



SERENA[®] **RELEASE MANAGER 2.0**

Installation and Configuration Guide

Serena Proprietary and Confidential Information

Copyright © 2011-2012 Serena Software, Inc. All rights reserved.

This document, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by such license, no part of this publication may be reproduced, photocopied, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Serena. Any reproduction of such software product user documentation, regardless of whether the documentation is reproduced in whole or in part, must be accompanied by this copyright statement in its entirety, without modification.

This document contains proprietary and confidential information, and no reproduction or dissemination of any information contained herein is allowed without the express permission of Serena Software.

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: March 21, 2012

Table of Contents

	Welcome to Serena Release Manager	7
	Guide to Serena Release Manager Documentation	7
	Accessing the Documentation	9
<i>Chapter 1</i>	Installation Prerequisites and Planning	11
	What is Serena Release Manager?	12
	Serena Release Manager Architecture	13
	Preparing for the Installation	13
	Installation Prerequisites	14
	Database Requirements	14
	Software Compatibility Requirements	15
	System Requirements	15
	Planning Checklists and Worksheets	15
<i>Chapter 2</i>	Installation and Configuration Quickstart	17
	Installation and Configuration Checklist	18
<i>Chapter 3</i>	Serena Release Manager Installation	23
	Installation Overview	24
	Installing Serena License Manager	24
	Existing Serena License Manager Systems	24
	New Serena License Manager Systems	25
	Installing Serena Business Manager	25
	Existing Serena Business Manager Systems	25
	New Serena Business Manager Systems	25
	Installing Dimensions CM	26
	Existing Dimensions CM Systems	26
	New Dimensions CM Systems	26
	Installing Serena Release Control	26
	Existing Serena Release Control Systems	27
	New Serena Release Control Systems	27
	Verifying the Web Services Installation	28
	Installing Serena Release Automation	29
	Existing Serena Release Automation Systems	29
	New Serena Release Automation Systems	29
	Installing Other Integrating Serena Products	29
	Applying Licenses	30
<i>Chapter 4</i>	System Activation and Configuration	31
	System Configuration Overview	32
	Serena Release Manager Runtime Communication	32
	Importing and Promoting Serena Release Control	33

Importing the Serena Release Control Solution	34
Creating an Environment for Serena Release Manager	35
Promoting the Snapshots	36
Publishing and Deploying the Process Apps	38
Configuring Required Objects in Serena Release Control	39
Putting the UI Shell files into the SBM Database	39
Creating an Administrative User	39
Configuring the Administrative User Privileges	40
Enabling Serena Release Control Project Roles	41
Configuring the Dashboard Page	42
Configuring Communication on the Dimensions CM Server	43
Specifying the ALF Event Configuration Information	43
Specifying Selection Criteria for Dimensions CM Events and Objects	44
Configuring Dimensions CM Communication in Release Manager	45
Configuring ZMF Communication on the z/OS Mainframe	46
Configuring the Notification URL	47
Configuring a Proxy User ID	48
Configuring TSO User IDs and Permissions	48
Configuring Approvers	49
Configuring ChangeMan ZMF Communication in Release Manager	49
Specifying the Web Server Information for ChangeMan ZMF	49
Specifying ALF Event Manager Connection Information for ZMF	51
Specifying Selection Criteria for ChangeMan ZMF Events and Objects	52
Configuring Communication on the Release Automation Server	52
Specifying the Serena Release Automation Server to Notify	52
Telling Release Automation Which Event Notifications to Send	53
Configuring Release Automation Communication in Release Manager	54
Specifying Release Automation Sign-on Credentials	55
Specifying Release Automation Connection Information	55

Chapter 5

Configuration and Administration of the Integrating Objects	57
Configuring Objects in Serena Release Control	58
Accessing the Standard SBM User Interface	58
Adding Your Application Names in Serena Release Control	58
Adding Your Server Names in Serena Release Control	59
Managing Release Control Users	60
Managing Release Control Reports and Notifications	62
Configuring Objects in Dimensions CM	68
Configuring the Dimensions CM Global Stage Lifecycle	68
Managing Dimensions CM Users	69
Configuring Dimensions CM Projects and Streams	70
Available Selection of Requests and Baselines	71
Configuring Objects in Serena Release Automation	71
Configuring Serena Release Automation Users	71
Configuring Serena Release Automation Processes and Servers	72
Configuring Objects in Serena Business Manager	73

Provider Configuration	75
Provider Configuration Overview	76
Configuring Access to Requests for Change	77
Designating the Details for Each RFC Provider	77
Telling Release Manager Which RFC Providers to Use	78
Configuring Access to Business Change Requests	78
Designating the Details for Each BCR Provider	78
Telling Release Manager Which BCR Providers to Use	79
Configuring Access to Development Change Requests	79
Designating the Details for Each DCR Provider	80
Telling Release Manager Which DCR Providers to Use	82
Configuring Access to Deployment Units	83
Designating the Details for Each DU Provider	83
Specifying Providers to Use	87

Serena Release Manager Upgrade	89
Upgrading from Serena Release Manager v1.1 to v2.0	90

Serena Release Manager Customization	97
Customizing the User Interface	98
Customizing the Serena Release Control Dashboard	98
Configuring the Calendar	100
Configuring the Inbox	101
Configuring Views and Dialog Boxes	103
Customizing Release Control Workflows	105
Modifying Release Types and Stages	106
Summary of Adding a Stage	107
Example of Adding a Stage	108
Adding Provider Connections	119
Creating a Class for Your Provider	120
Creating Properties Files for Your Providers	120
Building and Packaging	123
Telling Serena Release Manager to Use This Provider	123
Configuring Release Manager to Use a Different Port	124
Changing the Port on Which the Common Web Server Runs	125
Configuring a Non-Default Web Server Port in the Process Apps	126
Changing the Web Services to Point to a Different Port	128
Customizing the Serena Release Control Custom Shell	128
Log In Page with SSO	129
Other Pages	130
Applying the Changes	130
Activating Environment Association to Release Packages	131
Activating the Release Manager Objects for SSM	131
How SSM Detects Whether Serena Release Manager is Installed	132
Configuring Reports that SSM References	132
Configuring Events to Transition SSM Changes	132
Extending the Calendar	133

Troubleshooting	135
Troubleshooting Overview	136
Information from the Serena Release Control User Interface	136
Error Messages	136
Activity Logs	136
History	136
Information from the Product Log File	137
Symptoms and Solutions	137
Unexpected Display Results	137
Matches Not Found for Selections	139
Release Package Deployment Fails	139
Installer Errors	140
Snapshot Promotion Errors	140
Slow Response Time	141
Release Package Deployment Fails	142
Create Release Fails	142
Dimensions CM DCR Projects Appear in the Wrong Section	143

Workflow Reference	145
Workflow Relationships	146
Workflow Dependencies	146
Release Train Workflow	148
Application Release Workflow	148
Release Package Workflow	149
Start and Development States	150
Integration State	151
Staging and Production States	152
Exceptions State	153
Deployment Task Workflows	153
Automation Deployment Task Workflow	154
Manual Deployment Task Workflow	155
Vault Deployment Task Workflows	155
Deployment Process Template Workflow	158
Environment Workflow	159
Index	161

Welcome to Serena Release Manager

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

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

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

Before You Begin See the Readme for the latest updates and corrections for this manual.

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
<i>Serena Business Manager Installation and Configuration Guide</i>	Provides information on installing SBM and creating a database. Database and Web server configuration information is also provided.
<i>Serena Business Manager Composer Guide</i>	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.
<i>Serena Business Manager Application Administrator Guide</i>	Explains how to configure deployed applications. Instructions for managing projects, user and group accounts, and notifications are included.
<i>Serena Business Manager Application Repository Guide</i>	Provides information on using SBM Application Repository to deploy process apps to runtime environments and to promote configured applications from one environment to another.

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

Manual or Tutorial	Description
<i>Serena Business Manager Installation and Configuration Guide</i>	Provides information on installing SBM and creating a database. Database and Web server configuration information is also provided. (PDF manual)
<i>Serena Business Manager Composer Guide</i>	Provides details on using SBM Composer to create the tables, fields, workflows, orchestrations, and other design elements comprised in process apps. Information about saving, versioning, importing, and exporting process apps is also provided. This document is intended for individuals who want to design and maintain process apps. (PDF manual)
<i>Serena Business Manager Application Administrator Guide</i>	Provides information on using Application Administrator to deploy process apps to runtime environments and to promote configured applications from one environment to another. (PDF manual)
<i>Serena Business Manager System Administrator Guide</i>	Provides information on configuring and administering the SBM Application Engine. Instructions for managing projects, user accounts, system settings, and authentication are included. (PDF manual)
<i>Serena Business Manager Licensing Guide</i>	Explains how to manage licenses for Serena Business Manager. License types are discussed, along with instructions for installing and using the Serena License Manager. This guide is intended for administrators who will install and implement Serena Business Manager. (PDF manual)
<i>Serena Business Manager User's Guide</i>	Provides information about the SBM User Workspace and is intended for end users. Instructions on using the SBM User Workspace, including the robust reporting feature in SBM, are included. To ease the process of providing a copy for every user in your system, the Serena Business Manager User's Guide is provided in PDF and can be accessed from the Product Information tab of the About page in the SBM User Workspace. (PDF manual)
<i>Serena Release Automation Center Installation and Administration Guide</i>	Describes how install and configure Serena Release Automation. (PDF manual)
<i>Serena Dimensions CM Installation Guides for Windows and for UNIX</i>	Describes how install and configure Dimensions CM for the respective platforms. (PDF manuals)
<i>Serena Dimensions CM Administrator's Guide</i>	Describes how to administer the Dimensions CM tool. (PDF manual)
<i>Serena Dimensions CM and RM Installing the Serena License Manager</i>	Describes installing the Windows version of SLM. (PDF manual)
<i>Serena Dimensions CM User's Guide</i>	A user guide to the Dimensions CM Client Tools. (PDF manual)
<i>Serena Dimensions CM Process Modeling Guide</i>	Describes how to configure the process model using the Administration Console. (PDF manual)

Manual or Tutorial	Description
<i>Serena ChangeMan ZMF Administrator's Guide</i>	Describes ChangeMan ZMF features and functions with instructions for choosing options and configuring global and application administration parameters. (PDF manual)
<i>Serena ChangeMan ZMF Installation Guide</i>	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. (PDF manual)
<i>SER10TY User's Guide</i>	Gives instructions for applying licenses to enable ChangeMan ZMF and its selectable options. (PDF manual)
<i>Serena Service Manager Installation and Configuration Guide</i>	Provides information on installing and configuring Serena Service Manager. (PDF manual)
<i>Serena Service Manager User's Guide</i>	Provides information about the Serena Service Manager default implementation user interface and is intended for end users. (PDF manual)

Accessing the Documentation

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

When you click the **Help** link in Serena Release Control, the documentation center 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.

To download documentation and tutorials for current or prior releases of the Serena Release Manager suite, see the Serena Customer Support website [My Downloads page](#).

Chapter 1

Installation Prerequisites and Planning

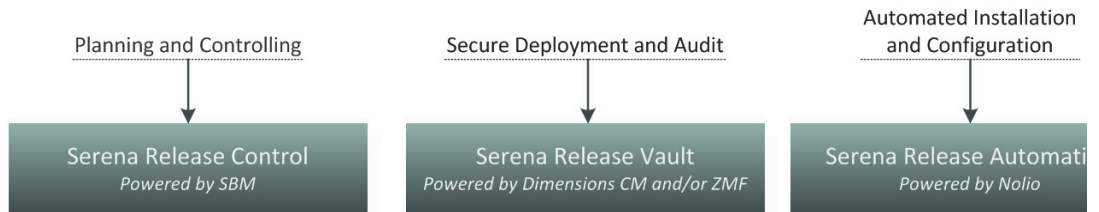
This section tells you what you need to do before installing Serena Release Manager. It provides you with information to help guide you through the Serena Release Manager installation.

What is Serena Release Manager?	12
Serena Release Manager Architecture	13
Preparing for the Installation	13

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 Manager

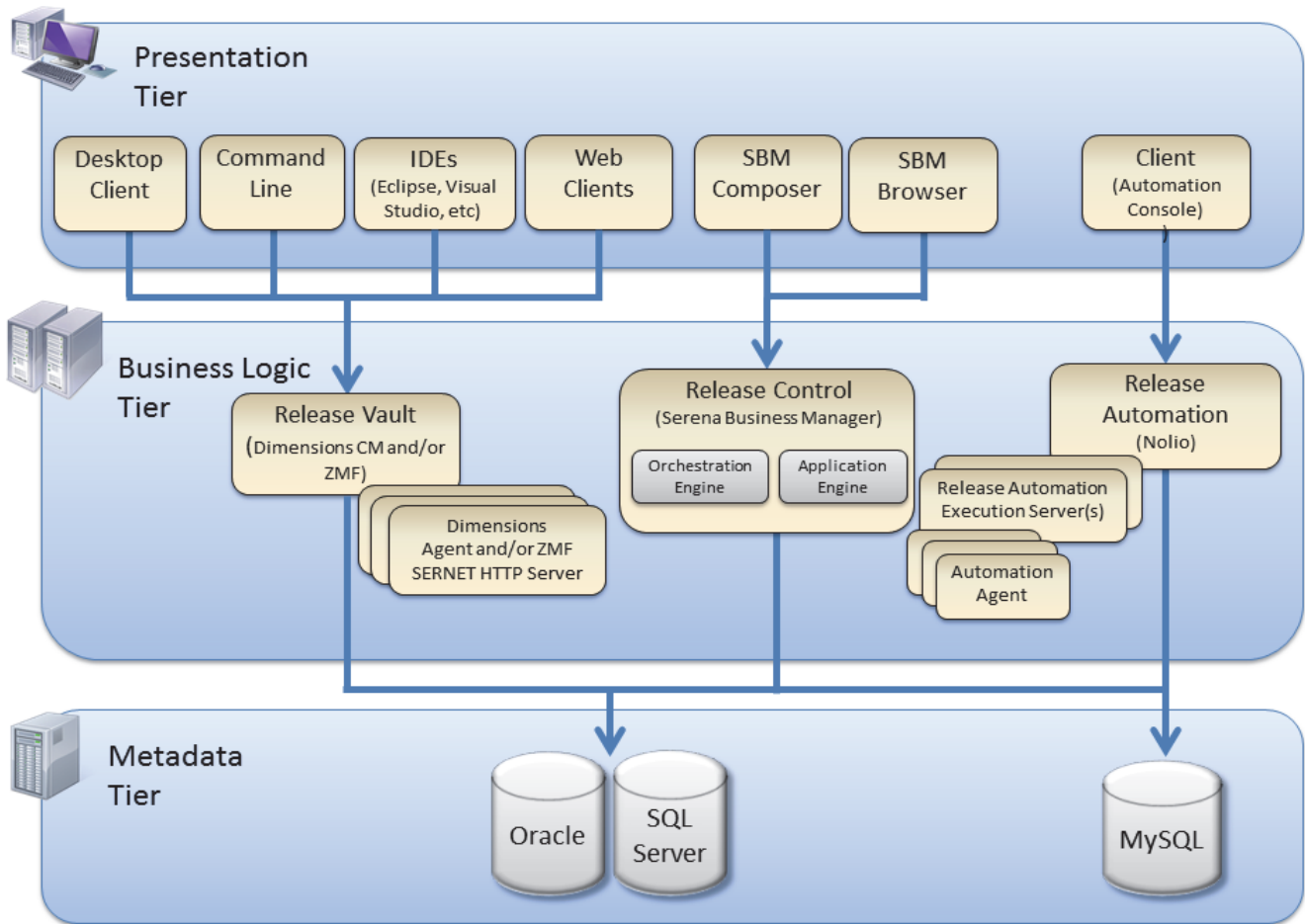


- **Serena Release Control**, powered by Serena Business Manager, helps you plan and control your application release processes across your enterprise, from definition to deployment.
- **Serena Release Vault**, powered by Dimensions CM for distributed systems and ChangeMan ZMF for IBM z/OS systems, ensures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



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 14
- "Database Requirements" on page 14
- "Software Compatibility Requirements" on page 15
- "System Requirements" on page 15
- "Planning Checklists and Worksheets" on page 15

Installation Prerequisites

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

Preparing for Serena Product Installation

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

- **Serena Business Manager**

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

- **Serena Release Control**

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

- **Dimensions CM**

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

- **Serena License Manager**

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

- **Common Supporting Files**

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



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 <i>Serena Business Manager Installation and Configuration Guide</i> .
Serena Release Control	Your SBM requirements address the requirements for Serena Release Control.
Serena Dimensions CM	<ul style="list-style-type: none"> ■ 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 <i>Serena Release Automation Center Installation and Administration Guide</i> .
ChangeMan ZMF (Optional)	"System Requirements" in <i>Serena ChangeMan ZMF Installation Guide</i> .

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 17.

