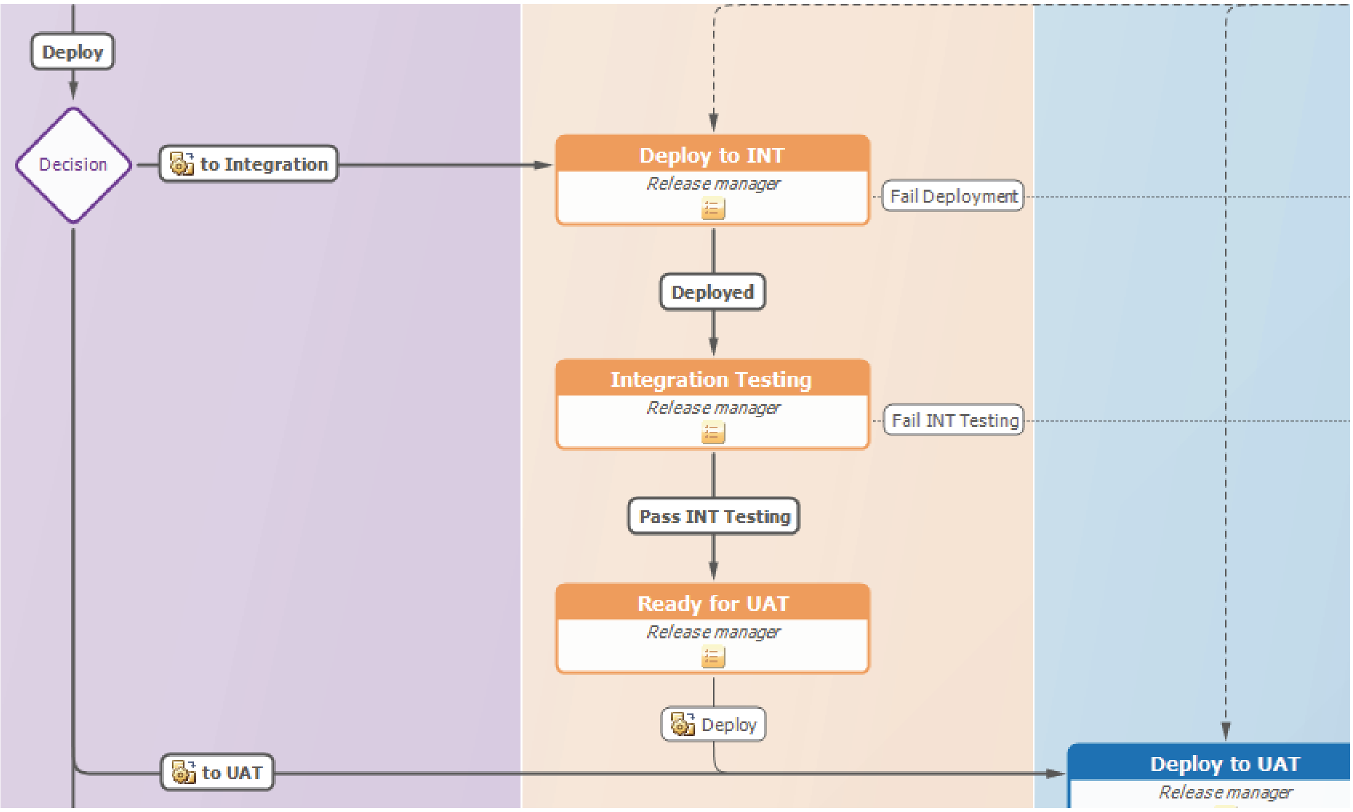
- Start and Development States
- "Integration State" on page 179
- "Staging and Production States" on page 180
- "Exceptions State" on page 181

Start and Development States

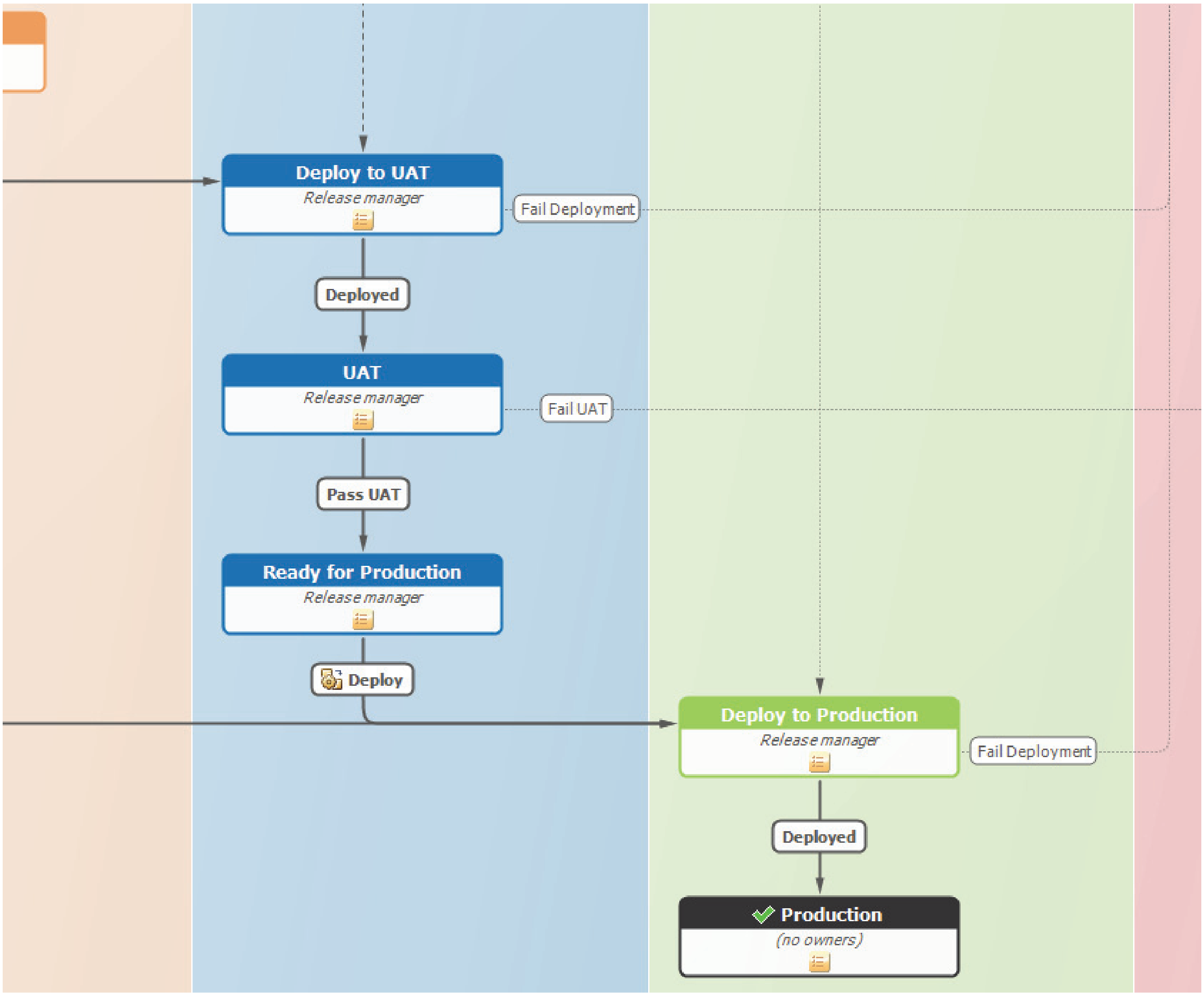


'Package Update' transition is used in the RLM process to update this item in the orchestration and RLM web services. Please do not modify or delete this transition.