Chapter 2

Installation and Configuration Quickstart

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

Installation and Configuration Checklist	18
1: Installation	18
2: Process Apps	18
3: Release Control Configuration	18
4: Dimensions CM Communication	19
5: ChangeMan ZMF Communication	19
6: Release Automation Communication	20
7: Integrating objects	20
8: Provider properties	21
9: Customization	21

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	<ul style="list-style-type: none"> <input type="checkbox"/> Install the products you plan to use in the suite and apply licenses to the products as needed. <p>See "Serena Release Manager Installation" on page 23 for details.</p>
2: Process Apps	<p>Import the Serena Release Control SBM solution and promote the snapshots of the process apps.</p> <ul style="list-style-type: none"> <input type="checkbox"/> Import the Serena Release Control solution, which contains a snapshot of each of the process apps. <input type="checkbox"/> Create a new application server environment for the process apps if needed. <input type="checkbox"/> Promote each of the snapshots, creating and selecting endpoints as needed. Use Security Token authentication for the endpoints. <input type="checkbox"/> Publish and deploy the process apps to upgrade the structure of the RESTGrid widgets. <p>See "Importing and Promoting Serena Release Control" on page 33 for details.</p>
3: Release Control Configuration	<p>Configure required objects in Serena Release Control as follows:</p> <ul style="list-style-type: none"> <input type="checkbox"/> Put the user interface shell files into the SBM database to ensure Serena Release Control elements appear as designed. <input type="checkbox"/> 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, <code>rladmin</code>. <input type="checkbox"/> Enable roles for Serena Release Control projects and verify that Serena Release Control is activated. <input type="checkbox"/> Configure the Dashboard page as needed. <p>See "Configuring Required Objects in Serena Release Control" on page 39 for details.</p>

Step	Action
4: Dimensions CM Communication	<p>Configure communication with Dimensions CM (Windows/UNIX systems release vault).</p> <ol style="list-style-type: none"> 1 Configure Dimensions CM communication on the Dimensions CM server. <ul style="list-style-type: none"> <input type="checkbox"/> Specify ALF event configuration information in the <code>dm.cfg</code> file. <input type="checkbox"/> Specify selection criteria for the Dimensions CM events and objects by updating the ALF event configuration file, <code>ALF_EVENTS_CONFIG.XML</code>. Specify your Dimensions CM database name, project name, baseline type, and deploy event. <p>See "Configuring Communication on the Dimensions CM Server" on page 43 for details.</p> 2 Configure Dimensions CM communication in Serena Release Manager. <ul style="list-style-type: none"> <input type="checkbox"/> Specify the Dimensions CM endpoints and user IDs in the common Tomcat Web server <code>webapps\r1m\WEB-INF\classes</code> folder <code>dimensions.properties</code> file. <p>See "Configuring Dimensions CM Communication in Release Manager" on page 45 for details.</p>
5: ChangeMan ZMF Communication	<p>Configure communication with ChangeMan ZMF (z/OS systems release vault).</p> <ol style="list-style-type: none"> 1 Configure ChangeMan ZMF communication on the z/OS mainframe. <ul style="list-style-type: none"> <input type="checkbox"/> Configure the <code>NTFYURL</code>; this is the URL Serena Release Manager uses to send information to SERNET through the server. <input type="checkbox"/> Configure a proxy user ID for each mainframe host, or LPAR, that Serena Release Manager uses to log in to ChangeMan ZMF. <input type="checkbox"/> Configure TSO user IDs that match the SBM user IDs. <input type="checkbox"/> Configure approvers. <p>See "Configuring ZMF Communication on the z/OS Mainframe" on page 46 for details.</p> 2 Configure ChangeMan ZMF communication in Serena Release Manager. <ul style="list-style-type: none"> <input type="checkbox"/> Specify sign-on information in the common Tomcat Web server <code>webapps\r1m\WEB-INF\classes</code> folder <code>zmf.properties</code> file. <input type="checkbox"/> Specify ALF event manager connection information in the common Tomcat Web server <code>webapps\zmfalf\WEB-INF\conf</code> folder <code>alfzmf_resources.properties</code> file. <input type="checkbox"/> Specify selection criteria for ZMF events and objects in the common Tomcat Web server <code>webapps\r1m\WEB-INF\classes</code> folder <code>zmf_packages.properties</code> file. <p>See "Configuring ChangeMan ZMF Communication in Release Manager" on page 49 for details.</p>

Step	Action
6: Release Automation Communication	<p>Configure communication with Serena Release Automation.</p> <ol style="list-style-type: none"> 1 Configure Serena Release Automation communication on the Serena Release Automation server. <ul style="list-style-type: none"> <input type="checkbox"/> Specify the Serena Release Automation server to notify when an event occurs in the Serena Release Automation <code>rest.integration.properties</code> file. If the file does not exist, create it. <input type="checkbox"/> Update the Serena Release Automation environment notifications for each application to tell Serena Release Automation the events about which to notify Serena Release Manager. <p>See "Configuring Communication on the Release Automation Server" on page 52 for details.</p> 2 Configure Serena Release Automation communication in Serena Release Manager. <ul style="list-style-type: none"> <input type="checkbox"/> Update the Serena Release Automation sign-on credentials in the common Tomcat Web server <code>webapps\rlm\WEB-INF\classes</code> folder <code>alf.properties</code> file. <input type="checkbox"/> Specify the Serena Release Automation connection information in the common Tomcat Web server <code>webapps\rlm\WEB-INF\classes</code> folder <code>noio-client.properties</code> file. <p>See "Configuring Release Automation Communication in Release Manager" on page 54 for details.</p>
7: Integrating objects	<p>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:</p> <ul style="list-style-type: none"> <input type="checkbox"/> Release Control (SBM) users, general reports, Dashboard page, and notifications <input type="checkbox"/> SBM projects (to provide BCRs or DCRs, for example) <input type="checkbox"/> Applications and environments (servers) (specific to your organization; add to the respective auxiliary tables) <input type="checkbox"/> Dimensions CM process model (GSL), projects and streams, baselines, and requests <input type="checkbox"/> ChangeMan ZMF applications, sites, promotion levels, approvals, and change packages <input type="checkbox"/> Release Automation environments, applications, processes, and servers <p>See "Configuration and Administration of the Integrating Objects" on page 57 for details.</p>

Step	Action
8: Provider properties	<p>Configure the provider properties as needed in the common Tomcat Web server <code>webapps\rlm\WEB-INF\classes</code> folder. These may include but aren't limited to the following:</p> <ul style="list-style-type: none"> <input type="checkbox"/> Serena Service Manager (SSM) request for change (RFC) properties (for example: <code>itsm.properties</code> and <code>providers.properties</code>) <input type="checkbox"/> Serena Business Manager (SBM) business change request (BCR) properties (for example: <code>bcr.properties</code> and <code>providers.properties</code>) <input type="checkbox"/> Serena Business Manager (SBM) development change request (DCR) properties (for example: <code>issues.properties</code> and <code>providers.properties</code>) <input type="checkbox"/> Dimensions CM deployment unit (DU) properties (for example: <code>dm_qlarius.properties</code> and <code>providers.properties</code>) <input type="checkbox"/> ChangeMan ZMF deployment unit (DU) properties (for example: <code>zmf_packages.properties</code> and <code>providers.properties</code>) <p>See "Provider Configuration" on page 75 for details.</p>
9: Customization	<p>Complete customization and additional configuration as needed.</p> <p>See "Serena Release Manager Customization" on page 97 for details.</p>



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	24
Installing Serena License Manager	24
Installing Serena Business Manager	25
Installing Dimensions CM	26
Installing Serena Release Control	26
Installing Serena Release Automation	29
Installing Other Integrating Serena Products	29
Applying Licenses	30

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 24](#).
- 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 25](#).
- 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 29](#).
- 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 26](#).
- 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 29](#).
- 6 Ensure that other providers that you plan to use are installed as needed. See ["Installing Other Integrating Serena Products" on page 29](#).

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. You do need to do the following:

- Ensure that you are running a supported version of Serena License Manager as indicated in the Serena Release Manager Readme.
- After installing the rest of the products in the suite, apply the licenses as needed. See "Applying Licenses" on page 30.

Please continue to ["Installing Serena Business Manager" on page 25](#).

New Serena License Manager Systems

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

Documentation References

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

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

Installing Serena Business Manager

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

Existing Serena Business Manager Systems

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

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

Please continue to ["Installing Serena Release Control" on page 26](#).

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 do need to do the following:

- Ensure that you are running a supported version of Dimensions CM as indicated in the Serena Release Manager Readme.
- Ensure that Single Sign On (SSO) is enabled for Dimensions CM. For configuring SSO if not already enabled, see Dimensions CM Support for SSO in the Dimensions CM Administrator's Guide.
- After installing the rest of the products in the suite, follow the post-installation system configuration instructions for Dimensions CM.

Please continue to ["Installing Serena Release Automation"](#) on page 29.

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 89](#).

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 25](#).

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 31](#).

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 124](#) 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 Click **Next**.

The **Configuration Details** page appears.

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

7 Click **Next**.

The **Install** page appears.

8 Click **Install**.

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

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

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

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

9 Click **Finish**.

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

```
C:\Program Files (x86)\Serena\Solutions\Release Control\  
  com.serena.rlm.sbm.shell.zip  
  \solution (contains the solution file)  
  \war (contains the war files)
```

The installer does the following automatically:

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

Verifying the Web Services Installation

Verify that the Serena Release Manager Web services were installed and started successfully.

To verify the Web Services:

- 1** In your Web browser, browse to the home page for your Tomcat installation. The default location is:

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

- 2** The **Apache Software Foundation** page appears with a list of services.

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

- RLMUtilService
- ReleaseRequestService
- DeploymentAutomationService
- DeployUnitService

- 3 Close the browser when you are finished.

Installing Serena Release Automation

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

Existing Serena Release Automation Systems

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

- Ensure that you are running a supported version of Serena Release Automation as indicated in the Serena Release Manager Readme.
- Ensure that the supported database you are using for your Serena Release Automation system is open for remote access according to the database system documentation.
Supported database systems include MySQL, Microsoft SQL Server, and Oracle.
- After installing the rest of the products in the suite, follow the post-installation system configuration instructions for Serena Release Automation.

If you plan to use ChangeMan ZMF as one of your release vaults, please continue to ["Installing Other Integrating Serena Products" on page 29](#). Otherwise, please continue to [Chapter 4, "System Activation and Configuration" on page 31](#).

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 Center 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 Center Installation and Administration Guide* in "Updating Serena Release Automation Center 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	32
Importing and Promoting Serena Release Control	33
Configuring Required Objects in Serena Release Control	39
Configuring Communication on the Dimensions CM Server	43
Configuring Dimensions CM Communication in Release Manager	45
Configuring ZMF Communication on the z/OS Mainframe	46
Configuring ChangeMan ZMF Communication in Release Manager	49
Configuring Communication on the Release Automation Server	52
Configuring Release Automation Communication in Release Manager	54

System Configuration Overview

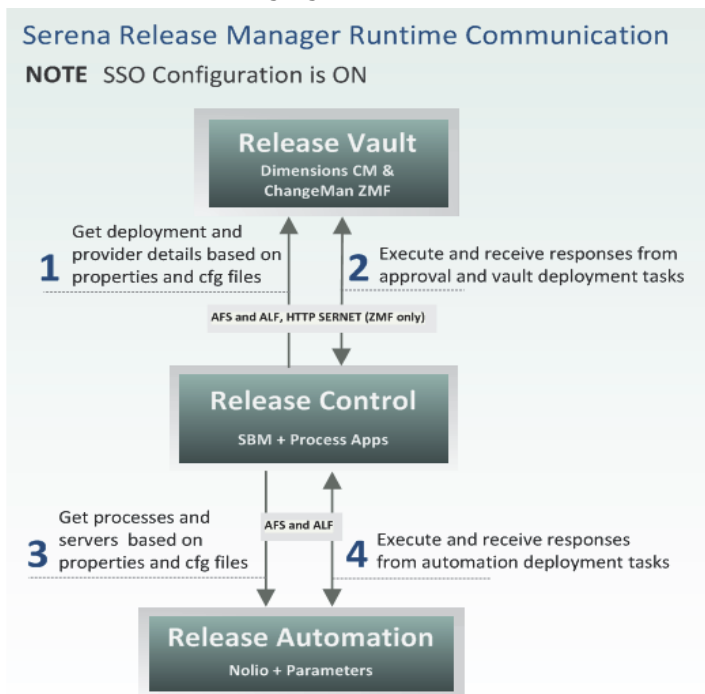
In the system configuration, you are enabling the communication between the components of Serena Release Manager.

So that Serena Release Manager can send and receive information on release management information and deployment tasks from Serena Release Automation and from release vaults, such as Dimensions CM and ChangeMan ZMF, you must configure the specific communication information for your environment.

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.

Serena Release Manager Runtime Communication

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



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.

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:
 - ["Importing and Promoting Serena Release Control" on page 33](#)
 - ["Configuring Required Objects in Serena Release Control" on page 39](#)
- For Dimensions CM:
 - ["Configuring Communication on the Dimensions CM Server" on page 43](#)
 - ["Configuring Dimensions CM Communication in Release Manager" on page 45](#)
- For ChangeMan ZMF:
 - ["Configuring ZMF Communication on the z/OS Mainframe" on page 46](#)
 - ["Configuring ChangeMan ZMF Communication in Release Manager" on page 49](#)
- For Serena Release Automation:
 - ["Configuring Communication on the Release Automation Server" on page 52](#)
 - ["Configuring Release Automation Communication in Release Manager" on page 54](#)

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 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.
- 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 25](#). 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.

Detailed steps are given in the following sections:

- ["Importing the Serena Release Control Solution" on page 34](#)
- ["Creating an Environment for Serena Release Manager" on page 35](#)
- ["Promoting the Snapshots" on page 36](#)
- ["Publishing and Deploying the Process Apps" on page 38](#)

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 2.0.m.bbb,

where m is maintenance release number and bbb is build number.

- 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 server environment for your Serena Release Control process application unless you are promoting it into an existing environment.

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

The procedures are 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.
 - 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 localhost and specify the port number for the server. For example:

```
http://localhost:80/gsoap/gsoap_ssl.dll?sbminternalservices72
```
- 4** Click **Test Connection** to test the connection.
- 5** Click **OK**.

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 localhost as the hostname.

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

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



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

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

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 localhost as the hostname. For example:


```
http://localhost:8085/eventmanager/services/ALFAdmin
```
 - d Click **Test Connection** to test the connection.
 - e Click **OK**.

Promoting the Snapshots

You must promote 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.



NOTE If you plan to run your implementation of Serena Release Manager on a port other than the default port of 9095, please see [Chapter 8, "Configuring Release Manager to Use a Different Port"](#) on page 124 before proceeding.

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 in the order suggested in the preceding section 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.
- 10 If the Destination Endpoint has not yet been defined for the selected Source, click **Create a new endpoint**.

To create and select an endpoint:

- a Name the endpoint.



TIP Name the new endpoint the same as the Source Endpoint for which 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	<code>http://localhost:9095/rlm/services/RLMUtilService</code>
ReleaseRequestService	<code>http://localhost:9095/rlm/services/ReleaseRequestService</code>

- 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.

11 Click **Done**.

The **Summary** page appears.

12 Click **Promote**.

The **Promotion Started** page appears.

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

14 Repeat the procedure for each snapshot.



PRIVILEGES Privileges for deleting or modifying the Serena Release 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.

Publishing and Deploying the Process Apps

After promoting the snapshots a single time in SBM Application Repository, you must then publish and deploy the process apps from SBM Composer. This is required to upgrade the structure of the RESTGrid widgets.

To publish and deploy the process apps in SBM:

- 1 In SBM Composer, select **Open** from the Composer menu and open a process app from the repository.
- 2 Click **Publish**.
- 3 Close the process app; check in to the repository when prompted.
- 4 Repeat for each process app.
- 5 In the SBM Application Repository, click **Process Apps** in the navigation pane and then open a process app.
- 6 Click **Deploy**.
- 7 Map endpoints as needed as you did when you promoted the snapshots.
- 8 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 Put the user interface shell files into the SBM database to ensure Serena Release Control user interface elements appear as designed.
- 2 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, `rladmin`.
- 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.

For details, see the following:

- "Putting the UI Shell files into the SBM Database" on page 39
- "Creating an Administrative User" on page 39
- "Configuring the Administrative User Privileges" on page 40
- "Enabling Serena Release Control Project Roles" on page 41
- "Configuring the Dashboard Page" on page 42

Putting the UI Shell files into the SBM Database

You must put the user interface shell files into the SBM database to ensure Serena Release Control elements appear as designed.

To put the files into the database:

- 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 must be done after you have promoted the snapshots, because SBM overlays certain UI elements during the promotion and deployment processes.

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.

The screenshot shows the SBM Administrator web interface. The browser address bar displays `sbm/tmtrack/tmtrack.dll?StdPage&Template=newwebadmin/index.html`. The page title is "Privileges" and the user is logged in as "admin". The breadcrumb trail is "Administrator Portal > Users > rlmadmin >".

In the "Tables" section, the following tables are listed:

- Application
- Stage (highlighted)
- Related Deploy Units
- Related Projects
- Related Requests
- ReleaseType_Stage
- Deployment Submit Map
- Related RFCs
- Related BCRs

The "Grant Privileges" section shows the following table:

Privileges	
Submit	<input checked="" type="checkbox"/>
Update	<input checked="" type="checkbox"/>
Delete	<input checked="" type="checkbox"/>
View	<input checked="" type="checkbox"/>

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.

Name	<input checked="" type="checkbox"/> rlmadmin
Release Manager	<input checked="" type="checkbox"/> 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 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 the ALF Event Configuration Information" on page 43](#)
- ["Specifying Selection Criteria for Dimensions CM Events and Objects" on page 44](#)

Specifying the 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 75](#).

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 communication on the Serena Release Manager server so that Serena Release Manager can connect to Dimensions CM. To do this, you must update the Dimensions CM Web service properties file with the correct connection information as follows:

- In the Serena Release Manager common Tomcat Web server classes folder, update the dimensions.properties file to specify the Dimensions CM endpoints and user IDs.

To update the Dimensions CM connection information:

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

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

- 2 Open the dimensions.properties file.

- 3 Make sure the variables are set to the correct connection information for your installation of Dimensions CM 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.
DIM_DBCONN	The Dimensions CM network instance to which you want to connect.
DIM_SERVER	Hostname where the Dimensions CM server is running.
JOB_STATE_SUCCESS	State that indicates success.
JOB_STATE_FAILURE	State that indicates failure.
DIM_USERID	User ID for the Dimensions CM administrative user for Serena Release Manager.
DIM_PASSWORD	Password for the Dimensions CM administrative user for Serena Release Manager.

- 4 Restart the Serena common Tomcat service.

Example

dimensions.properties

```
DIM_WS_URL = http://dimcm_host:8080/dmwebservices2/services/dmwebservices/

DIM_DBNAME = cm_typical
DIM_DBCONN = Dim12
DIM_SERVER = dimcm_host

JOB_STATE_SUCCESS = Succeeded
JOB_STATE_FAILED = Failed

# for testing
DIM_USERID = rlmadmin
DIM_PASSWORD = rlmadmin_pswd
```

Configuring ZMF Communication on the z/OS 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 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 47
- "Configuring a Proxy User ID" on page 48
- "Configuring TSO User IDs and Permissions" on page 48
- "Configuring Approvers" on page 49

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/zmfalf/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/zmfalf/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 JESMSGLOG (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 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. Once 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 User's 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 ChangeMan ZMF Communication in Release Manager

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

- Specify sign-on information in the Serena Release Manager common Tomcat Web server `classes` folder `zmf.properties` file.
- Specify ALF event manager connection information for ChangeMan ZMF in the Serena Release Manager common Tomcat Web server `classes` folder `alfzmf_resource.properties` file.
- Specify selection criteria for the ChangeMan ZMF events and objects in the Serena Release Manager common Tomcat Web server `classes` folder `zmf_packages.properties` file.

For details, see the following:

- ["Specifying the Web Server Information for ChangeMan ZMF" on page 49](#)
- ["Specifying ALF Event Manager Connection Information for ZMF" on page 51](#)
- ["Specifying Selection Criteria for ChangeMan ZMF Events and Objects" on page 52](#)

Specifying the Web Server Information for ChangeMan ZMF

So that Serena Release Manager can receive information from ChangeMan ZMF, you must update the ChangeMan ZMF configuration settings with the endpoints and sign-on credentials SBM uses for the connection.

The connection of SBM with ChangeMan ZMF is implemented using ALF events. Serena Release Manager will get the information through SBM.

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 `zmf.properties` file.
- 3 Set the properties for the connection as follows:

Parameter	Value
ZMF_WS_URL	URL to the server, or LPAR, where the RLM, ZMF, ZMFWS, ZMFALF Web services are installed, in the form of: <code>http://<hostname>:<port>/zmf/services/ZMFPackageServices/</code>
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 48 .
ZMF_SERVER_PROXY_PASSWORD	Password for the proxy user ID.