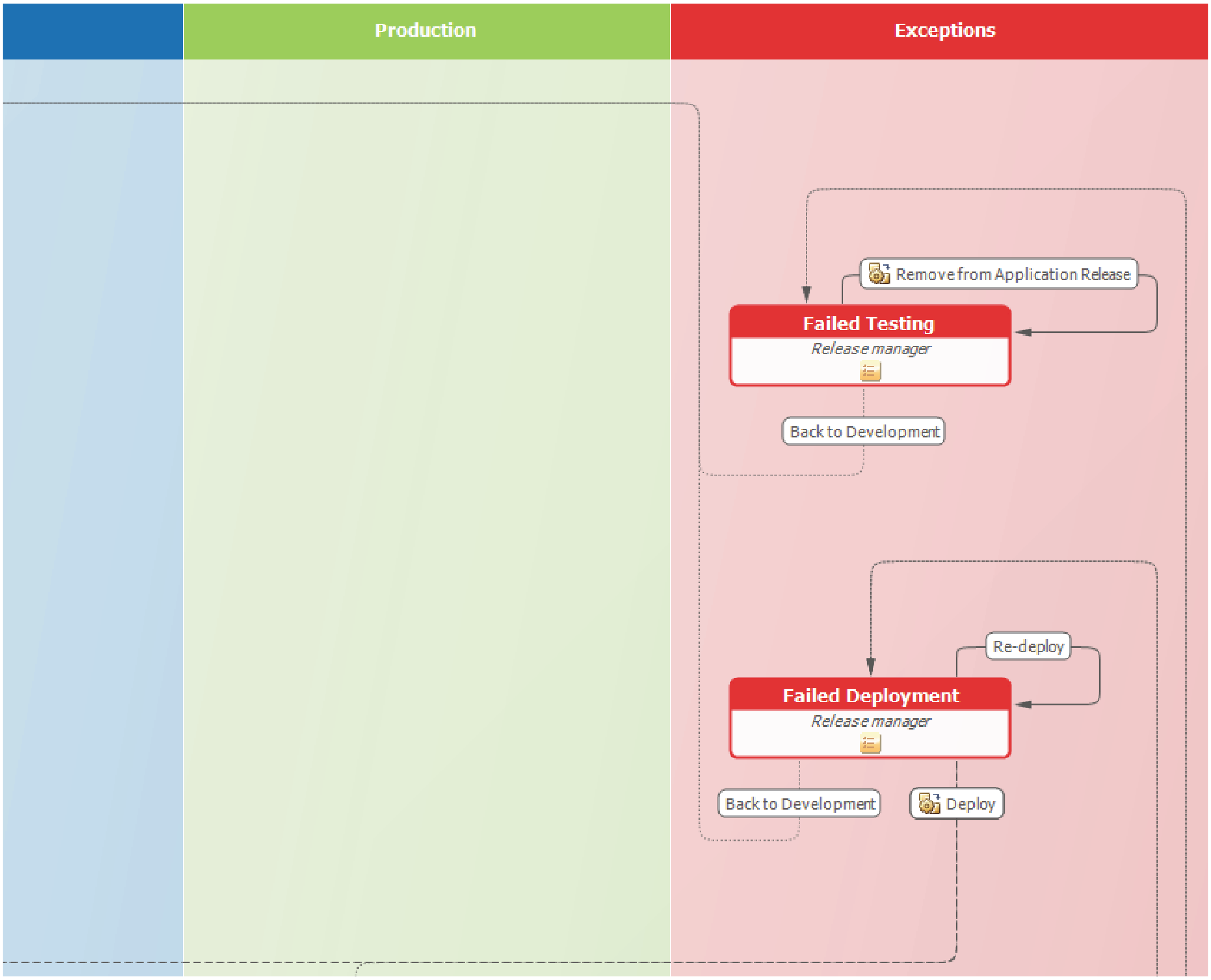
Integration State



Staging and Production States



Exceptions State



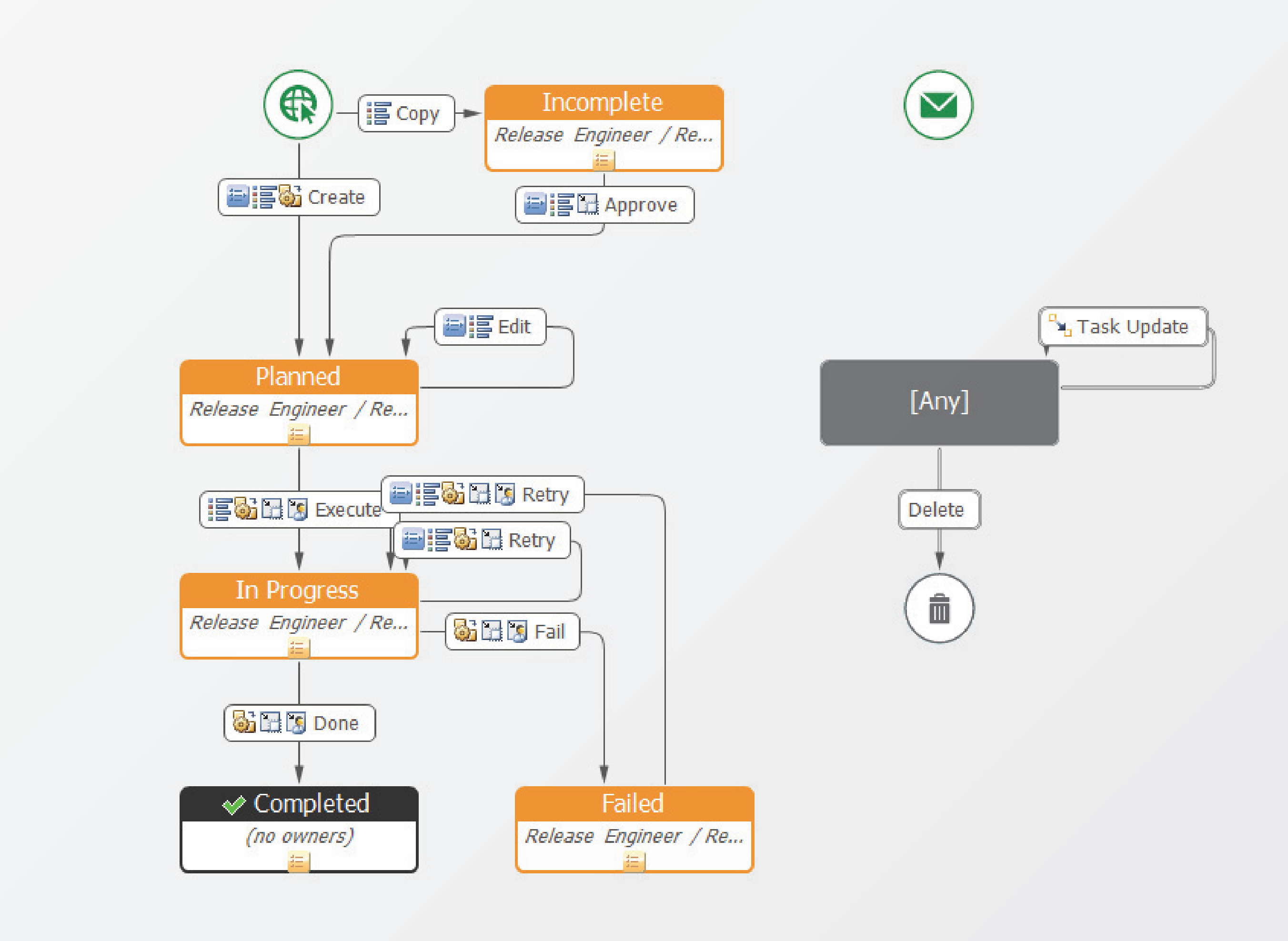
Deployment Task Workflows

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

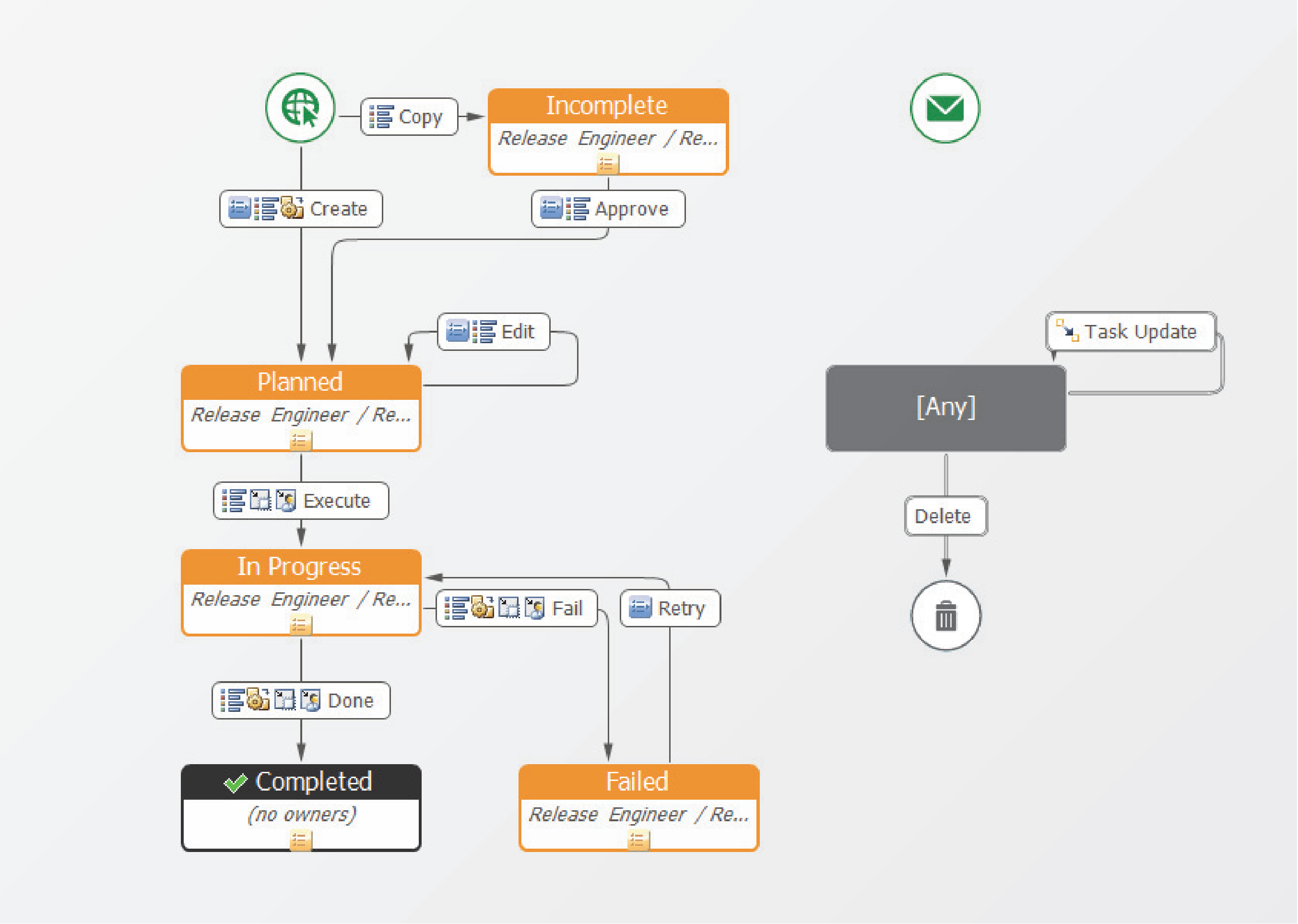
Related Topics

- Automation Deployment Task Workflow
- "Manual Deployment Task Workflow" on page 183
- "Vault Deployment Task Workflows" on page 184