- 4 Restart the Serena common Tomcat service.

Example

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

`zmf.properties`

```
# Specify the hostname and portid of the tomcat instance where the rlm, zmf, zmfws, zmfalf web applications
# are installed.
ZMF_WS_URL = http://localhost:9095/zmf/services/ZMFPackageServices/

# Specify the zOS Mainframe hostname or ip address where the ZMF server started task is running.
ZMF_SERVER_HOSTADDRESS    = zmf_host

# Specify the portid assigned to the ZMF server started task.
ZMF_SERVER_HOSTPORT      = 8080

# Specify the ZMF Server proxy userid and password to use to login on behalf of a SBM user to ZMF.
# All SBM users that require access to ZMF must have a mainframe userid similar to their SBM userid.
ZMF_SERVER_PROXYID       = RLMMAN
ZMF_SERVER_PROXY_PASSWORD = RLMPWD1

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

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\zmfalf\WEB-INF\conf folder. For example:
C:\Program Files\Serena\common\tomcat\6.0\webapps\zmfalf\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 Selection Criteria for ChangeMan ZMF Events and Objects

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 as documented in [Chapter 6, "Configuring Access to Deployment Units" on page 83](#).

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 52](#)
- ["Telling Release Automation Which Event Notifications to Send" on page 53](#)

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 `nolionnotification` servlet as follows:

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

For example:

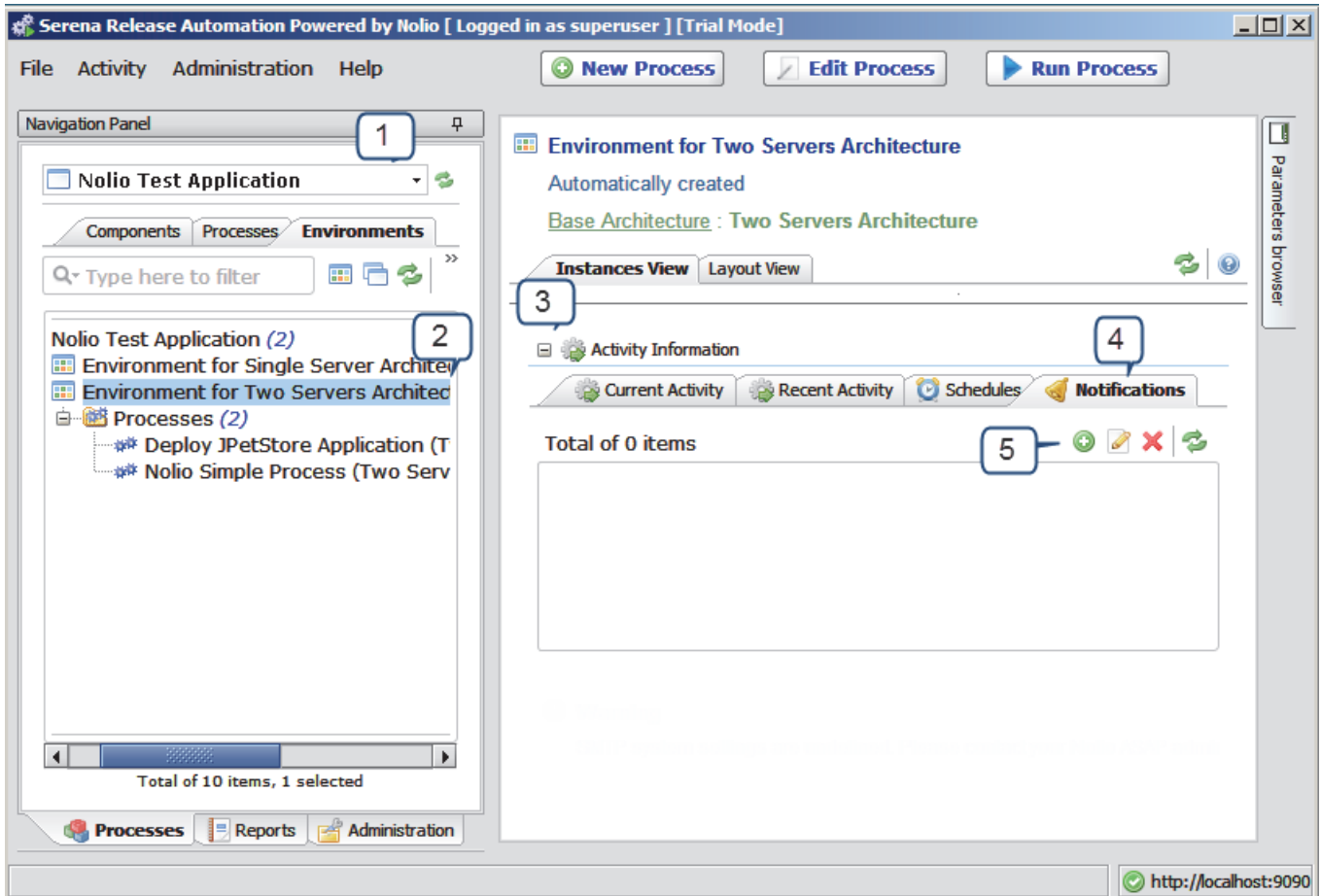
```
target.url=http://rlmhost:9095/rlm/servlet/nolionnotification
```

- 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 Center is shown in the following figure.



To configure Serena Release Automation environment notification:

- 1 From Serena Release Automation Center, for each application, select the application.
- 2 For each environment, select the environment.
- 3 Expand the **Activity Information** section.
- 4 Select the **Notifications** tab.
- 5 Click the **Add or modify notification settings** button.

The **Edit Environment Notification** dialog box appears.

- 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 on the Serena Release Manager server so that Serena Release Manager can connect to Serena Release Automation. To do this, you must do the following:

- Update the Serena Release Automation sign-on credentials in the Serena Release Manager common Tomcat Web server `classes` folder `alf.properties` file.
- Specify the Serena Release Automation connection information in the Serena Release Manager common Tomcat Web server `classes` folder `nlio-client.properties` file.

For details, see the following:

- "Specifying Release Automation Sign-on Credentials" on page 55
- "Specifying Release Automation Connection Information" on page 55

Specifying Release Automation Sign-on Credentials

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

To update the Serena Release Automation sign-on credentials:

- 1 Navigate to the Serena Release Manager common Tomcat Web server `classes` folder.
For example:
`C:\Program Files\Serena\common\tomcat\6.0\webapps\rlm\WEB-INF\classes`
- 2 Open the `alf.properties` file.
- 3 Set the `AE_USERID` and `AE_PASSWORD` variables with the user ID and password of your Serena Release Manager administrative user.
- 4 Restart the Serena Common JBOSS and IIS Admin Service services.

Example

`alf.properties`

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

Specifying Release Automation Connection Information

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

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.properties` file.

3 Set the properties for the connection 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.
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. Login credentials are specified for the Serena Release Automation execution server.

```
nolio-client.properties
```

```
nolio.ws.openapi.uri=http://relauto_host:9090/datamanagement/ws/OpenAPIService?wsdl

nolio.username=superuser
nolio.password=suser

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_FAILED_PAUSED,PRE_FAILED,FLOW_STOPPED
```


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 need to 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	58
Configuring Objects in Dimensions CM	68
Configuring Objects in Serena Release Automation	71
Configuring Objects in Serena Business Manager	73

Configuring Objects in Serena Release Control

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

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

- "Accessing the Standard SBM User Interface" on page 58
- "Adding Your Application Names in Serena Release Control" on page 58
- "Adding Your Server Names in Serena Release Control" on page 59
- "Managing Release Control Users" on page 60
- "Configuring Reports" on page 62
- "Configuring Notifications" on page 65

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://r1mhost/tmtrack/tmtrack.dll?shell=r1m
```

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

```
http://r1mhost/tmtrack/tmtrack.dll?
```

Adding Your Application Names in Serena Release Control

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

What Can You Change?

You can change the following application information:

- Add application name and description.

What is the Impact?

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

How Do You Change It?

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

To change the Application table entries:

- 1 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?

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?

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 will need to 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 will also need to:

- 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.
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.

Role Name	Description
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 and Notifications

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

Configuring Reports

Some of the Serena Business Manager reports are used to populate the Serena Release Control user interface views. You can configure those as documented in [Chapter 8, "Customizing the User Interface"](#) on page 98.

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, "Customizing the Serena Release Control Dashboard" on page 98.
	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 Calendar" on page 100.
	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 101.
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 101.
	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	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 101.
	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 101.
	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.
	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?

You can change the following report information:

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

What is the Impact?

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

How Do You Change It?

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



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

Configuring Notifications

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

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

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

Entity	Notifications
Release Package	RP - Any Release Package changes owner
	RP - Any Release Package changes state
	RP - Any Release Package changes to inactive
	RP - Any Release Package I submitted changed state
	RP - Any Release Package I submitted changed to inactive
	RP - Any Release Package is submitted
	RP - I become the owner of any Release Package
Deployment Task	D - Any 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
	Deployment Process Template
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

Entity	Notifications
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
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?

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 entities needed.

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

- ["Configuring the Dimensions CM Global Stage Lifecycle" on page 68](#)
- ["Managing Dimensions CM Users" on page 69](#)
- ["Configuring Dimensions CM Projects and Streams" on page 70](#)
- ["Available Selection of Requests and Baselines" on page 71](#)

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?

- 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?

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 75.

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 entities needed.

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

- ["Configuring Serena Release Automation Users"](#) on page 71
- ["Configuring Serena Release Automation Processes and Servers"](#) on page 72

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. The Serena Release Automation MySQL database must be open for remote access for the communication between Serena Release Manager and Serena Release Automation to work.

What is the Impact?

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

How Do You Change It?

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 Center Installation and Administration Guide*.

Configuring Serena Release Automation Processes and Servers

Serena Release Manager relies on the information that you configure as part of the ongoing use of Serena Release Automation.

If you use Serena Release Automation to install and configure deployment units on the servers, or environments, to which you deploy, you must configure the required information in Serena Release Automation. Information required from Serena Release Automation 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 combination of process 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 entities in Serena Release Automation according to the Serena Release Automation documentation.

What is the Impact?

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

How Do You Change It?

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 entities in Serena Release Automation is in the *Serena Release Automation Center Installation and Administration Guide*.

Configuring Objects in Serena Business Manager

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

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

What Can You Change?

You can change the following project information:

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

What is the Impact?

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

How Do You Change It?

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

Documentation References

- Complete documentation on configuring SBM projects is in the *Serena Business Manager 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".

Chapter 6

Provider Configuration

This section tells you how to configure existing provider connections.

See the following sections for details.

Provider Configuration Overview	76
Configuring Access to Requests for Change	77
Configuring Access to Business Change Requests	78
Configuring Access to Development Change Requests	79
Configuring Access to Deployment Units	83
Specifying Providers to Use	87

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 119](#).

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

- ["Configuring Access to Requests for Change" on page 77](#)
- ["Configuring Access to Business Change Requests" on page 78](#)
- ["Configuring Access to Development Change Requests" on page 79](#)
- ["Configuring Access to Deployment Units" on page 83](#)

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 119](#).

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

- 1 ["Designating the Details for Each RFC Provider" on page 77](#)
- 2 ["Telling Release Manager Which RFC Providers to Use" on page 78](#)

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\r1m\WEB-INF\classes
```

- 2 Open one of your provider's 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 [Chapter 8, "Adding Provider Connections" on page 119](#) for details if you plan to use a provider other than Serena Service Manager for RFCs.

For example settings for the default Serena Service Manager properties files, see the following examples.

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

# itsm.table.tableName=UIM_INCIDENTS
itsm.table.tableName=TSM_CHANGEREQUEST
itsm.table.field.issueId=ISSUEID
itsm.table.field.state=STATE
itsm.table.field.releasetrain=

itsm.defaultState=Approved Changes, Approved
```



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

- [Chapter 8, "Activating the Release Manager Objects for SSM" on page 131](#)
- In the *Serena Service Manager User's Guide*, "Serena Release Manager Integration".

Telling Release Manager Which RFC Providers to Use

Once you have defined the login information for the providers in separate properties files, you tell Serena Release Manager which providers you want to use by specifying those properties file names in the providers properties file. See "[Specifying Providers to Use](#)" on [page 87](#).

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 119](#).

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

- 1 "[Designating the Details for Each BCR Provider](#)" on [page 78](#)
- 2 "[Telling Release Manager Which BCR Providers to Use](#)" on [page 79](#)

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's 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 [Chapter 8, "Adding Provider Connections" on page 119](#) for details if you plan to use a provider other than SBM for BCRs.

For example settings for the default SSM properties files, see the following examples.

- 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.ws.app.url=http://svr-dev-sbm01:80/gsoap/gsoap_ssl.dll?sbmappservices72
bcr.user=rlmadmin
bcr.password=rlmadmin_test

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

bcr.defaultState=Approved Changes, Approved
```

Telling Release Manager Which BCR Providers to Use

Once you have defined the login information for the providers in separate properties files, you tell Serena Release Manager which providers you want to use by specifying those properties file names in the providers properties file. See ["Specifying Providers to Use" on page 87](#).

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 119](#).

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

- 1 ["Designating the Details for Each DCR Provider" on page 80](#)
- 2 ["Telling Release Manager Which DCR Providers to Use" on page 82](#)

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's properties files. For example:

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

- 3 The details are unique for each properties file, and variables and parameters are defined in the implementation for the provider. See [Chapter 8, "Adding Provider Connections" on page 119](#) for details if you plan to use a provider other than Dimensions CM or SBM for DCRs.

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

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

See:

- ["Designating DCR Provider Details for Dimensions CM Requests" on page 80](#)
- ["Designating DCR Provider Details for SBM Issues" on page 81](#)
- ["Designating DCR Provider Details for SBM Incidents" on page 82](#)

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 43](#).

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 = 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.

Telling Release Manager Which DCR Providers to Use

Once you have defined the login information for the providers in separate properties files, you tell Serena Release Manager which providers you want to use by specifying those properties file names in the providers properties file. See ["Specifying Providers to Use" on page 87](#).

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 119](#).

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

- 1 ["Designating the Details for Each DU Provider" on page 83](#)
- 2 ["Specifying Providers to Use" on page 87](#)

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's 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 119](#) 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.

See:

- ["Designating Dimensions CM Deployment Unit Selection Criteria" on page 84](#)
- ["Designating ChangeMan ZMF Deployment Unit Selection Criteria" on page 84](#)

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 43](#).

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

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
<code>deploy.units.provider.name</code>	Deployment unit provider name. Default value = <code>ZMF_PACKAGES</code>
<code>deploy.units.provider.description</code>	Deployment unit provider description. Default value = ChangeMan Deployment Unit Provider for ZMF packages

Parameter	Value
FILTER_ZMF_PACKAGES_IN_DEVELOPMENT_STATUS	<p>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.</p> <p>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.</p> <p>Default value = N</p>
FILTER_ZMF_PACKAGES_BY_WORKREQNO	<p>Setting this to Y will list only packages that have an empty or null work request number value.</p> <p>Default value = Y</p> <p>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.</p>
FILTER_PLANNED_PERMANENT_ZMF_PACKAGES	<p>Include change packages with package type planned permanent. (Values Y or N)</p> <p>Default value = Y</p>
FILTER_PLANNED_TEMPORARY_ZMF_PACKAGES	<p>Include change packages with package type planned temporary. (Values Y or N)</p> <p>Default value = Y</p>
FILTER_UNPLANNED_PERMANENT_ZMF_PACKAGES	<p>Include change packages with package type unplanned permanent. (Values Y or N)</p> <p>Default value = Y</p>
FILTER_UNPLANNED_TEMPORARY_ZMF_PACKAGES	<p>Include change packages with package type unplanned temporary. (Values Y or N)</p> <p>Default value = Y</p>
FILTER_SIMPLE_ZMF_PACKAGES	<p>Include change packages with the level of simple. (Values Y or N)</p> <p>Default value = Y</p>
FILTER_PARTICIPATING_ZMF_PACKAGES	<p>Include change packages with the level of participating. (Values Y or N)</p> <p>Default value = Y</p>
FILTER_ZMF_PACKAGES_BY_PROMOTION_LEVEL	<p>Include change packages with this promotion level and above.</p> <p>The last promotion level must be <i>greater than or equal to</i> the promotion level filter.</p> <p>For example, if you have the following promotion levels in ZMF, setting this value to 10 returns change packages in these promotion levels:</p> <ul style="list-style-type: none"> 10 – INT Integration Test 20 – UAT User Acceptance Test 30 – PAT Production Acceptance Test <p>Default value = 10</p>

Parameter	Value
FILTER_ZMF_PACKAGES_BY_AUDIT_LEVEL	Include audit return code. The audit return code must be <i>less than or equal to</i> the audit level filter. Default value = 04
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
```

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

# ZMF package status filter(Y/N).
Setting this to "Y" will list all packages that are in "FRZ" and "DEV" status.
FILTER_ZMF_PACKAGES_IN_DEVELOPMENT_STATUS = N

# ZMF work request number filter (Y/N).
Setting this to "Y" will only list packages that have work request number value that is empty is set to "NULL".
FILTER_ZMF_PACKAGES_BY_WORKREQNO = Y

# zmf package type filters (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

# zmf package level filters (Y/N)
FILTER_SIMPLE_ZMF_PACKAGES = Y
FILTER_PARTICIPATING_ZMF_PACKAGES = Y

# Numeric promotion level filter.
The last promotion level return code must be greater than or equal to the promotion level filter.
FILTER_ZMF_PACKAGES_BY_PROMOTION_LEVEL = 10

# Numeric audit return code filter.
The current audit return code must be less than or equal to the audit level filter.
FILTER_ZMF_PACKAGES_BY_AUDIT_LEVEL = 04

#deploy unit and release package relationship filters (Y/N)
RELATE_DU_ITEMID_TO_ZMF_PACKAGE_WORKREQNO = Y
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 ChangeMan ZMF Communication in Release Manager"](#) on page 49.

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.

Specifying Providers to Use

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

You can select one or more providers for 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 83.

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

For details on configuring the corresponding properties files, see the following:

- ["Configuring Access to Requests for Change" on page 77](#)
- ["Configuring Access to Business Change Requests" on page 78](#)
- ["Configuring Access to Development Change Requests" on page 79](#)
- ["Configuring Access to Deployment Units" on page 83](#)

Chapter 7

Serena Release Manager Upgrade

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

[Upgrading from Serena Release Manager v1.1 to v2.0](#)

90

Upgrading from Serena Release Manager v1.1 to v2.0

If you need to upgrade Serena Release Manager from version 1.0.0.01 to version 1.1, please see the documentation for Serena Release Manager version 1.1 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 15](#).

You can upgrade Serena Release Manager from version 1.1 to version 2.0 on SBM 2009 R2-R4 using the following steps.