Automation Deployment Task Workflow



Manual Deployment Task Workflow

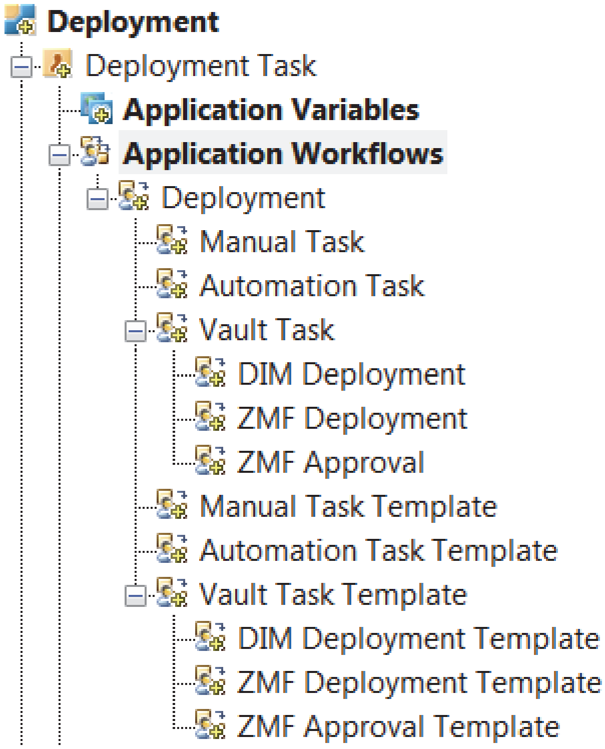


Vault Deployment Task Workflows

There are several vault deployment workflow types to support variations of vault deployment tasks.

Deployment Task Hierarchy

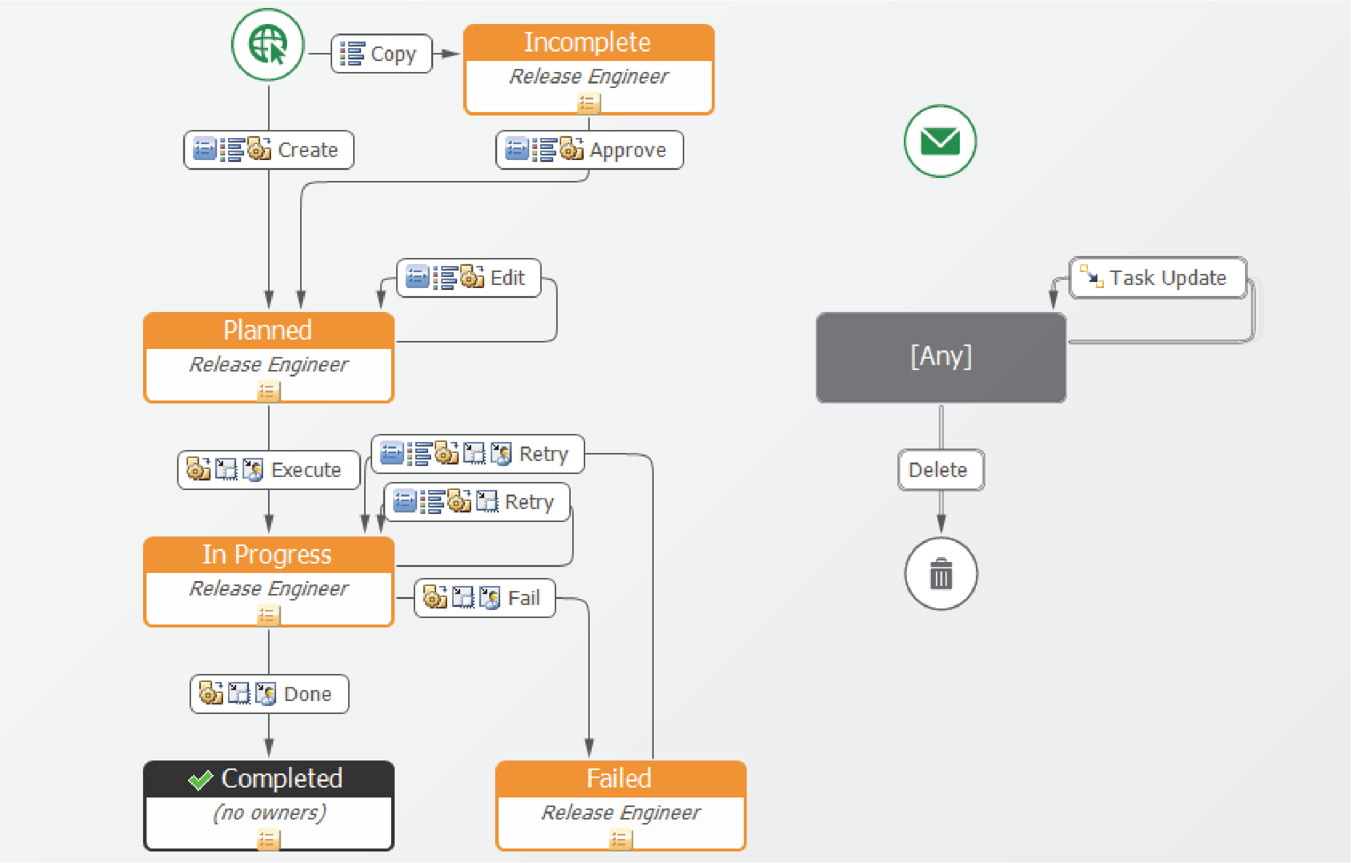
The default hierarchy of workflows and sub-workflows for vault deployment tasks is shown in the following figure.



- **Deployment**
  - Deployment Task
    - **Application Variables**
    - **Application Workflows**
      - Deployment
        - Manual Task
        - Automation Task
        - Vault Task
          - DIM Deployment
          - ZMF Deployment
          - ZMF Approval
        - Manual Task Template
        - Automation Task Template
        - Vault Task Template
          - DIM Deployment Template
          - ZMF Deployment Template
          - ZMF Approval Template

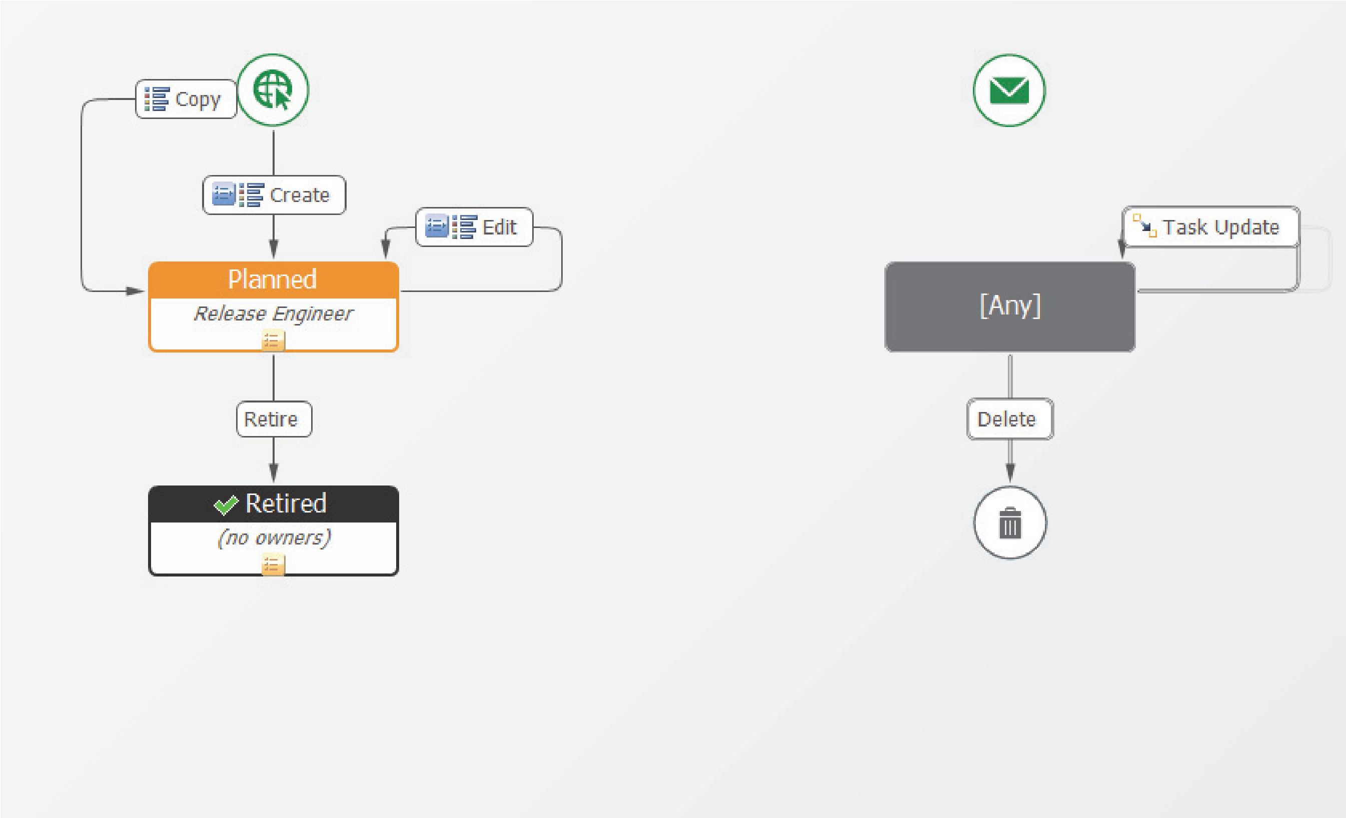
Main Vault Task Workflow

The DIM Deployment, ZMF Deployment, and ZMF Approval sub-workflows inherit from the main workflow for Vault Task. The main Vault Task workflow is shown in the following figure.