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

Step	Description	Actions
1	Back up files in target directories	<p>(Optional) It is a good practice to snapshot your system or back up existing files and data before beginning the upgrade.</p> <p>At minimum, you should back up as follows:</p> <ol style="list-style-type: none"> 1 Copy the Serena Common Web services rlm folder, for example C:\Program Files\Serena\common\tomcat\6.0\webapps\rlm, to a temporary folder. 2 Copy the Serena\Solutions\Release Control folder, for example C:\Program Files\Serena\Solutions\Release Control, to a temporary folder. 3 Copy the files from the Serena Common Web services rlm\WEB-INF\classes folder, for example C:\Program Files\Serena\common\tomcat\6.0\webapps\rlm, to a temporary folder.
2	Copy the upgrade package to the Release Control directory	<ol style="list-style-type: none"> 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 RC3.2-bbb-Win32-Upgrade.zip, to a temporary directory. 3 Extract the upgrade package to the Serena\Solutions\Release Control folder. 4 The following files should now appear under the Release Control folder: <ul style="list-style-type: none"> ■ com.serena.rlm.sbm.shell.zip ■ solution file <ul style="list-style-type: none"> • RLM_Solution_Pack-2.0.m.bbb.sln, where m is maintenance release number bbb is build number ■ war files <ul style="list-style-type: none"> • rlm.war • zmf.war • zmfalf.war • zmfws.war ■ zmfalf_resource.properties

Step	Description	Actions
3	Configure the Web services files in the Serena Common Tomcat Web server	<ol style="list-style-type: none"> 1 Select Start Administrative Tools Services and stop the Serena Common Tomcat service. 2 Navigate to the Serena Common Tomcat webapps folder. For example: <pre>C:\Program Files\Serena\common\tomcat\6.0\webapps</pre> 3 Delete the rlm folder. 4 Delete the rlm.war file. 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 directory. For example: <pre>C:\Program Files\Serena\common\tomcat\6.0\webapps</pre> 6 Restart the Serena Common Tomcat service. This war file contents are extracted to new directories in that location. You should now see the following directories under webapps: <pre>rlm zmf zmfws zmfalf</pre> 7 From the Release Control folder where you extracted it in a preceding step, copy the alfzmf_resource.properties file to the Serena Common Tomcat webapps\zmfws\WEB-INF directory. For example: <pre>C:\Program Files\Serena\common\tomcat\6.0\webapps\zmfws\WEB-INF</pre>
4	Install the solution file that contains the Serena Release Manager process apps and all related orchestrations, reports, and tables	<ol style="list-style-type: none"> 1 From the Release Control folder, copy the solution pack .sln file, such as rlm_solution_pack_2_0.0.sln, to the SBM WEB-INF\solutions folder. For example: <pre>C:\Program Files\Serena\SBM\Common\jboss405\server\default\deploy\mashupmgr.war\WEB-INF\solutions</pre>
5	Import the Release Manager solution	<ol style="list-style-type: none"> 1 Log into the SBM Application Repository as the SBM administrative user. 2 Navigate to the Solutions tab and import the solution. For example: <pre>RLM_Solution_Pack 2.0.m.bbb,</pre> <p>where m is maintenance release number and bbb is build number.</p> 3 See Chapter 4, "Importing the Serena Release Control Solution" on page 34.

Step	Description	Actions
6	Promote the snapshots	<p>1 Navigate to the Process App Snapshots tab.</p> <p>2 Promote the snapshots in the following order. When promoting the snapshots, make sure to select the endpoints as needed, and make sure the endpoints are authenticated with Single Sign-on.</p> <ul style="list-style-type: none"> ■ Release Train ■ Application Release ■ Release Package ■ RLM_AUX ■ Deployment ■ Environment ■ ReleaseTemplate <p>3 Promote the following snapshots again to enable all fields to be created for two-way dependencies between process apps.</p> <ul style="list-style-type: none"> ■ Release Train ■ Application Release ■ Release Package ■ RLM_AUX ■ Deployment <p>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 140.</p> <p>For detailed instructions see Chapter 4, "Promoting the Snapshots" on page 36.</p>
7	Deploy the process apps	<p>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.</p> <p>To redeploy the process apps:</p> <ul style="list-style-type: none"> 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. <p>NOTE Follow the guidelines in the SBM documentation to deploy the Serena Release Manager process apps.</p>

Step	Description	Actions
8	If you are changing the port on which the Serena Common Tomcat runs, reconfigure Serena Release Manager to use the new port number	<p>1 If you are changing the port on which the Serena Common Tomcat runs, follow the instructions in Chapter 8, "Configuring Release Manager to Use a Different Port" on page 124.</p>
9	Install the shell user interface and report templates	<p>1 Extract the <code>com_serena_rlm_sbm_shell.zip</code> file directly to the SBM Application Engine folder. For example:</p> <pre>C:\Program Files\Serena\SBM\Application Engine\</pre> <p>The files in the zip file should extract to the appropriate directory structure.</p> <p>Verify the extraction by looking at the dates of the files in the <code>template\shell\rlm</code> folder, for example <code>C:\Program Files\Serena\SBM\Application Engine\template\shell\rlm</code>. The files should have the date close to that of the upgrade package you used.</p> <p>2 From SBM System Administrator, select File Put Files Into Database. Confirm when prompted.</p> <p>This puts the UI shell files into the SBM database.</p>
10	Set privileges for the administrative user to the Serena Release Control objects, such as projects, reports, and tables	<p>1 Set privileges for the new deployment process template objects and verify privileges are set for all other Serena Release Control objects.</p> <p>See Chapter 4, "Configuring the Administrative User Privileges" on page 40.</p>
11	Enable roles for Serena Release Control projects and verify that Serena Release Manager is activated	<p>1 Enable the roles in the new Release Template project and verify that roles are enabled for all other Serena Release Control projects.</p> <p>See Chapter 4, "Enabling Serena Release Control Project Roles" on page 41.</p> <p>2 Verify that Serena Release Manager is activated by entering the URL in your Web browser. For example:</p> <pre>http://rlmhost/tmtrack/tmtrack.dll?shell=rlm</pre>
12	Configure the dashboard if needed	<p>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 42.</p>

Step	Description	Actions
13	Update the registry to ensure the proper version of Serena Release Control is registered for future reference	<p>(Optional)</p> <ol style="list-style-type: none"> 1 Manually update the key as follows: <ol style="list-style-type: none"> a Edit the registry with a program such as Regedit. b View the following key: <pre style="margin-left: 20px;">HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{F711E0DD-CAE6-4ADE-8CAA-8F54BB92214F}</pre> c Modify DisplayVersion by changing the Value data entry to 3.2.0. <p>NOTE This registry location has all the information that you will see in Windows Add/Remove programs and Programs and Features.</p>
14	Update the configuration of the system, application objects, and providers as needed	<ol style="list-style-type: none"> 1 Configure the communication and application objects as needed. See Chapter 4, "System Activation and Configuration" on page 31, Chapter 5, "Configuration and Administration of the Integrating Objects" on page 57, and Chapter 6, "Provider Configuration" on page 75. <p>Changes for this release include the following:</p> <ul style="list-style-type: none"> ■ There are several new properties files to support the ZMF integration. ■ The properties file for Serena Release Automation is now called <code>nolio-client.properties</code>. <code>nolio.properties</code> is no longer used.

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.

Customizing the User Interface	98
Customizing Release Control Workflows	105
Modifying Release Types and Stages	106
Adding Provider Connections	119
Configuring Release Manager to Use a Different Port	124
Customizing the Serena Release Control Custom Shell	128
Activating Environment Association to Release Packages	131
Activating the Release Manager Objects for SSM	131
Extending the Calendar	133

Customizing 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.

User interface configuration topics are as follows:

- "Customizing the Serena Release Control Dashboard" on page 98
- "Configuring the Calendar" on page 100
- "Configuring the Inbox" on page 101
- "Configuring Views and Dialog Boxes" on page 103



NOTE This section includes typical user interface configuration. This does not include customization of the UI shell to include new form elements. Some UI customization is required when you add or modify stages. This customization is documented in context in Chapter 8, "Serena Release Manager Customization" on page 97.

Customizing the Serena Release Control Dashboard

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

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

What Can You Change?

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

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

What is the Impact?

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

How Do You Change It?

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

Selecting the Report for the Dashboard Page

Upon initial install or upgrade, 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 the dashboard as follows.

- 1 In SBM System Administrator, select the **Users** tab.
- 2 Select your Login ID and click **Edit**.
- 3 Select the **Preferences** tab and deselect the **Show Launch Page** field.
- 4 Select the **Settings** tab.
 - a In the **Application** list, select **Release Train** and then click **Set Preferred Application**.
 - b In the Home Page Report field, select **Base Project: Dashboard**.
- 5 Click **OK**. 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 modify 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:

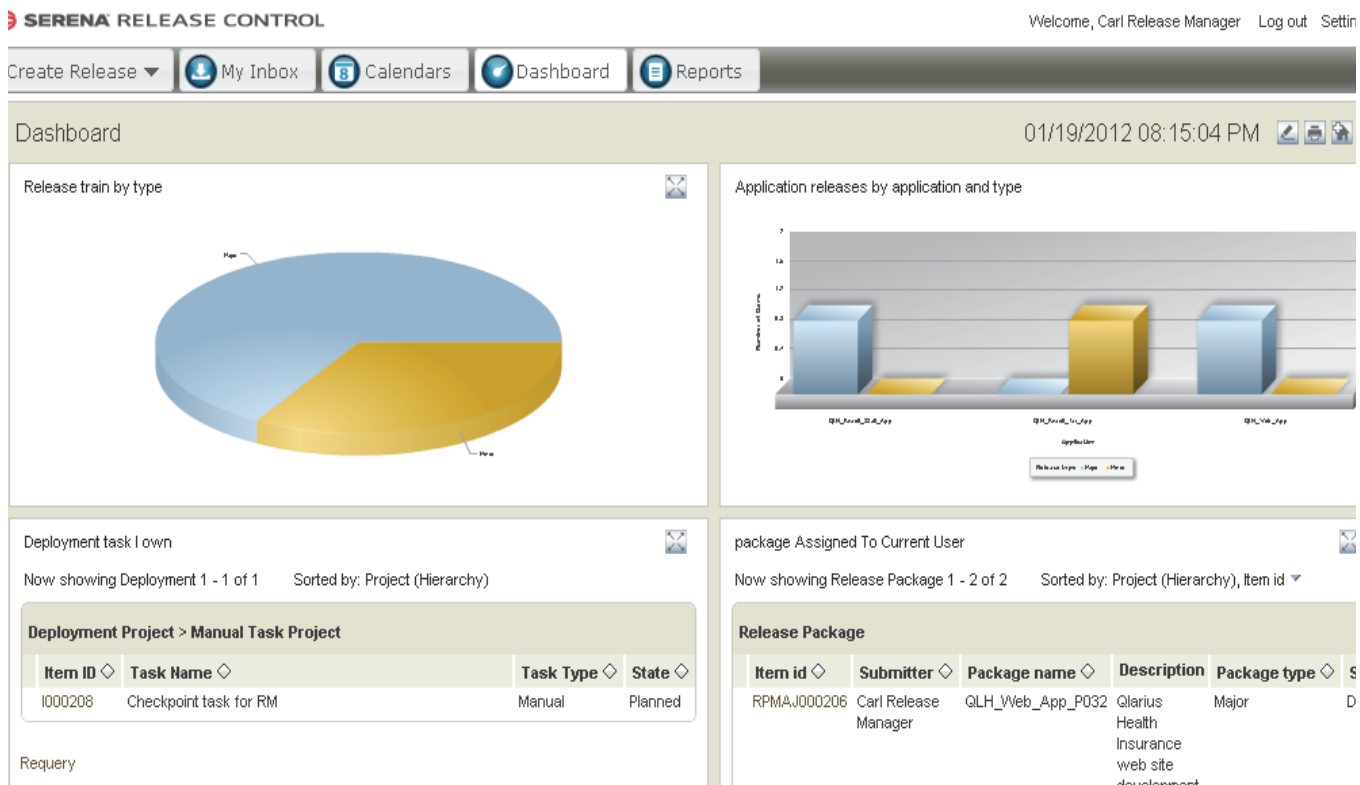
- 1 From the Serena Release Control standard SBM user interface, click the **RTrain** tab.
- 2 Expand the **Reports** section of the navigation pane and click **Browse Multi-View Reports**.
- 3 Select the **Dashboard** report from the list of reports.
- 4 Click **Edit report** and configure the Dashboard 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**, **Release packages list by owner**, and **Deployment tasks I own** is shown in the following figure.



The screenshot shows the Serena Release Control dashboard interface. At the top, there is a navigation bar with 'Create Release', 'My Inbox', 'Calendars', 'Dashboard', and 'Reports'. The main content area is divided into four sections:

- Release train by type:** A 3D pie chart showing two segments, one blue and one yellow.
- Application releases by application and type:** A 3D bar chart with three bars of different heights and colors (blue, yellow, blue).
- Deployment task I own:** A table showing one deployment task.

Item ID	Task Name	Task Type	State
I000208	Checkpoint task for RM	Manual	Planned
- package Assigned To Current User:** A table showing one release package.

Item id	Submitter	Package name	Description	Package type	S
RPMAJ000206	Carl Release Manager	QLH_Web_App_P032	Clarius Health Insurance web site development	Major	D

Configuring the Calendar

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

What Can You Change?

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

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

What is the Impact?

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

How Do You Change It?

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

To change the calendar report:

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

- 2 Select **Reports** from the navigation pane and search for reports with Gantt in the title.
- 3 Under **Release Train New Project (Release Train)**, select **Release Trains in Gantt** and then click **Modify Listing** for that report.
- 4 Expand **Additional Options** and look at the **Optional HTML Template** field.
This lists the HTML page for the selected report. The default report is:
ganttview.htm
- 5 Modify the report selections and the corresponding HTML page as needed.



NOTE The **Calendars** page appears empty unless at least one release train exists to display in the ganttview.htm report.

Documentation References

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

Configuring the Inbox

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

What Can You Change?

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

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



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

What is the Impact?

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

How Do You Change It?

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

To change reports for the inbox:

- 1 From the Serena Release Control standard SBM user interface, select the process app for which you want to change a report, such as **RTrain**.

- 2 From the **Reports** section of the navigation pane, search for all reports.
- 3 Select the report you want to change and then click **Save**.
- 4 Overtyping the existing name with a new name, making sure that you do not change the report ID.
- 5 Repeat for all Serena Release Control process apps for which you want to change **My Inbox** page reports.

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

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



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

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

Documentation References

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

Configuring Views and Dialog Boxes

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

What Can You Change?

You can change the views and dialog boxes as follows:

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



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

What is the Impact?

Possible impacts for changing dialog boxes are as follows:

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

How Do You Change It?

To change information displayed in a selection field:

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

To change information displayed in a table:

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

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

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

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

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

- 2 Verify that the updated report information shows properly in the associated UI.
- 3 To show or hide columns of information returned from the report, click the down-arrow beside the last column heading and then select or deselect column names in the list.

Documentation References

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

Customizing Release Control Workflows

Before the people who participate in the release management processes in your organization begin using Serena Release 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?

- You can change the workflows
- You can change the 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 106](#).

What is the Impact?

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

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

How Do You Change It?

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

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

Documentation References

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

Modifying Release Types and Stages

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

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

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

What Can You Change?

You can change the following release type and stage information:

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

What is the Impact?

- The release types and stages are interrelated, and if you change one aspect, you need to change all related information in both release train and release package primary and auxiliary tables, forms, workflows, and UI Javascript.
- You must be careful when changing the Release Package system field definitions because they are used by Serena Release 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 there is failure in the release package deployment process. The default Release package workflow value is set to Fail Deployment.
- You must be careful when changing the Release Package process app not to adversely affect the implementation of the Deployment Task execution. Deployment Task field dependencies are as follows:
 - Deployment Task Status Single Selection control field:

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

To indicate when a deployment task can be executed for deployment, the task status value must be set to Planned. This is currently set in the Create transition.

- The automation transition actions associated to the Task Update transition are mapped to the value set in the Task Status. Currently the default actions are set as follows:

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

How Do You Change It?

You add or change most stage information in SBM Composer.

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

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

Summary of Adding a Stage

Adding a Stage for the Release Train Process App

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

Adding the Stage for the Release Package Process App

- 1 Add the new stage name value in the primary table Release package.
- 2 Create a new swim lane for the new stage process.
- 3 Create required states for the new stage process.
- 4 Add From and To transitions with all associated forms, mappings, and overrides for the new states.
- 5 Map a new deployment transition action.
- 6 Add a new re-deploy transition action.
- 7 Configure the Stage auxiliary table to relate the new stage to the Release Train Start and End Dates and the Release Package Deploy state.

- 8 Optionally add a new deploy decision rule to the Release Package stage process.
 - a Add the decision to the workflow.
 - b Add a new **Package type** single-selection field value.
 - c Add a rule for the selection, **Package type** in value.
 - d Add a new transition for the deploy decision.
 - e Add the new rule to the deploy decision box.