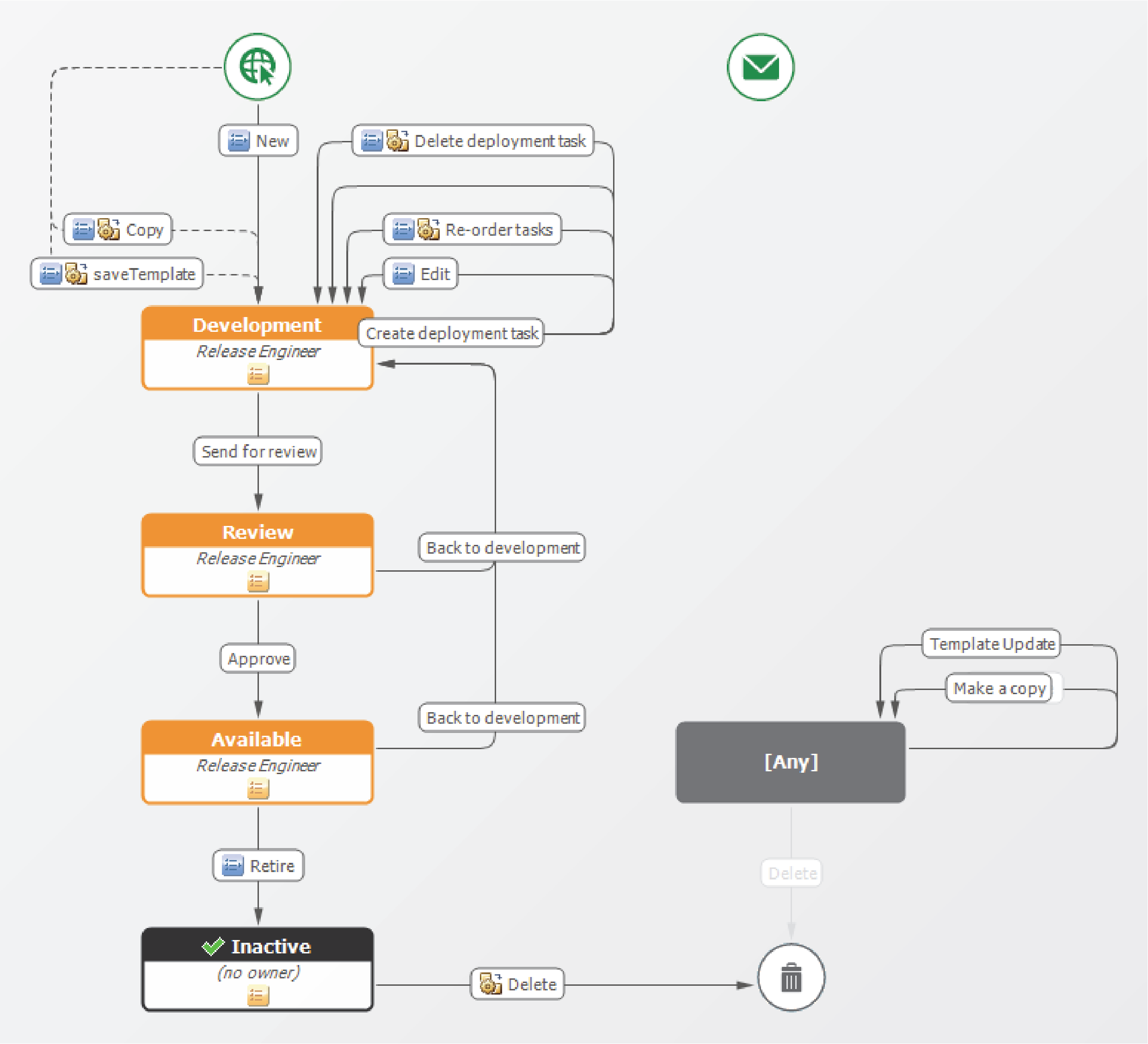
Main Vault Task Template Workflow

The DIM Deployment Template, ZMF Deployment Template, and ZMF Approval Template sub-workflows inherit from the main workflow for Vault Task Template. The main workflow for Vault Task Template is shown in the following figure.



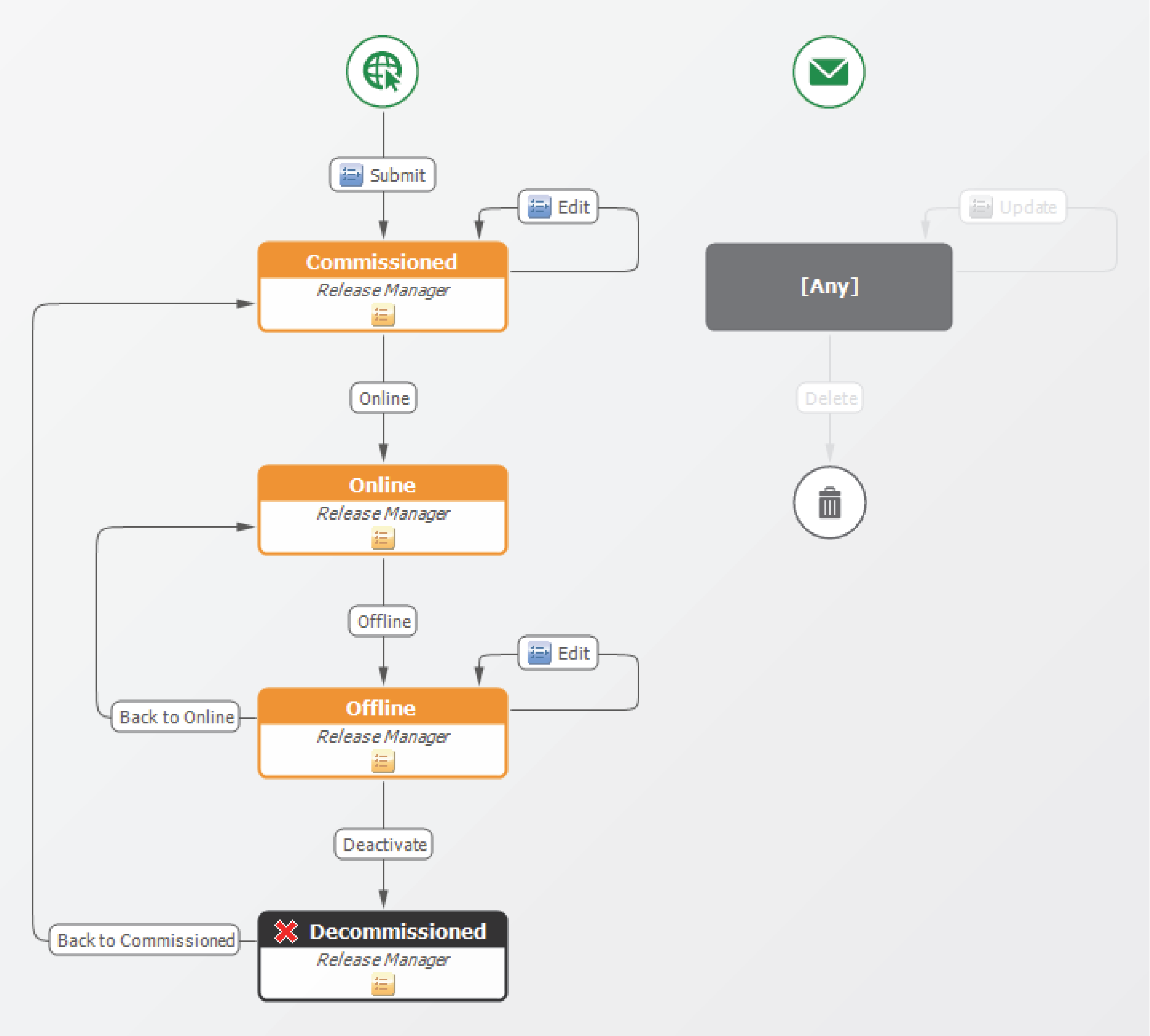
Deployment Process Template Workflow

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



Environment Workflow

The Environment workflow in the default version of Serena Release Manager is shown in the following figure.



Appendix B

ZMF: SERNET HTTP Server Setup

To support communications with Serena Release Manager, ChangeMan ZMF users require SERSERVC, a native mainframe HTTP server than runs under SERNET. SERSERVC is written in REXX and runs under IBM's Unix System Services (USS) on the host. It supports UTF-8 character encoding only.

This section discusses installation, operation, and runtime considerations for SERSERVC.

SERSERVC Prerequisites

SERSERVC has the following general requirements:

- A dedicated TCP/IP port ID must be assigned to the HTTP server.
- A version of ChangeMan ZMF that supports Serena Release Manager must be installed.

  Refer to the Serena product software compatibility matrix for information on versions of ChangeMan ZMF that integrate with Serena Release Manager.

NOTE SERSERVC works with versions of ChangeMan ZMF that may not be certified to work with Serena Release Manager and may be installed independently of Serena Release Manager.

SERNET User ID

To run the SERSERVC HTTP server under z/OS Unix System Services (USS), SERNET requires a RACF userid with the following features:

- *OMVS segment* Establish this segment to manage USS privileges in RACF.
- *User privileges* In the OMVS segment assigned to the SERNET user ID, set the Unix user ID number to a value that will give SERNET the highest user privilege level allowable in your shop. Superuser privileges (that is, `UID(0)`) are not required.
- *Unix home directory* Assign a home directory in the OMVS segment for the user ID. By convention, the home directory path name takes the form

  ```
  /u/userid
  ```

  where `userid` is the SERNET user ID you assign in RACF.

  In the examples below, we will use a SERNET user ID of `sernet` and a Unix home directory path name of `/u/sernet`.

PRIVILEGES Serena recommends that SERNET be assigned the same user ID as the ChangeMan ZMF started task class.

Verifying SERNET User ID Privileges

If you are already using TCP/IP for communications with ChangeMan ZMF, SERNET will already have a user ID with an OMVS segment assigned. For example, you may use SERNET to communicate with ChangeMan ZDD or use the ChangeMan ZMF Load Balancing Option (LBO). However, user privileges and/or a Unix home directory may not be established.