Example of Adding a Stage

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

Adding a Stage for the Release Train Process App

To add a stage for the Release Train process app:

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

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

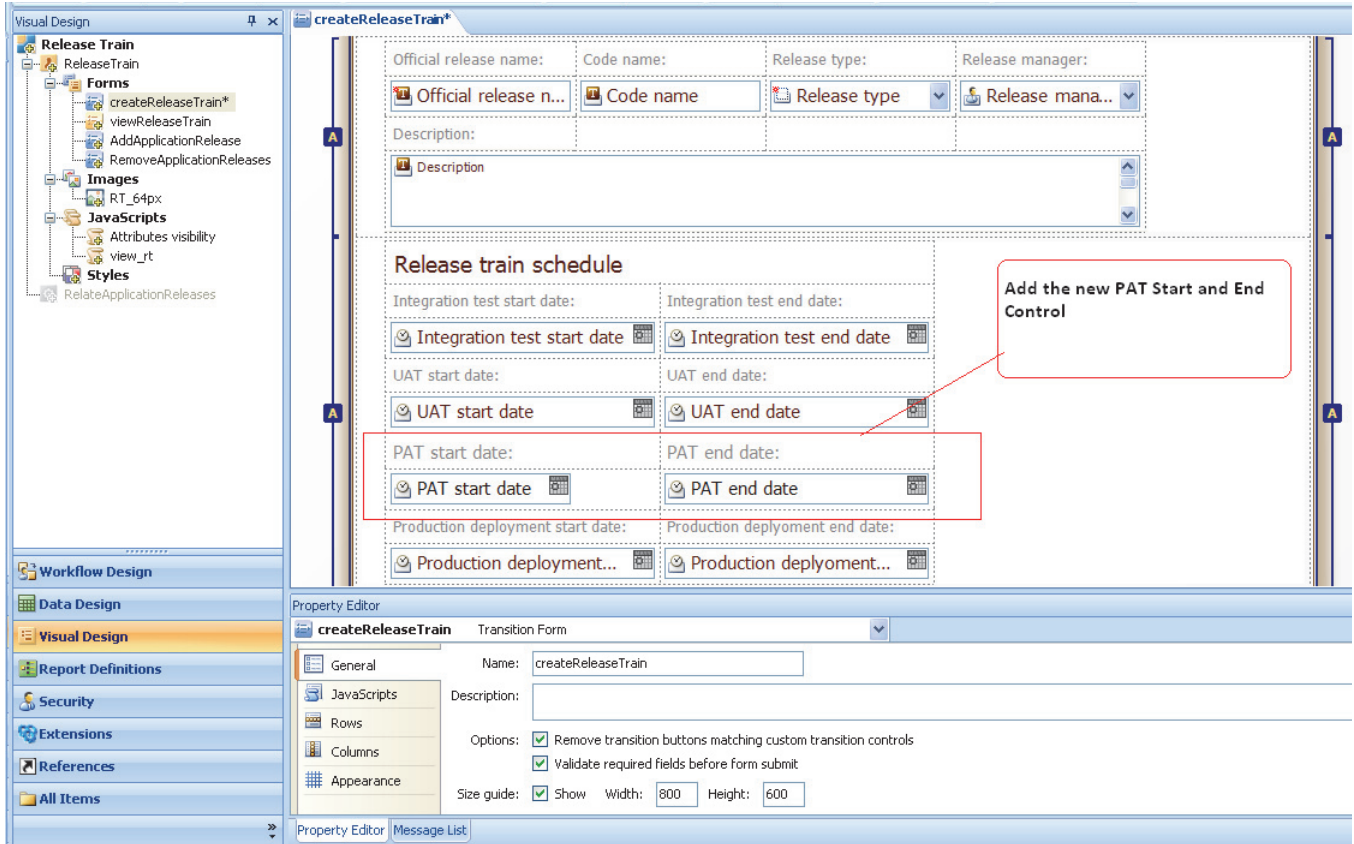
The screenshot displays the Data Design tool interface. On the left, a tree view shows the 'Release Train' project structure. The main window shows the 'Release Train (Primary)' table structure. The table has the following fields:

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 deployment 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

The 'PAT end date' field is highlighted in blue. A red box highlights the 'Production Acceptance Testing' label in the Property Editor for this field. The Property Editor shows the following details for 'PAT end date':

- Name: PAT end date
- Database field name: PAT_ENDDATE
- Type: Date/Time

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



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

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

```

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

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

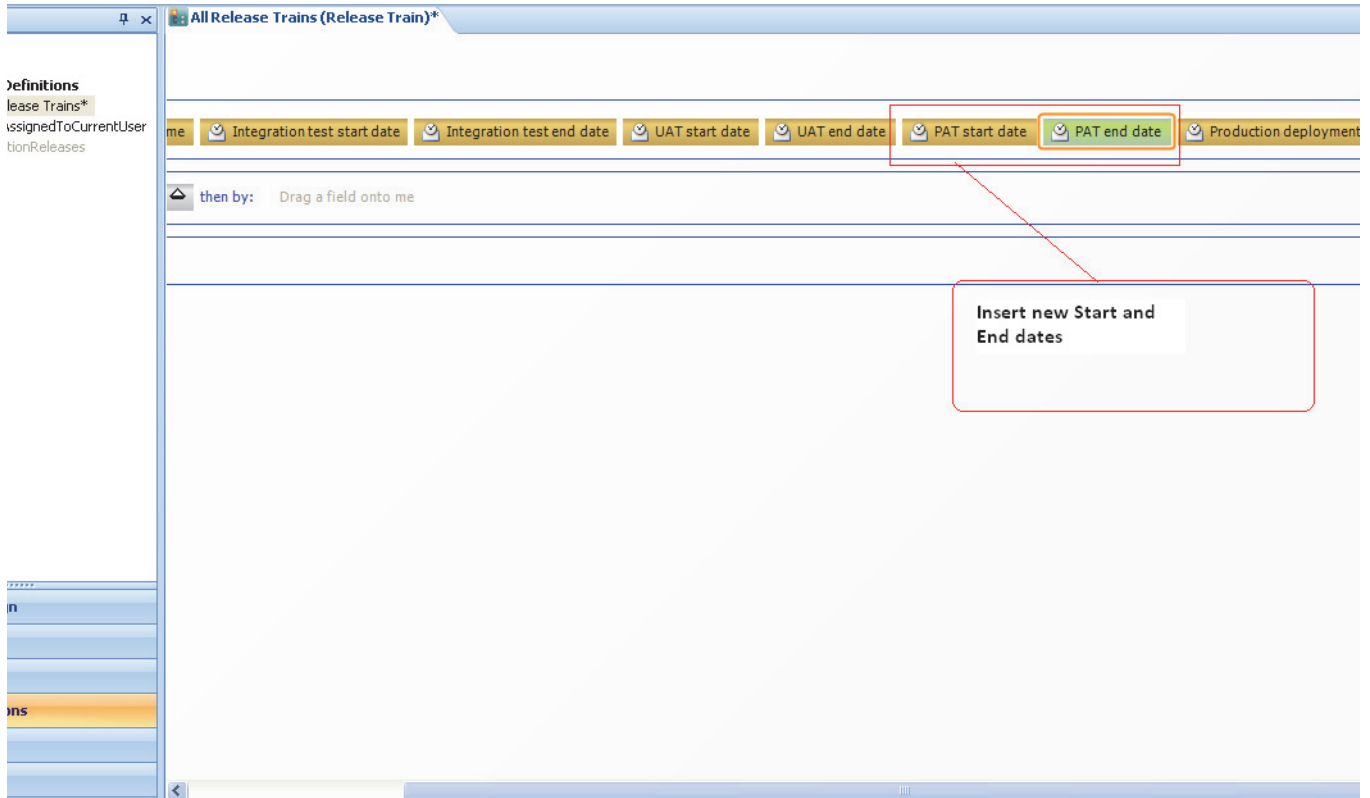
    whatToHide(type, fields);
    hideNeeded(fields);
  }
);
AddChangeCallback("ISSUETYPE",
  function() {
    var type = GetFieldValue("ISSUETYPE");
    var fields = {};

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

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

```

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



Adding the Stage for the Release Package Process App

To add the stage for the Release Package process app:

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

Release Package (Primary Table)
[No description]

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

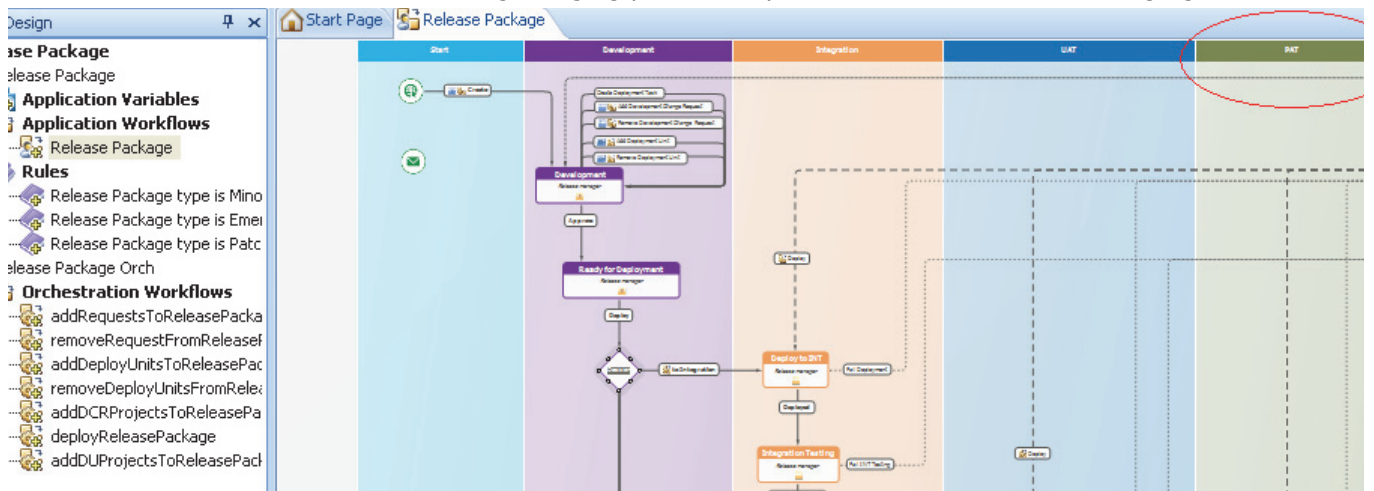
Property Editor
Deploy state: Single Selection Field

Style: Allow searching, Single drop-down list

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 val.

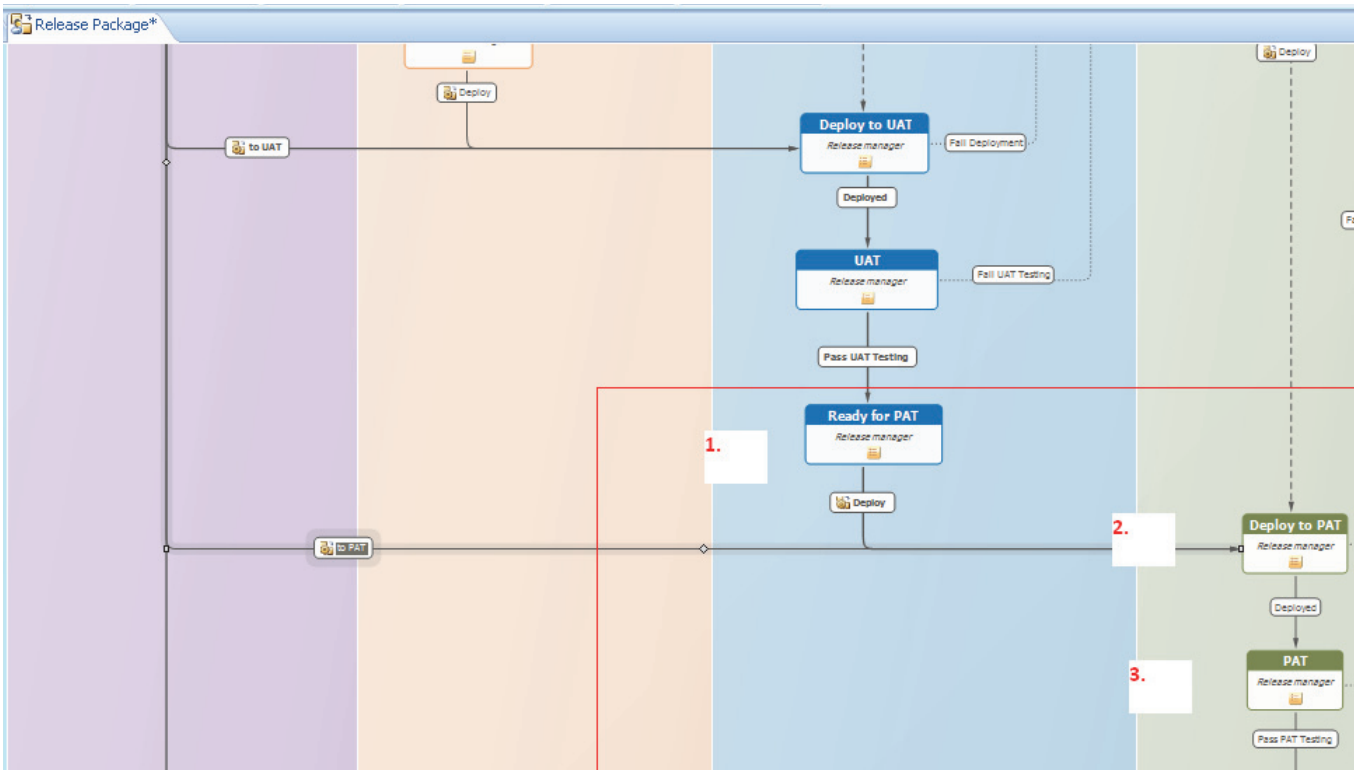
- 2 Create a new swim lane for the new stage process and arrange it according to the Release Package staging process sequence as shown in the following figure.



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

- Ready for PAT
- Deploy to PAT
- PAT

The added states are shown in the following figure.



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

State	Transition	Option	Selection
Ready for PAT	From Transition: Pass UAT Testing		
	To Transition: Deploy	Options	Quick transition
		Form	None
		Field Privileges	default values
		Field Overrides	Failed Deploy Transition <ul style="list-style-type: none"> ■ 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
	To transition: Fail Deployment	Restrict by Role	default values
		Options	Quick transition
		Form	None
		Field Privileges	default values
		Field Overrides	Deploy state <ul style="list-style-type: none"> ■ 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
	To transition: Fail PAT Testing	Restrict by Role	default values
		Options	Quick transition
		Form	None
		Field Privileges	default values
		Field Overrides	None
		Actions	None
		Restrict by Type	default values
Restrict by Role	default values		

- 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.

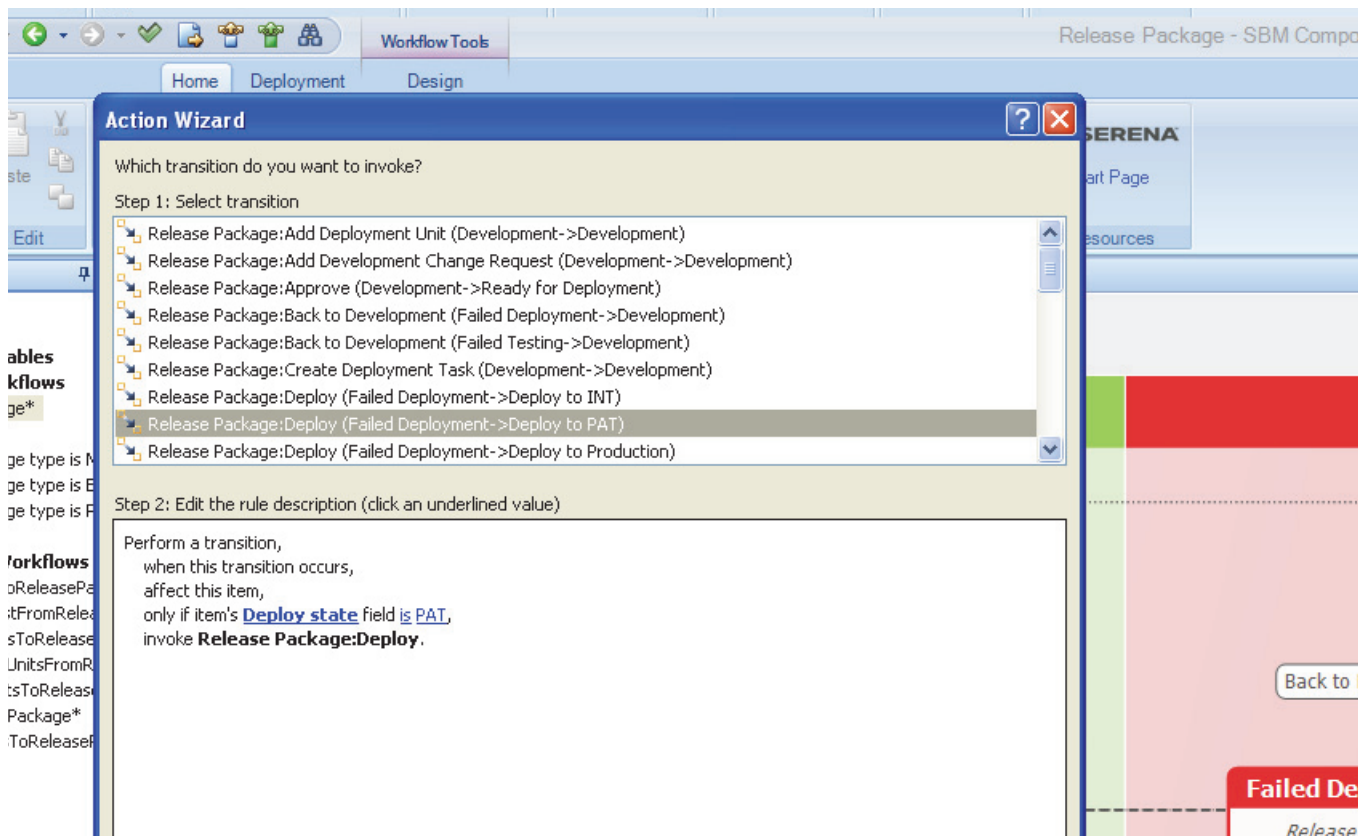
State	Transition	Option	Selection
From state: Failed Deployment To state: Deploy to PAT	Deploy	Options	Quick transition and Hide button on form
		Form	None
		Field Privileges	default values
		Field Overrides	Failed Deploy Transition <ul style="list-style-type: none"> ■ Read Only ■ Set to default: Fail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

- Add a new re-deploy transition action.

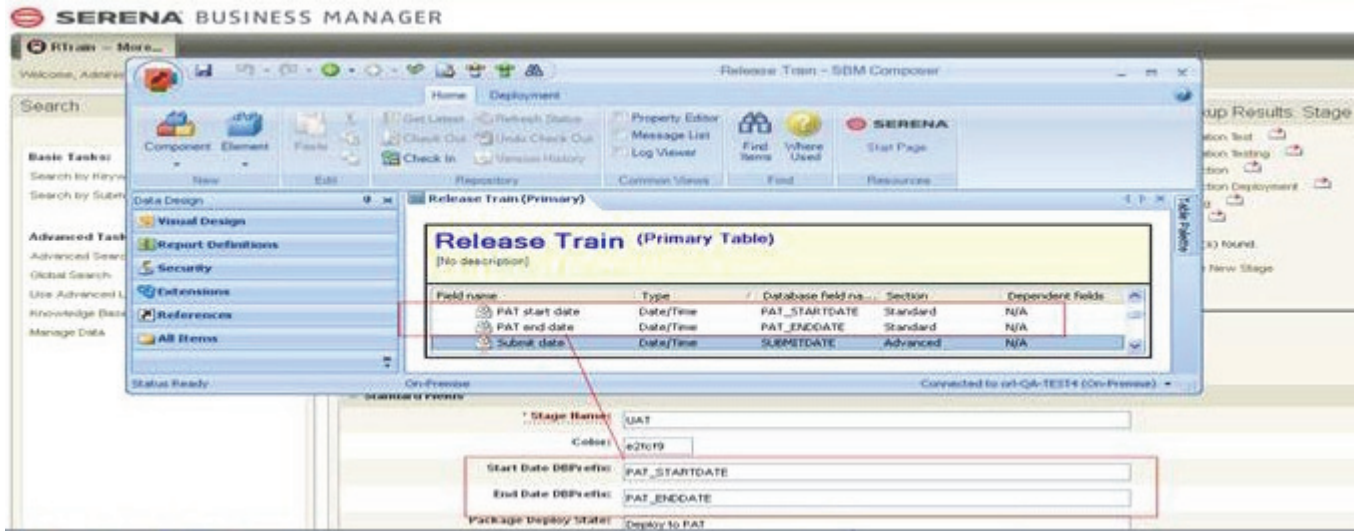
Set the rule as follows:

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.



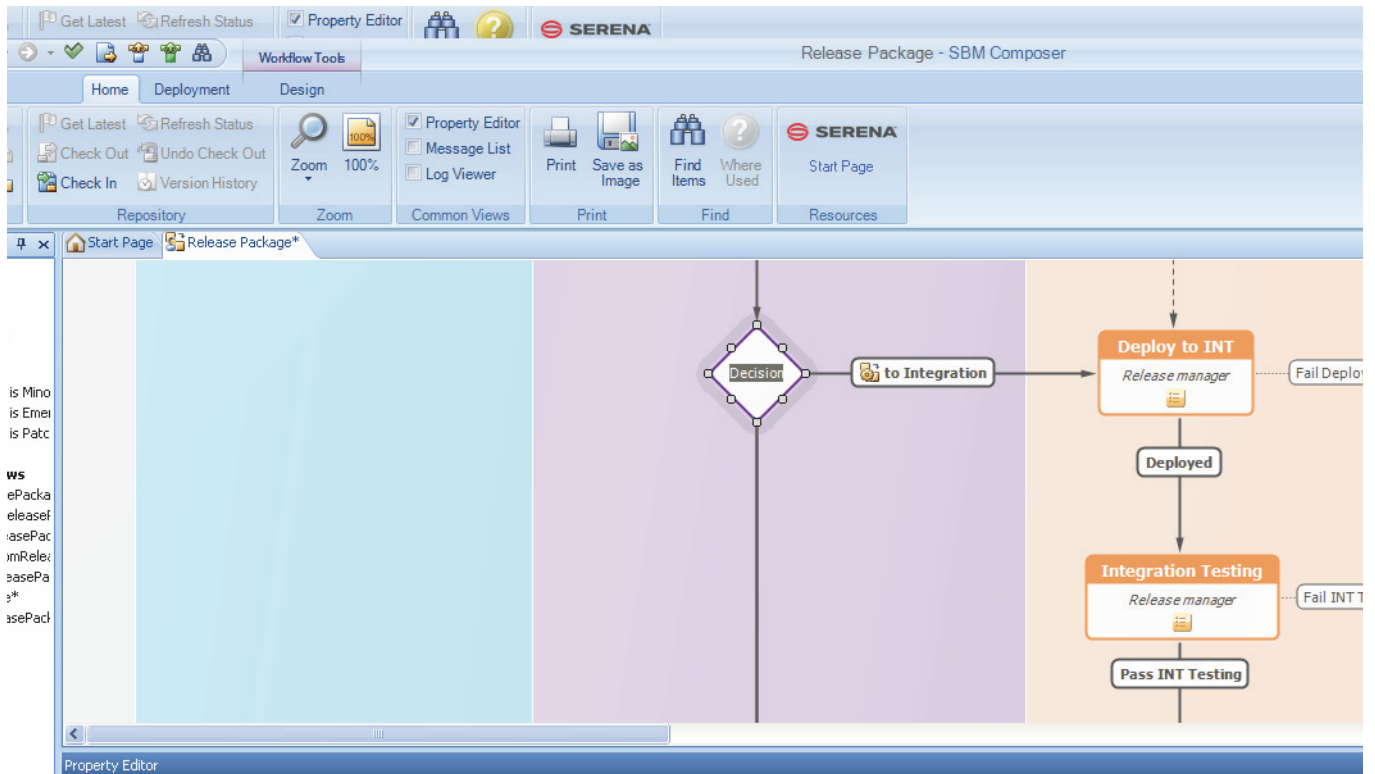
- 7 Configure the Stage auxiliary table to relate the new stage to the Release Train Start and End Dates and the Release Package Deploy state.
 - Relate the PAT Start and End date fields in the Release Train table. This relationship is indicated in the following figure.



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

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

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



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

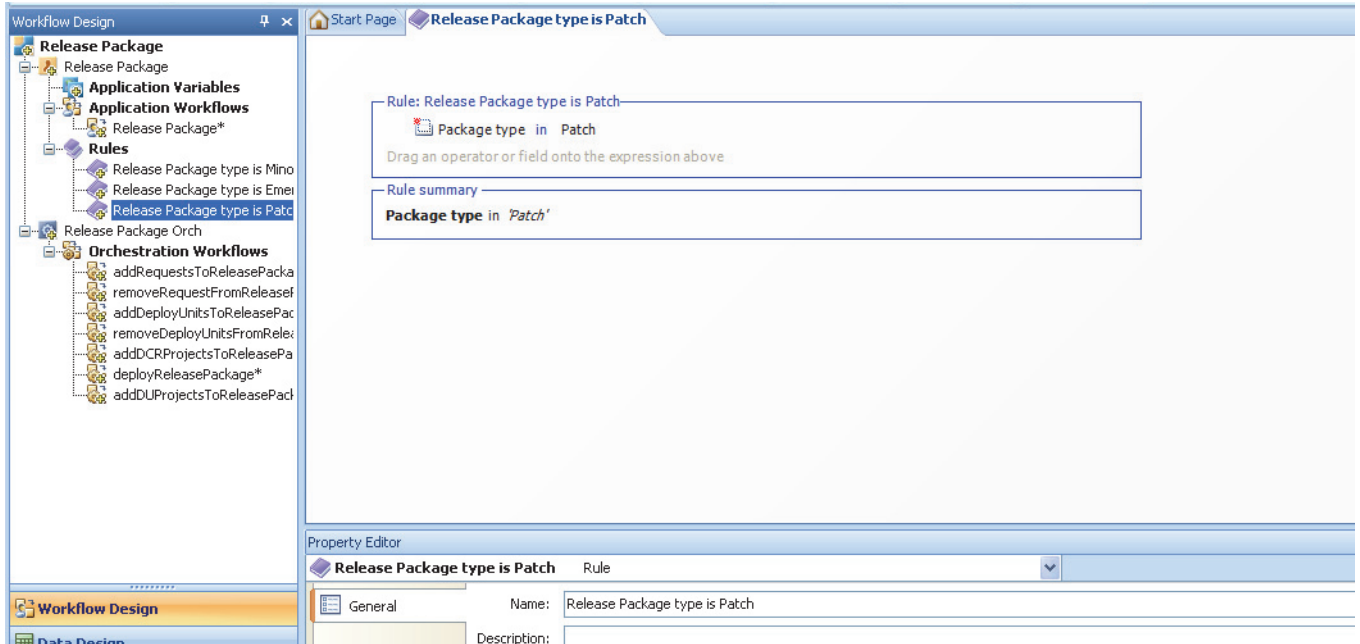
The screenshot shows the 'Release Package (Primary Table)' configuration window. The table below lists the fields:

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	

Below the table is the 'Property Editor' for the 'Package type' field, which is a 'Single Selection Field'. The 'Options' section is checked, and the 'Values' table is shown:

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

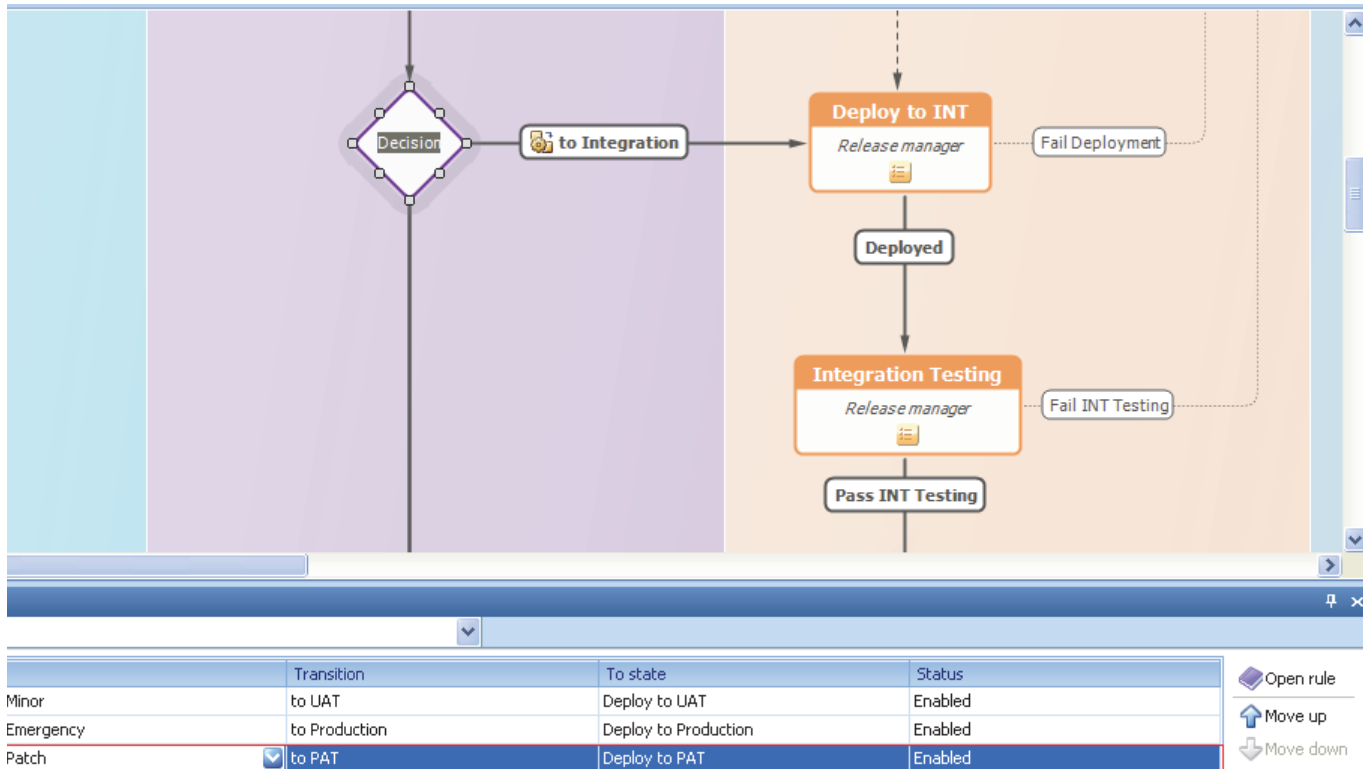
- c Add a rule for the selection, **Package type** in Patch, as shown in the following figure.



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

To and From	Transition	Option	Selection
From decision box: Deploy To state: Deploy to PAT	to PAT	Options	Quick transition and Hide button on form
		Form	None
		Field Privileges	default values
		Field Overrides	Failed Deploy Transition <ul style="list-style-type: none"> ■ 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.



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

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

- ["Creating a Class for Your Provider" on page 120](#)
- ["Creating Properties Files for Your Providers" on page 120](#)

- "Building and Packaging" on page 123
- "Telling Serena Release Manager to Use This Provider" on page 123



NOTE For documentation on configuring existing provider connections, see [Chapter 6, "Provider Configuration"](#) on page 75.

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 121.
- Define all instance-specific values for parameters in a properties file. See ["Telling Serena Release Manager to Use This Provider"](#) on page 122.

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, usage of a properties file is a good practice and is included in the example provided. Using a properties file allows you to define several possible configurations so you can easily change details without code modification. Without a properties file, you must hard code name, description, and other specific parameters for your provider.

Examples

fs_example.properties

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

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

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

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

requests.txt

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

depunits.txt

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

stages.txt

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

areas.txt

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

Building and Packaging

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

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

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

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

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

```
provider_fs.zip
```

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

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

Telling Serena Release 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.

If you want to use a different port for Serena Release Manager, you must change the port number as follows:

- 1 Install the Serena Common Tomcat Web server and Serena Release Control as usual.
- 2 Rerun the Serena Release Control installer at the command line with parameters to change the port the Serena Common Tomcat Web server runs on.
- 3 Configure the endpoints for the snapshots to point to the non-default port, and then promote the snapshots and deploy the process apps.
- 4 Configure all RESTgrid widgets in the process app forms to point to the non-default port and then redeploy the process apps.
- 5 Change the port to which the Web Services WSDLs point and re-import the WSDLs.

See:

- [Chapter 3, "Installing Serena Release Control" on page 26](#)
- ["Changing the Port on Which the Common Web Server Runs" on page 125](#)
- [Chapter 4, "Promoting the Snapshots" on page 36](#)
- ["Configuring a Non-Default Web Server Port in the Process Apps" on page 126](#)
- ["Changing the Web Services to Point to a Different Port" on page 128](#)



CAUTION! 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.



NOTE 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.

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\RBuild\Win64
```

- 3 Enter the following command at the prompt:

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

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

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

The Serena Release Control installer appears.

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



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

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

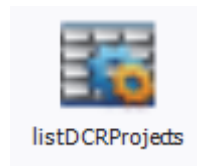
If you want to run the Serena Common Web server on a port other than 9095, you need to modify the port numbers in the Serena Release Manager process apps that have RESTgrid widgets. After you modify the process apps, you need to redeploy them.

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 modify and redeploy the process apps:

- 1 Open the process app you want to change in SBM Composer. For example, open Release Package from the Application Repository.
- 2 Display the **Visual Design** view.
- 3 Under **Forms**, select one of the forms. For example, select createRelPackage. For a list of forms that you should update, see ["Forms with RESTgrid Widgets" on page 127](#).
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.
- 9 When you are finished, redeploy the process apps as follows:
 - 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.

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.



NOTE Follow the guidelines in the SBM documentation to deploy the Serena Release Manager process apps.

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

Process App	Forms	Controls
Release Train	addRFC	listRfcRequests
Deployment	newAutomationTask, newAutomationTemplate, editAutomationTask, editAutomationTemplate	gridApps, gridEnv, gridProc
	newVaultDimDeployTask, editVaultDimDeployTask	gridDepUnits, gridDepStages, gridDepAreas
	newVaultZMFDeployTask, editVaultZMFDeployTask	gridDepUnits, gridDeploymentSites, gridDeploymentAreas
	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.

Customizing the Serena Release Control Custom Shell

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 application.

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 modify 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 modify 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.

Activating the Release Manager Objects for SSM

Although you can associate SSM Changes with Serena Release Manager RFCs in the default implementation, the Serena Release Manager reports that are referenced in the default SSM user interface are not yet configured in Serena Release Manager, and the events that send information back to SSM to transition the related Changes are not in place. To activate the connection, you must configure these reports and events.

- ["How SSM Detects Whether Serena Release Manager is Installed" on page 132](#)

- "Configuring Reports that SSM References" on page 132
- "Configuring Events to Transition SSM Changes" on page 132

How SSM Detects Whether Serena Release Manager is Installed

In the default implementation, SSM detects that Serena Release Manager is installed if both are process apps in the same instance of SBM.

Configuring Reports that SSM References

The Serena Release Manager reports that are referenced in the default SSM user interface are not yet configured in Serena Release Manager. To display the information you expect to see in SSM, you must create the reports with the information that is relevant for your organization.

To create the 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. Use query parameters for the date range start and end dates, using the value Query at Runtime.

For example, create reports with reference names AllTrains-Planned and Suitable_Trains. Use basic conditions with selections of Production deployment end date = (Query At Runtime) and Production deployment start date = (Query At Runtime).

Configuring Events to Transition SSM Changes

To make sure the SSM Change transitions along with the associated RFC, you must create events in Serena Release Manager.

To configure the events to transition SSM Changes:

- 1 When the release train goes from **Review** to **In Progress**, you should transition the associated RFC to **Implemented** using the **Assigned to RLM** transition.
- 2 When the release train goes from **In Progress** to **Completed**, you should transition the RFC from **Implemented** to **Post Implementation** using the **Implemented via RLM** transition.
- 3 When we do an association in Serena Release Manager by adding an RFC to a release train, data should be updated in our RLMAux table and information posted to the SSM system so that the associated SSM Change gets updated with the Serena Release Manager information.

Extending the Calendar

The default calendar reports shows the essential information about release trains, including the release stages and their schedule windows. If you want to extend your calendar to be a richer IT calendar, please contact [Serena Support](#) or check the knowledgebase for information on how to get the extension files.

Chapter 9

Troubleshooting

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

Troubleshooting Overview	136
Information from the Serena Release Control User Interface	136
Information from the Product Log File	137
Symptoms and Solutions	137

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 stored in the Serena Release Manager product log file, `r1m.log`.
- Troubleshooting information for integrating products, such as SBM, Dimensions CM, ChangeMan ZMF, and Serena Release Automation.

Information from the Serena Release Control User Interface

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

Error Messages

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

If an error message is returned from one of the integrating products, it may not direct you to the exact solution to the problem. See ["Symptoms and Solutions" on page 137](#) for more assistance in these situations.

Activity Logs

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

For deployment task failures, see ["Information from the Product Log File" on page 137](#) for more troubleshooting techniques.

History

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

Information from the Product Log File

The product log file is `r1m.log`. Logging is done through the common Tomcat Web server using the Apache `log4j` Java-based logging utility.

The `r1m.log` file contains status messages returned to the AFS server. The level of messaging depends on the settings in the `log4j` settings. Debugging and `append` are on by default.

The default location of the `r1m.log` file is the catalina home directory, set by the variable `${catalina.home}`. For example:

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

To change the location of the log files:

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

```
C:\Program Files\Serena\common\tomcat\6.0\webapps\r1m\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>\r1m.log`

Symptoms and Solutions

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

- ["Unexpected Display Results" on page 137](#)
- ["Matches Not Found for Selections" on page 139](#)
- ["Release Package Deployment Fails" on page 139](#)
- ["Installer Errors" on page 140](#)
- ["Slow Response Time" on page 141](#)
- ["Release Package Deployment Fails" on page 142](#)
- ["Create Release Fails" on page 142](#)
- ["Dimensions CM DCR Projects Appear in the Wrong Section" on page 143](#)

Unexpected Display Results

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

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

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

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

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

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

The Serena Common Tools page appears.

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

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

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

Services that should be active include:

- DeploymentAutomationService
- RLMUtilService
- ReleaseRequestService

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

Serena Release Control data is not displayed as expected

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

See [Chapter 4, "Configuring Required Objects in Serena Release Control"](#) on page 39.

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

The New Color Scheme Doesn't Appear in the UI

If the user interface shell has a mix of the old and new color schemes and icons after an upgrade, new installation, or customization, you may need to clear your browser cache. This occurs because certain UI elements are stored in the browser cache for faster refresh time.

To clear the cache, choose the option in your browser to delete history and select cache from the options given.

You should also refresh the template cache by entering the following URL in your browser:

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

Most But Not All of the UI Elements Appear

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.

See [Chapter 4, "Putting the UI Shell files into the SBM Database"](#) on page 39.

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 for a release package project selection

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

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

```
DimClientException ... Error: Not an authorized user
```

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

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

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

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

Release Package Deployment Fails

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

A deployment vault task fails when you deploy a release package

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

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

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

- The same user ID and password must be used in both SBM and Dimensions CM.
- SSO must be enabled for both SBM and Dimensions CM on the same SSO server.

- The user ID must be given a role in the Serena Release Control projects, such as Release Packages and Deployment Tasks.

An automation deployment task fails when you deploy a release package

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

Installer Errors

If the installer fails, here are some possible solutions.

Common Tools files are missing from the install

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

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

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

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

Snapshot Promotion Errors

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

The 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 environment target servers and verify that they are set up properly as documented in [Chapter 4, "Creating an Environment for Serena Release Manager"](#) on

page 35.