To verify privileges for an existing SERNET user ID:

1 Find the current SERNET user ID.

a Use SDSF to examine a running ChangeMan ZMF task or the JESMSGLG of the job output from a previous execution of ChangeMan ZMF.

b At the top of the message log, usually next to the message

```
$HASP373 STARTED
```

find message `IEF695I`. This message includes an `ASSIGNED` message line that identifies the SERNET user ID.

2 Retrieve the OMVS segment for the SERNET user ID.

a At the TSO command line, issue the following command:

```
LU userid OMVS
```

where `userid` is the SERNET user ID found above, such as `sernet`.

b If no OMVS segment is returned, ask your systems programmer or security administrator to add one. (See "SERNET User ID" for required privileges.)

c If an OMVS segment exists for the user ID, verify that the `UID` is set to zero and note the Unix home directory (for example, `/u/sernet`). If either are missing, ask your security administrator to add it to the OMVS segment.

PRIVILEGES

- UID(0) is not required for the HTTP server.
- UID(0) is recommended but not required for the SERNET server. The SERNET server already runs as APF-authorized. The server must have read, execute, and write privileges to the HFS file system, and in particular, anything that starts with the home directory of the server's user ID.

Installing SERSERVC

To install `SERSERVC` perform the following steps.

SERSERVC Runtime Directory

1 Create the `SERSERVC` runtime directory with a path name of the form

```
/u/userid/serservc
```

where `userid` is your actual SERNET user ID. (In our examples, this is `sernet`.)

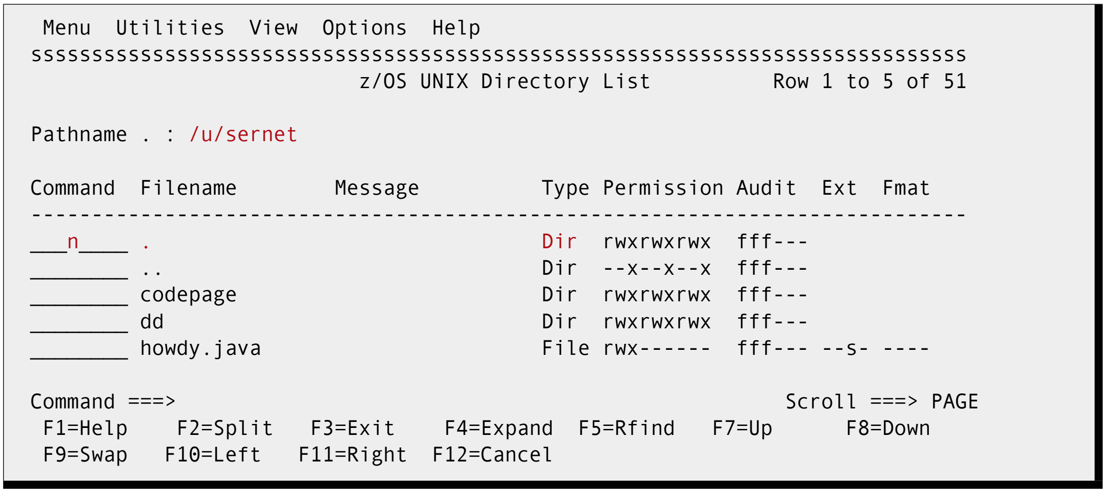
a Select ISPF option `3.17` to invoke the UNIX directory list utility.

b In the **Pathname** field of the **z/OS Unix Directory List Utility** panel, type the SERNET home directory path (for example, `/u/sernet`). Leave the **Option==>** prompt blank to request a display of directory contents and press Enter.

The directory list for the SERNET home directory displays.

c In the **Unix Directory List** panel for the SERNET home directory, type the **N (New)** line command at the root level of the file hierarchy. The **Filename** for this level is listed as a single period (`.`) and the **Type** is "Directory" (`Dir`).

For example:



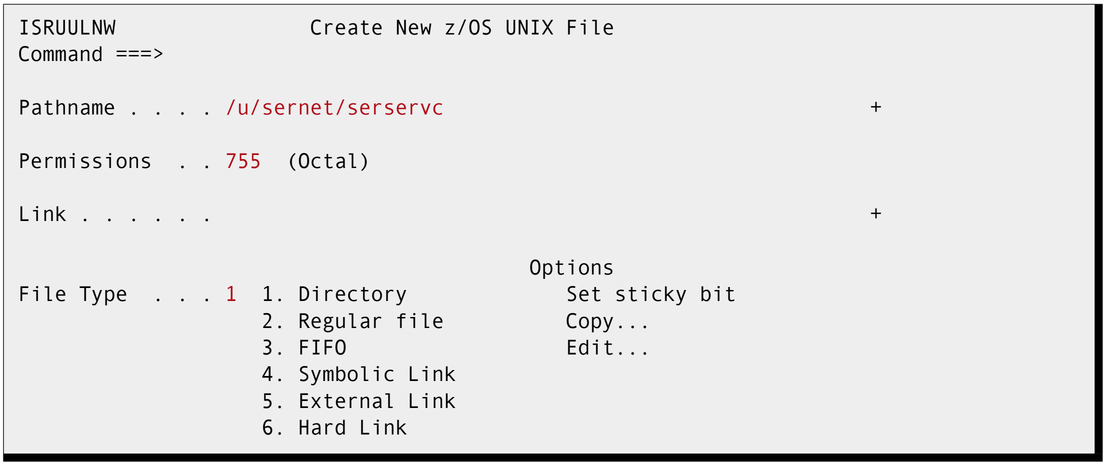
```
 Menu  Utilities  View  Options  Help
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                           z/OS UNIX Directory List          Row 1 to 5 of 51

Pathname . : /u/sernet

Command  Filename        Message          Type Permission Audit  Ext  Fmat
-----------------------------------------------------------------------------
___n____ .                                Dir  rwxrwxrwx  fff---
________ ..                               Dir  --x--x--x  fff---
________ codepage                         Dir  rwxrwxrwx  fff---
________ dd                               Dir  rwxrwxrwx  fff---
________ howdy.java                       File rwx------  fff--- --s- ----

Command ===>                                                  Scroll ===> PAGE
 F1=Help     F2=Split    F3=Exit     F4=Expand  F5=Rfind   F7=Up      F8=Down
 F9=Swap    F10=Left    F11=Right   F12=Cancel
```

Press Enter.

d When the **Create New z/OS UNIX File** window displays, create a new directory called `serservc` with permission level 755. Required field values are:

- **Pathname** Give the full path name for the new directory `serservc`. For example: `/u/sernet/serservc`
- **Permissions** Type `755` to assign read, write, and execute permissions over the directory to the directory owner, the owner's security group, and all others.
- **File Type** Type `1` to identify the new data object as a directory.

For example:



```
ISRUULNW                Create New z/OS UNIX File
Command ===>

Pathname . . . . /u/sernet/serservc                                   +

Permissions  . . 755  (Octal)

Link . . . . . .                                                      +

                                          Options
File Type  . . . 1  1. Directory             Set sticky bit
                    2. Regular file          Copy...
                    3. FIFO                  Edit...
                    4. Symbolic Link
                    5. External Link
                    6. Hard Link
```

Press Enter and then exit the utility with PF3.

NOTE The creation of the UNIX directory `/u/userid/serservc` must be done by a user with the proper authority, such as a systems programmer. Some user IDs may not have access to ISPF 3.17.

SERSERVC Runtime JCL

2 Copy the sample runtime JCL module SERSERVC to your actual installation PROCLIB. Member SERSERVC resides in the CNTL library where you unloaded the ChangeMan ZMF SERCOMC installation libraries.