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. See [Chapter 4, "Promoting the Snapshots"](#) on page 36.

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.

Release Package Deployment Fails

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

A deployment vault task fails when you deploy a release package

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

If you are sure the task works in native Dimensions CM, check the **Activity Log** tab for information. If that does not give enough information for you to identify the problem, check the details in the `r1m.log` file.

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

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

An automation deployment task fails when you deploy a release package

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

Create Release Fails

If you attempt to create a release train, application release, or release package and the operation fails with the following error:

```
Error occurred during web service invocation:  
SOAP Fault Code: env:Client
```

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

it 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.

Possible causes are as follows.

A previous version of the rlm.war file is present in your webapps directory

If you are using the default port number of 9095, check to 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` directory.

Your Web services are pointing to the wrong port number

If you are using a non-default port number (not 9095), ensure that you have followed the recommended procedures for configuring this. This message may indicate that you have not changed the port in the Web services WSDLs. See [Chapter 8, "Configuring Release Manager to Use a Different Port"](#) on page 124.

Dimensions CM DCR Projects Appear in the Wrong Section

If you are using Dimensions CM as a DCR provider, you must do some additional configuration to ensure that the Dimensions CM DCR projects and DU projects appear correctly in the Release Package and Projects views.

You must do the following:

- 1 Modify the Dimensions CM provider properties file to specify the appropriate provider names for the DCR and DU providers.
- 2 Modify the reports that underlie the User Interface elements for the project listings so that they filter using the new names.

See:

- [Modify the Dimensions CM Provider Properties Files](#)
- [Modify the User Interface Reports](#)

Modify the Dimensions CM Provider Properties Files

Modify the Dimensions CM provider file, such as `dm_qlarius.properties`, so the file specifies a different provider name for each of the types of elements Dimensions CM provides.