3 Customize the runtime JCL for SERSERVC.

The following model SERSERVC JCL segment is supplied for your reference when making these changes. The actual downloaded JCL may vary from this example.

```
//SERSERVC PROC OUTC=H,                                  * CLASS
//              PORT=6657,                               * PORT
//              PATH='/u/sernet/serservc',               * PATH
//              PROCLIB='USER.PROCLIB'                   * PROCLIB
//*=================================================================*
//*         JCL TO EXECUTE SERSERVC SDSF SERVER                     *
//*=================================================================*
//SERSERVC EXEC PGM=BPXBATCH,
//          PARM='sh &PATH./serserv &PORT'
//*
//SYSEXEC  DD  PATH='&PATH/'
//SYSPRINT DD  SYSOUT=&OUTC
//SYSTSPRT DD  SYSOUT=&OUTC
//STDOUT   DD  PATH='&PATH./stdout',
//             PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//             PATHMODE=SIRWXU
//STDERR   DD  PATH='&PATH./stderr',
//             PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//             PATHMODE=SIRWXU
//MSGLOG   DD  PATH='&PATH./msglog'
//STDENV   DD  DUMMY
//*=================================================================*
```

a For the PORT parameter, change the sample port number to the actual IP port assigned for the exclusive use of the SERSERVC HTTP server.

b In the PATH parameter, replace the sample home directory, /u/sernet, with the actual Unix home directory you created for SERNET.

CAUTION! The home directory is the top-level directory for SERNET. Do not change the name of the serservc subdirectory in this path.

c For the PROCLIB parameter, replace the sample value USER.PROCLIB with the name of your actual installation PROCLIB.

SERSERVI Install Job

4 Customize the JCL for the SERSERVI install job. Member SERSERVI resides in the CNTL library where you unloaded the ChangeMan ZMF SERCOMC installation libraries.

The following sample SERSERVI JCL segment is supplied for your reference when making these changes. The actual downloaded JCL may vary from this example.

```
//jobcard   JOB ,'USS JOB',CLASS=A,
//          NOTIFY=userid
//*
//STEP1    EXEC PGM=IKJEFT01,DYNAMNBR=200,COND=EVEN
//SYSTSPRT DD  SYSOUT=*
//HFSOUT   DD  PATH='/u/sernet/serservc/stdout',
//             PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//             PATHMODE=(SIRWXU,SIRWXG,SIRWXO)
```

```
//HFSERR   DD  PATH='/u/sernet/serservc/stderr',
//             PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//             PATHMODE=(SIRWXU,SIRWXG,SIRWXO)
//EMPTY    DD  *
/*
//NEWLOG   DD  PATH='/u/sernet/serservc/msglog',
//             PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//             PATHMODE=(SIRWXU,SIRWXG,SIRWXO)
//NEWHDR   DD  PATH='/u/sernet/serservc/headers',
//             PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//             PATHMODE=(SIRWXU,SIRWXG,SIRWXO)
//MSGARCH  DD  *
******** ******** SerServ *------------------ Archive Restart ---*
/*
//NEWSRV   DD  PATH='/u/sernet/serservc/serserv',
//             PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//             PATHMODE=(SIRWXU,SIRWXG,SIRWXO)
//NEWTSK   DD  PATH='/u/sernet/serservc/sertask',
//             PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//             PATHMODE=(SIRWXU,SIRWXG,SIRWXO)
//SERSERV  DD  DISP=SHR,DSN=CMNPRD.CMN.SE56.#000031.REX(SERSERV)
//SERTASK  DD  DISP=SHR,DSN=CMNPRD.CMN.SE56.#000031.REX(SERTASK)
//SERCMD   DD  DISP=SHR,DSN=CMNPRD.CMN.SE56.#000031.CLS(SERCMD)
//NEWCMD   DD  DISP=SHR,DSN=USER.SYS1.CLIST(SERCMD)
//SERVA    DD  DISP=SHR,DSN=USER.PROCLIB(SERSERVA)
//SERVP    DD  DISP=SHR,DSN=USER.PROCLIB(SERSERVP)
//SYSPRINT DD  SYSOUT=*
. . .
```

a Copy the SERSERVI JCL sample to a work library for editing.

b Edit the job card as needed.

c In each occurrence of the PATH parameter that is supplied to various job steps in this job, change the sample home directory name, /u/sernet, to the actual name of the Unix home directory you defined for SERNET.

> **CAUTION!** The home directory is the top-level directory for SERNET. Do not change the name of the serservc subdirectory or any lower-level directories or files in the PATH parameter value.

d For jobs SERSERV and SERTASK, change the dataset names in the sample DD statements to point to the CEXEC library where you unloaded the ChangeMan ZMF SERCOMC installation libraries.

e For the SERCMD job, change the dataset name in the sample DD statement to point to the CLIST library where you unloaded the ChangeMan ZMF SERCOMC installation libraries.

f For the NEWCMD job, change the CLIST library in the sample DD statement to point to the actual REXX execution library where SERSERVC will reside at runtime.

> **TIP** This may be either a SYSEXEC or SYSPROC library, depending on your installation standards. Run ISRDDN from TSO if you are uncertain about how your REXX execution libraries are handled.

g In the SERVA and SERVP sample DD statements, change the name of the library containing members SERSERVA and SERSERVP from USER.PROCLIB to your actual installation PROCLIB dataset name.

5 Run SERSERVI.

This job installs the HTTP server software in the REXX execution library where it will reside at runtime.

SERCMD Server Control Routine

6 Modify REXX EXEC module SERCMD to use the actual IP address and port number assigned to SERSERVC.

SERCMD is copied to the actual REXX execution library where SERSERVC resides by the SERSERVI install job.

Verifying the Installation of SERSERVC

To verify the installation of SERSERVC, do the following:

1 Start the server by issuing the /S (Start) console command in SDSF:

```
/S SERSERVC
```

2 Ping SERSERVC locally to verify that it is operational.

a At the TSO command line, enter:

```
SERCMD PING
```

b You should receive the response ok.

3 Ping SERSERVC from a Web browser to verify network connectivity.

a From any Web browser, type

http://*ip:port*/?PING

where

ip is the IP address assigned to the LPAR where SERNET resides

port is the port number assigned to the exclusive use of SERSERVC

b You should receive the response ok.

SERSERVC Runtime Considerations

Runtime considerations for SERSERVC include the following startup, shutdown, and timing synchronization issues.

Startup and Shutdown

Console Commands

The SERSERVC HTTP server can be started and stopped using standard console commands in SDSF. To start the server, enter:

```
/S SERSERVC
```

The server can be stopped (cancelled) from SDSF at any time. To stop the server, enter:

```
/C SERSERVC
```

IPL Startup However, SERSERVC is designed for high availability. When testing is complete, consider adding SERSERVC to the list of started tasks that are brought up at IPL time.

Orderly Shutdown It does no harm to cancel SERSERVC with a console command, but orderly shutdown is the preferred method. To initiate an orderly shutdown of the server, type

```
SERCMD SHUTDOWN
```

at the TSO command line. You should receive the response `ok`.

During an orderly shutdown, SERSERVC copies its message log (`msglog`), error log (`stderr`), and standard output (`stdout`) to `SYSOUT *` before terminating execution.

Network Synchronization

SERSERVC requires the local network time to be synchronized with server time on the host. Time zone differences of an integer number of hours are acceptable. However, the minutes and seconds (mm:ss) on the local network clock may not differ by more than 59 seconds from the minutes and seconds on the host clock.

Verifying Host Clock Time To verify that local network time is synchronized with server time on the host, do the following:

1. From a Web browser running on the local network, type

   `http://`*ip:port*`/?TIME`

   where

   `ip` is the IP address assigned to the LPAR where SERNET resides

   `port` is the port number assigned to the exclusive use of SERSERVC

2. You should receive the response `hh:mm:ss`, which is the time on the mainframe where SERSERVC is running.

3. Discard the hours and compare the minutes and seconds on the host with the minutes and seconds reported on your local network. If a difference greater than 59 seconds is found, your local network time must be synchronized to the host.

The mainframe time is considered correct because its clock is built in at manufacture and cannot be changed.

Running Multiple Instances of SERSERVC

You can run multiple instances of SERSERVC if needed. For example, you may want to run a development HTTP server and a production HTTP server. Each instance must have its own directory and a unique port.

To run another instance of the HTTP server:

- Follow the installation instructions in "Installing SERSERVC" on page 191, substituting another name for this instance, such as SERSERVD, in place of SERSERVC.

- Specify a different directory and port for this instance according to your company's installation standards.

Here are examples of JCL segments for a production system and a development system.

Production SERSERVC example:

```
//SERSERVC JOB MSGLEVEL=1
//STARTING EXEC SERSERVC
XXSERSERVC PROC OUTC=H,                             * CLASS
XX             PORT=8188,                           * PORT
XX             PATH='/u/serstart/serservc',         * PATH
XX             PROCLIB='USER.PROCLIB'               * PROCLIB
```

Development SERSERVC example:

```
//SERSERVD JOB MSGLEVEL=1
//STARTING EXEC SERSERVD
XXSERSERVD PROC OUTC=H,                             * CLASS
XX             PORT=6157,                           * PORT
XX             PATH='/u/sernet/serservc',           * PATH
XX             PROCLIB='USER.PROCLIB'               * PROCLIB
```

Appendix C

Configuration File Reference

Serena Release Manager uses configuration files to specify information for the Web services to use. These configuration files contain client connection information, filters, and other information necessary to execute a comprehensive set of release management operations.

This section lists and explains the use of the configuration files.

Configuration Files on the Serena Release Manager Server

Configuration files are used by Serena Release Manager to set implementation-specific details. Most of the configuration files reside on the Serena Release Manager server.

Related Topics

- Files in the Classes Folder
- Files in Other Folders

Files in the Classes Folder

Most of the configuration files on the Serena Release Manager server are located under the `rlm` Web service folder in the `WEB-INF\classes` folder. For example:

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

The configuration file names and purpose are given in the following table.

File Name	Purpose
alf-client-connection.properties	Use this file to specify credentials for the ALF server to be used for notifications sent from the Serena Release Automation server. When the automation deployment task is completed, Serena Release Manager sends an ALF event to SBM server. This requires the credentials to access the ALF server, because Serena Release Automation isn't implemented to use SSO. See Chapter 4, "Configuring Connections using the Release Manager Configurator" on page 49.
bcr.properties	Provides provider definitions for Business Change Requests (BCRs). See Chapter 6, "Configuring Access to Business Change Requests" on page 96.
bcr-connection.properties	Provides connection information for Business Change Requests (BCRs). See Chapter 4, "Configuring Connections using the Release Manager Configurator" on page 49.
commons-logging.properties	Apache log file installed with this software.
dm_qlarius.properties	Provides filtering information for Dimensions CM objects accessed by Release Manager. See Chapter 6, "Configuring Access to Development Change Requests" on page 97 and Chapter 6, "Configuring Access to Deployment Units" on page 100.
dm-client.properties	Provides client-specific information for Dimensions CM. See See Chapter 6, "Configuring Access to Development Change Requests" on page 97 and Chapter 6, "Configuring Access to Deployment Units" on page 100.
dm-client-connection.properties	Provides connection information for Dimensions CM. See Chapter 4, "Configuring Connections using the Release Manager Configurator" on page 49.
itsm.properties	Provides provider definitions for ITSM (for RFCs). See Chapter 6, "Configuring Access to Requests for Change" on page 95.

File Name	Purpose
itsm-connection.properties	Provides connection information for ITSM (for RFCs). See Chapter 4, "Configuring Connections using the Release Manager Configurator" on page 49.
LICENSE.txt	Apache licensing information for use with this software.
log4j.properties	Tells the location and behavior for the primary message log file for Serena Release Manager, rlm.log. See Chapter 9, "Information from Log Files" on page 160.
logback.xml	This file is used for logging confirmation. Do not change this file.
messages.properties	This file is planned for future use to define strings used in the Release Manager Configurator UI.
nolio-client.properties	Provides client-specific information for Serena Release Automation. See Chapter 4, "Configuring Release Automation Communication in Release Manager" on page 69.
nolio-client-connection.properties	Provides connection information for Serena Release Automation. See Chapter 4, "Configuring Connections using the Release Manager Configurator" on page 49.
nolio-client-queries.properties	Provides filtering information for Serena Release Automation. See Chapter 4, "Configuring Release Automation Communication in Release Manager" on page 69.
NOTICE.txt	Notice that Apache software is used and distributed with this software.
providers.properties	Specifies the providers to be used for this implementation of Serena Release Manager. See Chapter 6, "Telling Release Manager Which Providers to Use" on page 105.
rlm.properties	This file is used for global rlm.war settings. Most settings are system settings for Web services. The only setting that you should change is: `rlm.config.service.authdUsers=<list of user IDs delimited by commas>` This defines the list of users who can access the Release Manager Configurator through SSO. See Chapter 4, "Configuring Connections using the Release Manager Configurator" on page 49.
sbm_incidents.properties	Provides provider definitions for SBM Incidents, typically associated with BCRs. See Chapter 6, "Designating the Details for Each BCR Provider" on page 96.
sbm_issues.properties	Provides provider definitions for SBM Issues, typically associated with DCRs. See Chapter 6, "Designating the Details for Each DCR Provider" on page 97.
sbm-client.properties	Sets values for the Serena Release Manager Web services. See Chapter 8, "Setting Maximum Associations for Release Control Objects" on page 155. **CAUTION!** The only values that should be changed in this file are the maximum object association limits for release trains, application releases, and release packages.
sbm-client.xml	System file. Do not change.

File Name	Purpose
`sbm-client-connection.properties`	Provides connection information for SBM. See Chapter 4, "Configuring Connections using the Release Manager Configurator" on page 49.
`sbm-client-model.xml`	System file. Do not change.
`zmf_packages.properties`	Provides filtering information for ChangeMan ZMF. See Chapter 6, "Designating ChangeMan ZMF Deployment Unit Selection Criteria" on page 102.
`zmf-client.properties`	Provides client-specific information for ChangeMan ZMF. See Chapter 4, "Configuring ZMF Communication in Release Manager" on page 63.
`zmf-client-connection.properties`	Provides connection information for ChangeMan ZMF. See Chapter 4, "Configuring Connections using the Release Manager Configurator" on page 49.

Files in Other Folders

Some of the configuration files on the Serena Release Manager server are located in other folders. The configuration file names, location, and purpose are given in the following table.

File Name	Location and Purpose
`alfzmf_resource.properties`	`..\Program Files\Serena\common\tomcat\6.0\webapps\almzmfalf\WEB-INF\conf` ALF event manager information for ChangeMan ZMF. See Chapter 4, "Specifying ALF Event Manager Connection Information for ZMF" on page 64.

Configuration Files on the Dimensions CM Server

Configuration files on the Dimensions CM server are used to configure return communication to Serena Release Manager from Dimensions CM. The configuration file names, location, and purpose are given in the following table.

File Name	Location and Purpose
`dm.cfg`	`..\Program Files\Dimensions 12.1\CM` You must update this file on the Dimensions CM server with Dimensions CM ALF event configuration information Serena Release Manager needs. See Chapter 4, "Specifying Dimensions CM ALF Event Configuration Information" on page 57.
`ALF_EVENTS_CONFIG.XML`	`..\Program Files..\Dimensions\12.1\CM\dfs` You must update this file on the Dimensions CM server to specify selection criteria for Dimensions CM information Serena Release Manager accesses. Chapter 4, "Specifying Selection Criteria for Dimensions CM Events and Objects" on page 58

Configuration Files on the Release Automation Server

Configuration files on the Serena Release Automation server are used to configure return communication to Serena Release Manager from Serena Release Automation. The configuration file names, location, and purpose are given in the following table.

File Name	Location and Purpose
`rest.integration.properties`	`..\Program Files\Serena\Serena Release Automation\conf` Tells Serena Release Automation what server to notify when an event occurs. See Chapter 4, "Specifying the Serena Release Automation Server to Notify" on page 67.

Index