Example

```
dm_qlarius.properties
```

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

# deploy units provider definitions
deploy.units.provider.name = DU_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
```

Modify the User Interface Reports

So that the User Interface project listings pick up the new report names, you must modify the reports used for this purpose.

In the **Rim Aux** process app, change the following reports to use the values in the search filters given:

1 Modify the report DCR Projects Assigned to a Release Package.

Modify the report search filter similarly to the following example, using a condition for the Provider Name to match your entry in the `requests.provider.name` value.

a Select Use Basic Conditions.**b Set the conditions to:**

Related package in (Query at Runtime)

and

Provider Name contains "DCR_"

2 Modify the report Dimensions CM Projects Assigned to a Release Package.

Modify the report search filter similarly to the following example, using a condition for the Provider Name to match your entry in the `deploy.units.provider.name` value.

a Select Use Basic Conditions.**b Set the conditions to:**

Use Basic Conditions

Related package in (Query at Runtime)

and

Provider Name contains "DU_"

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 for Release Train, Application Release, Release Packages, Deployment Tasks, and Deployment Process Templates are based on typical release workflows.

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

This 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	146
Release Train Workflow	148
Application Release Workflow	148
Release Package Workflow	149
Deployment Task Workflows	153
Deployment Process Template Workflow	158
Environment Workflow	159

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

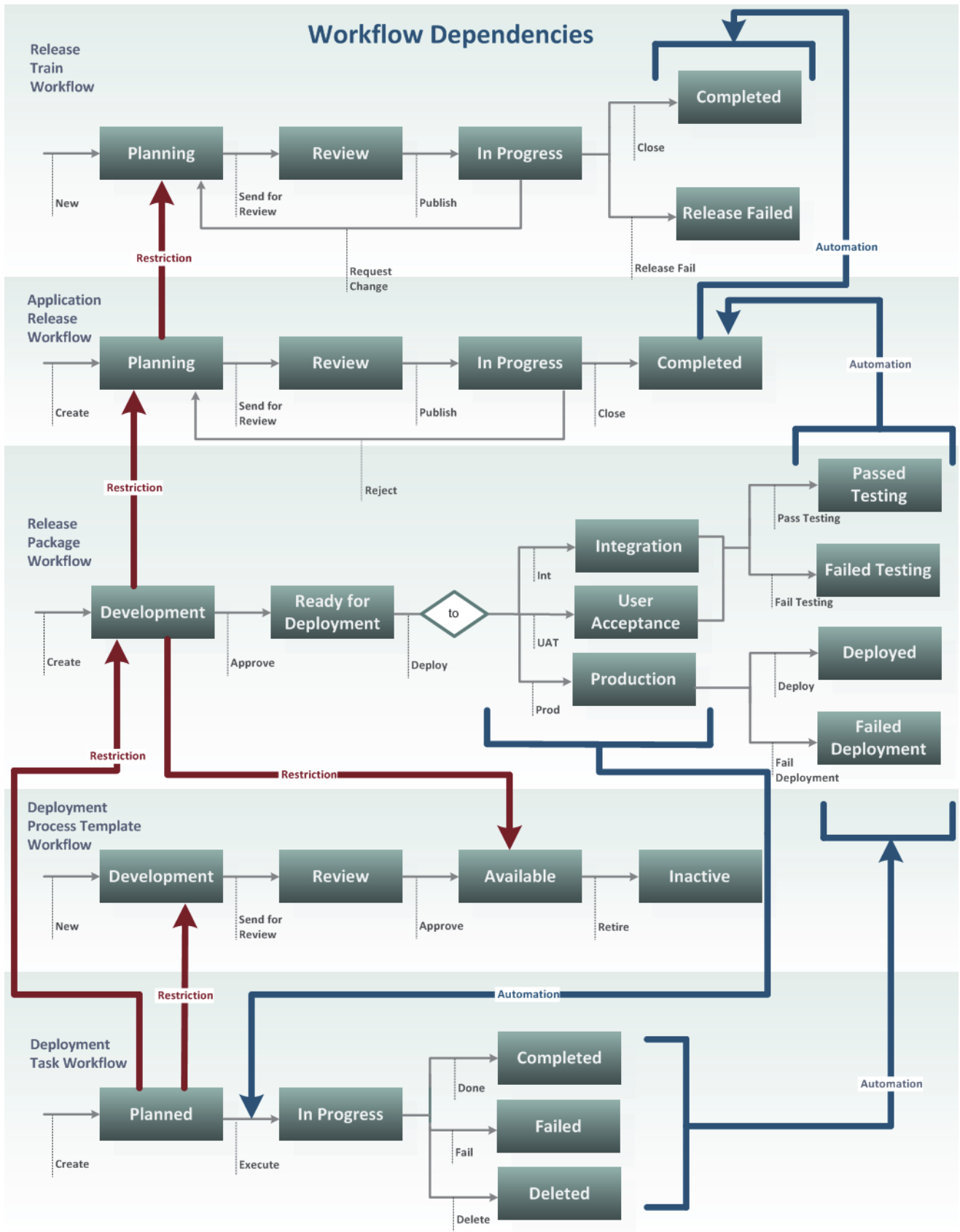
Most of the default workflows allow editing and associating entities to other entities only when items are in the Development or Planning states. For example, you can associate application releases with a release train only when that release train is in the Planning state. You can associate release packages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

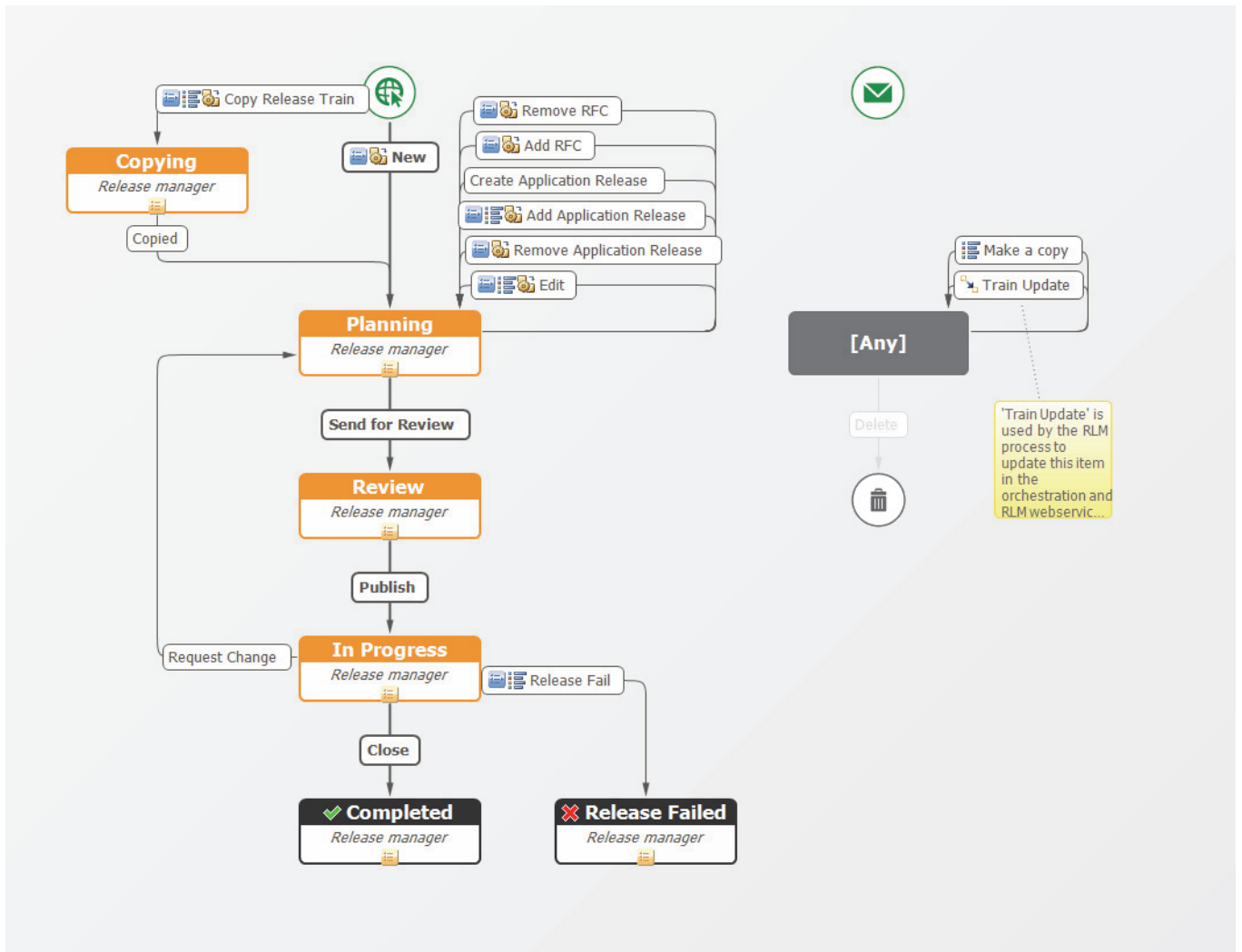
Many of the workflows are moved to completion states automatically when associated entities are complete. The automation is usually initiated at the lowest level of association and moves up the hierarchy as each level of release activities are completed. For example, after all deployment tasks for a 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.



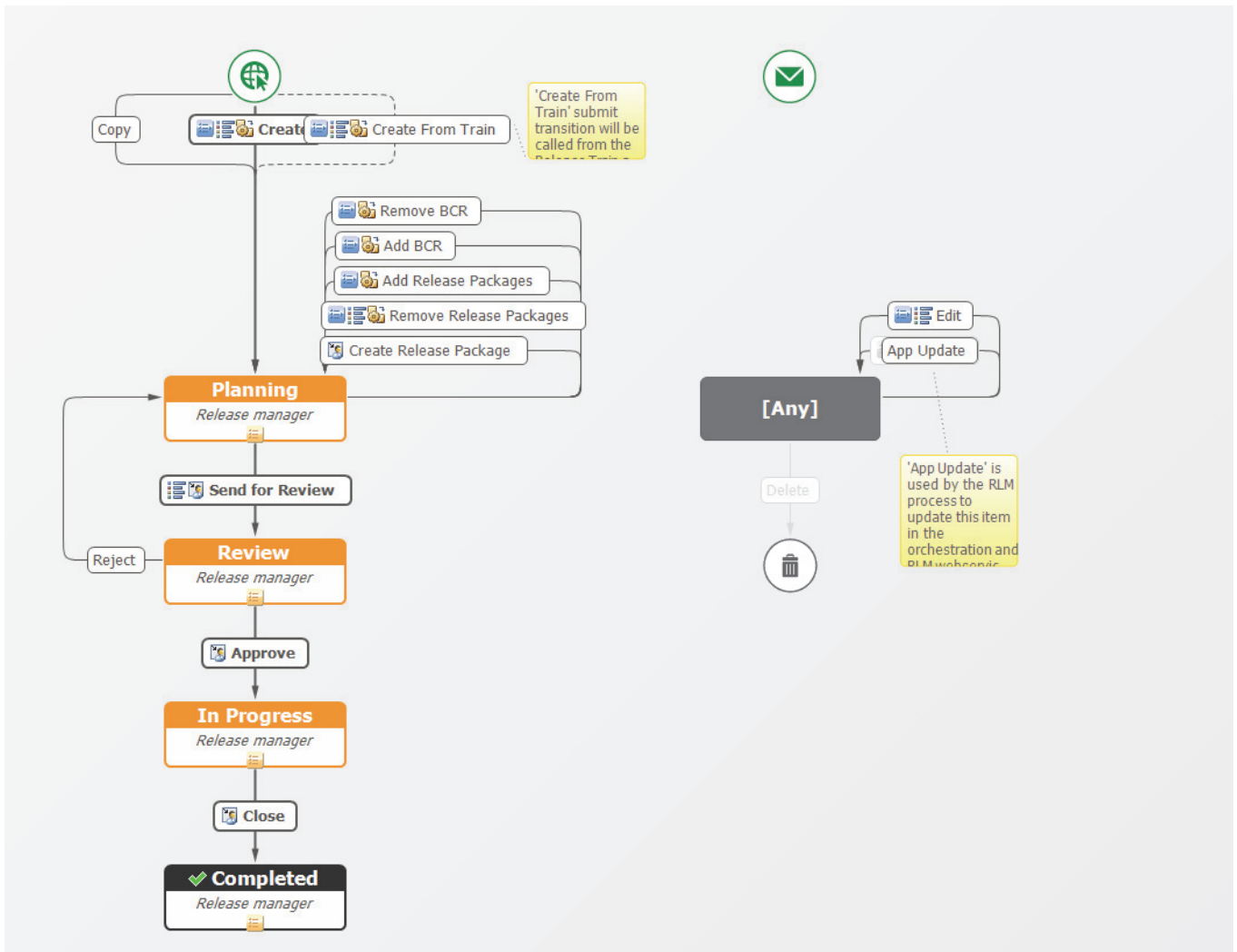
Release Train Workflow

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



Application Release Workflow

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

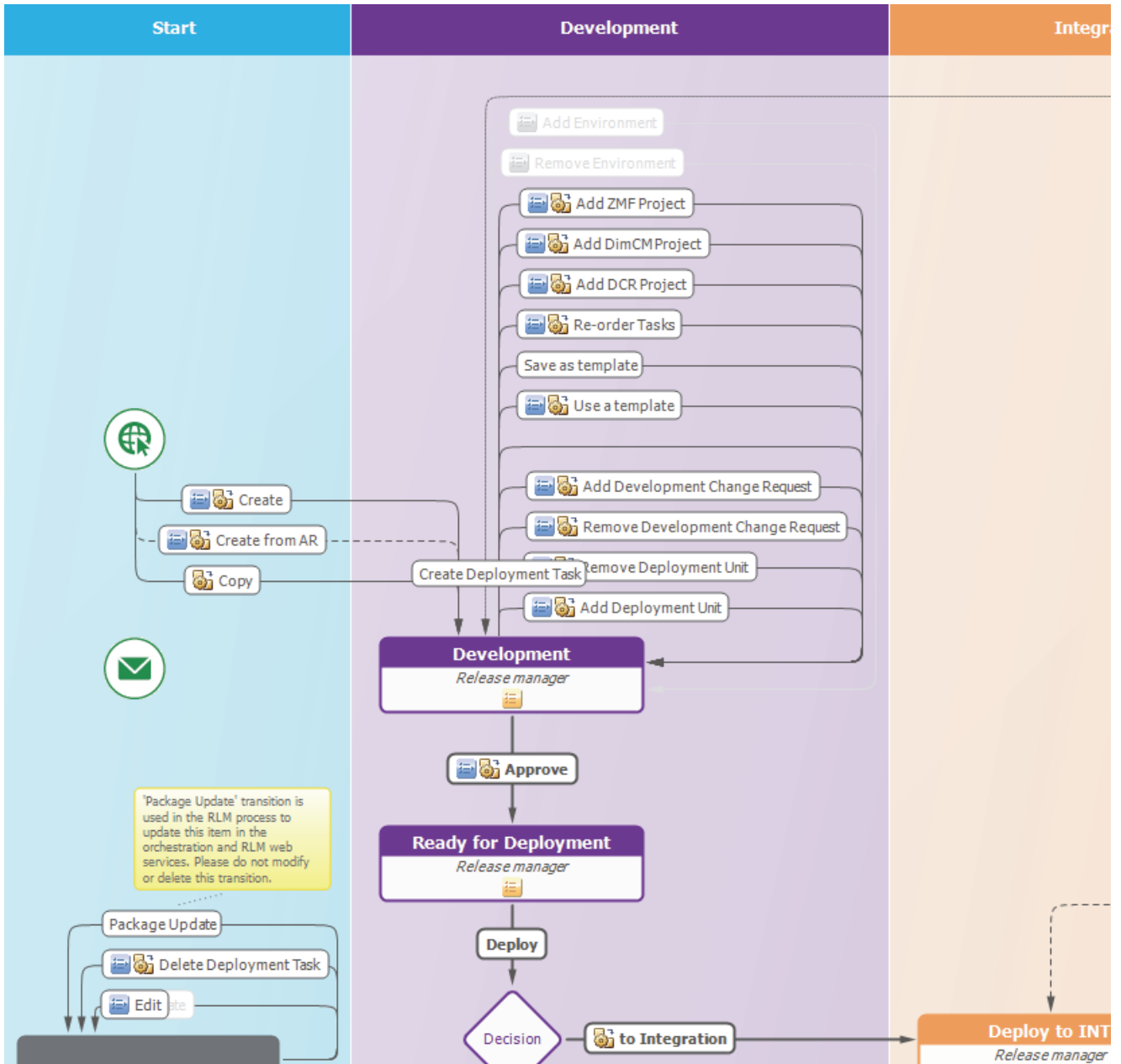


Release Package Workflow

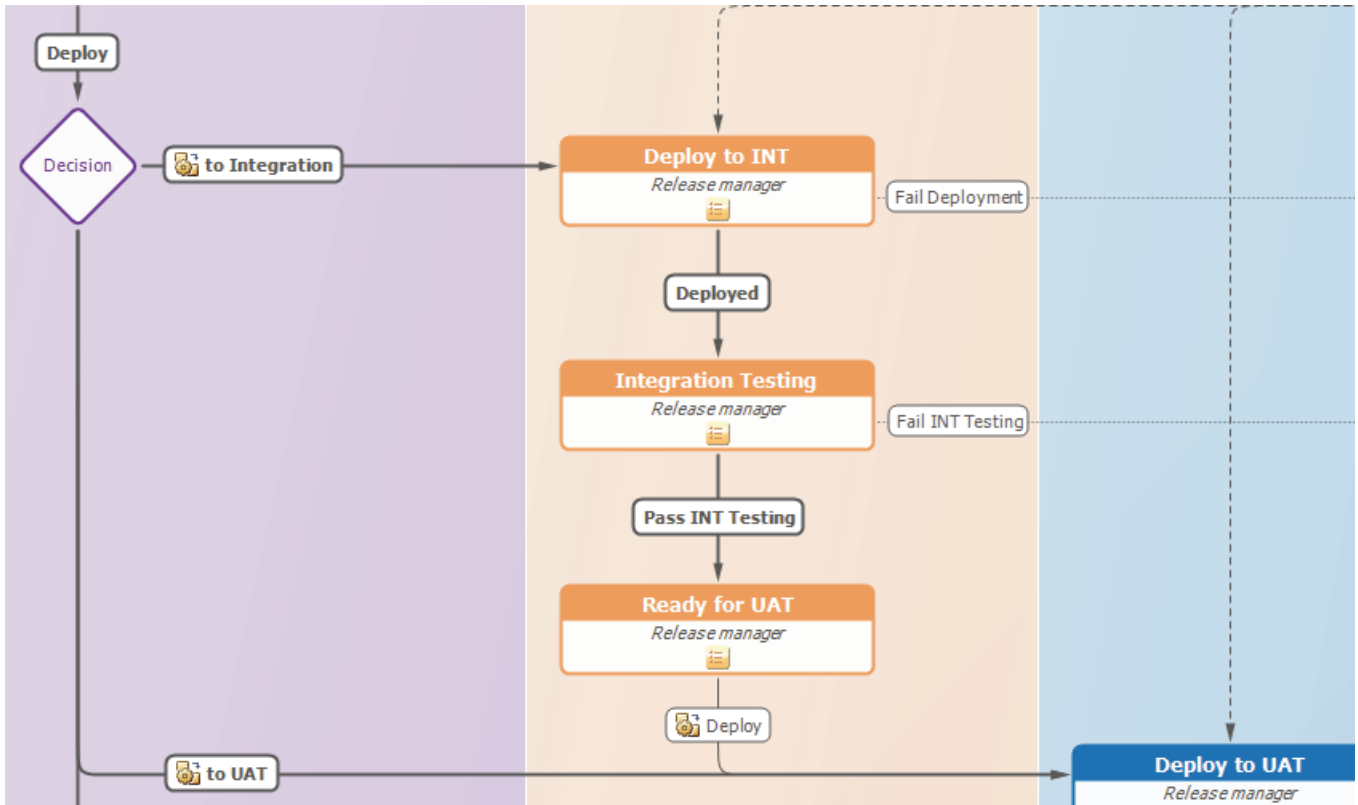
The Release Package workflow in the default version of Serena Release Manager is shown in the following figures. Because the workflow is a conditional workflow with multiple swimlanes, the workflow is shown in parts for ease of viewing.

- "Start and Development States" on page 150
- "Integration State" on page 151
- "Staging and Production States" on page 152
- "Exceptions State" on page 153

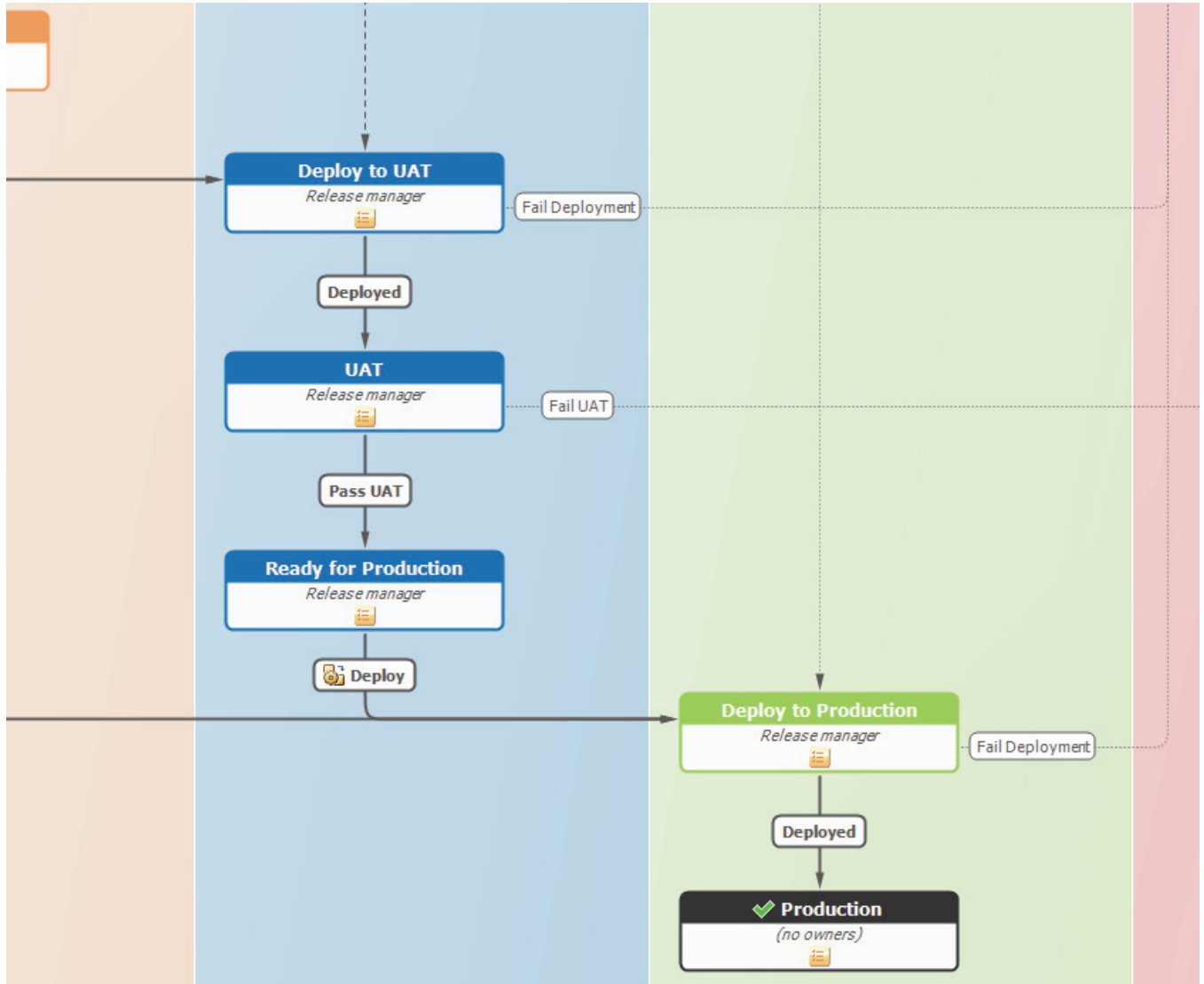
Start and Development States



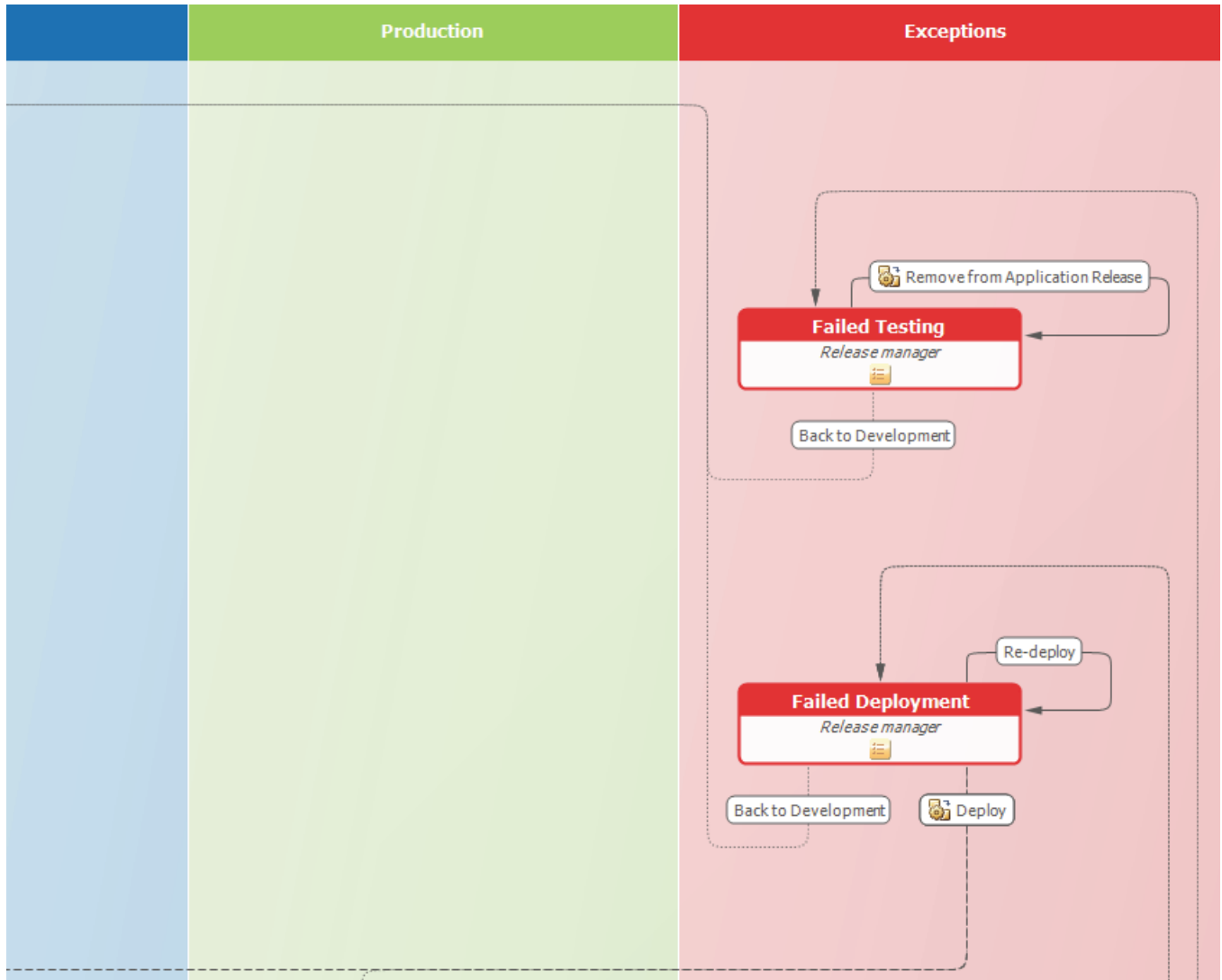
Integration State



Staging and Production States



Exceptions State

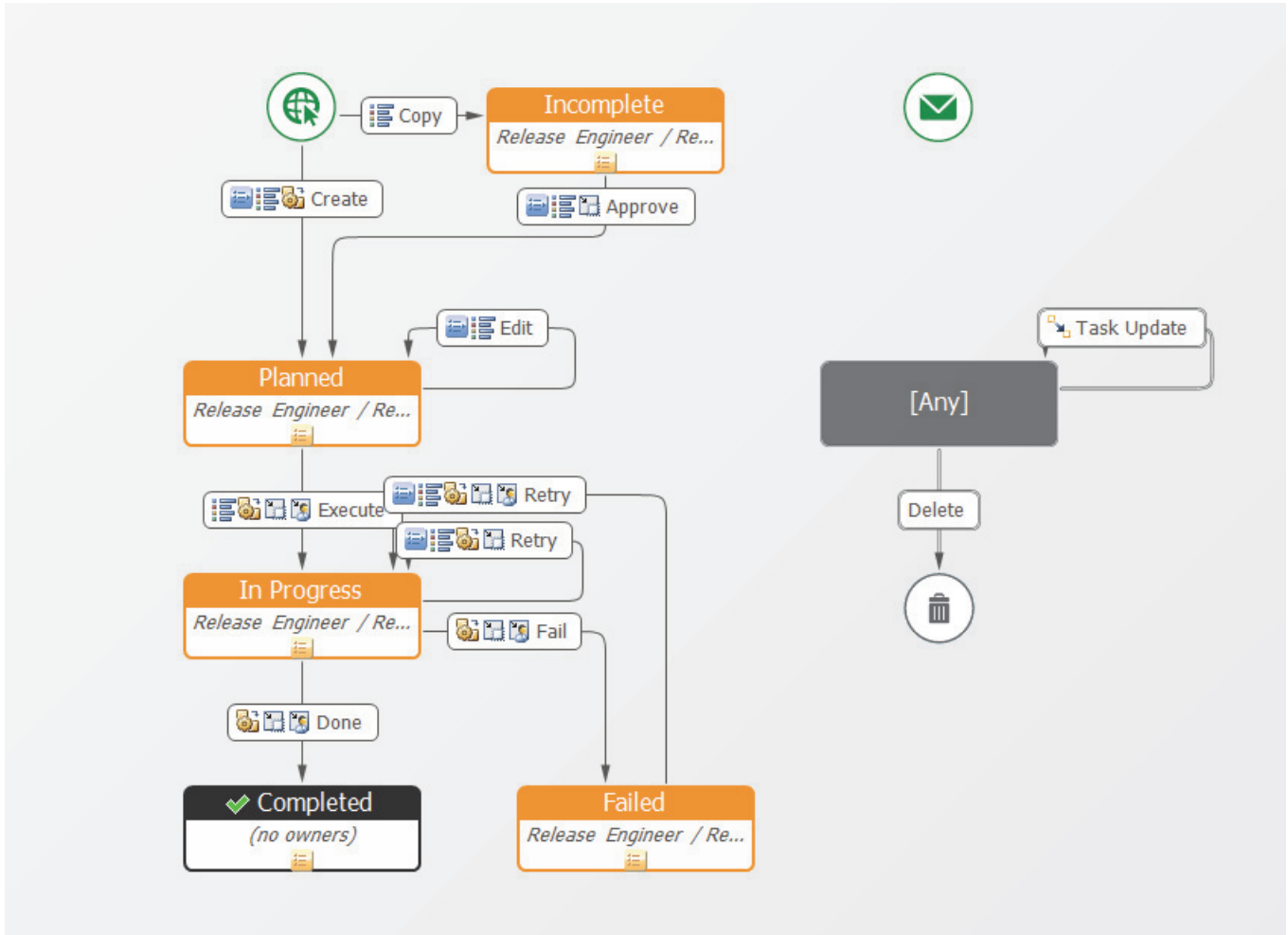


Deployment Task Workflows

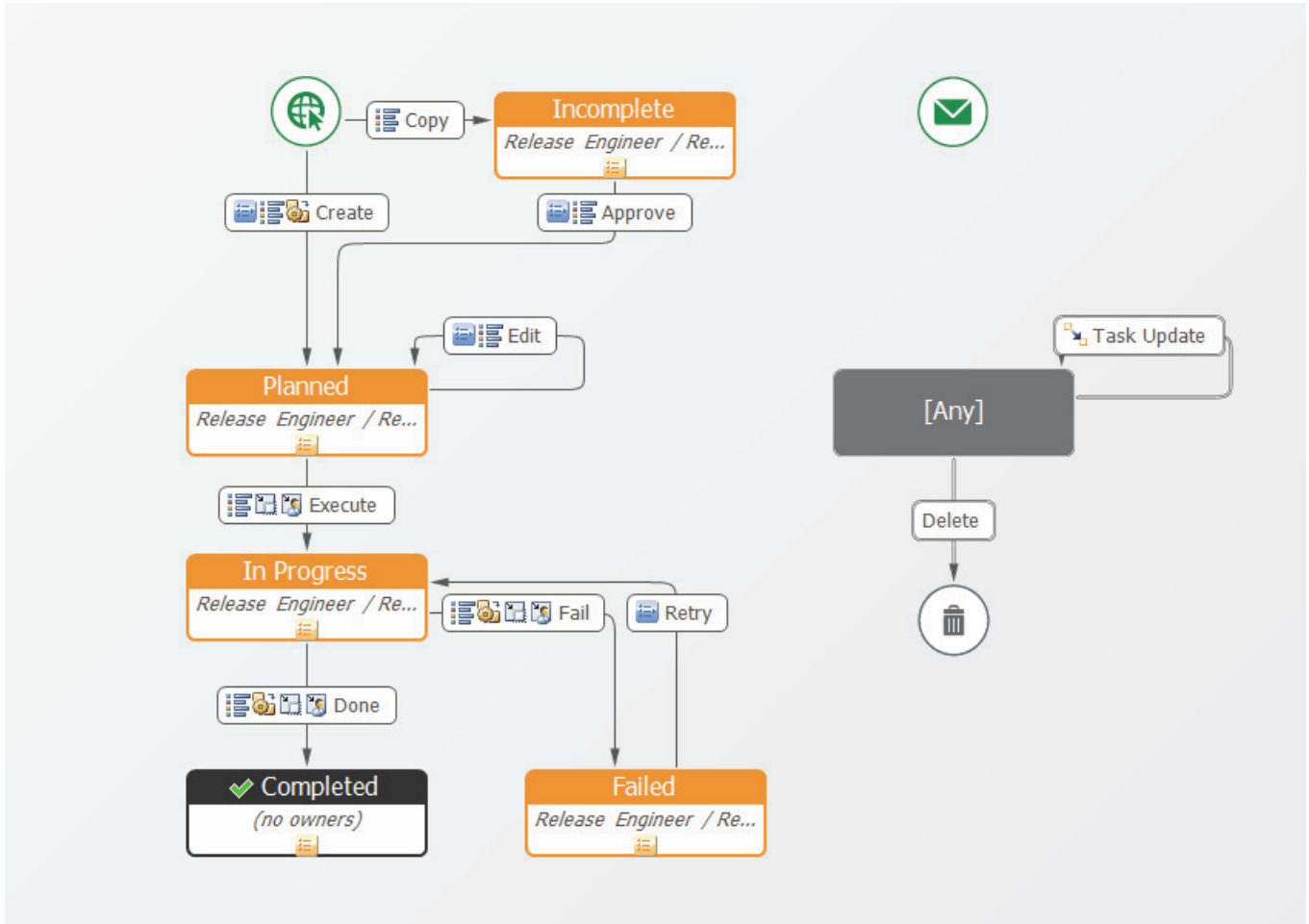
The Deployment Task workflows in the default version of Serena Release Manager are shown in the following figures.

- "Automation Deployment Task Workflow" on page 154
- "Manual Deployment Task Workflow" on page 155
- "Vault Deployment Task Workflows" on page 155

Automation Deployment Task Workflow



Manual Deployment Task Workflow

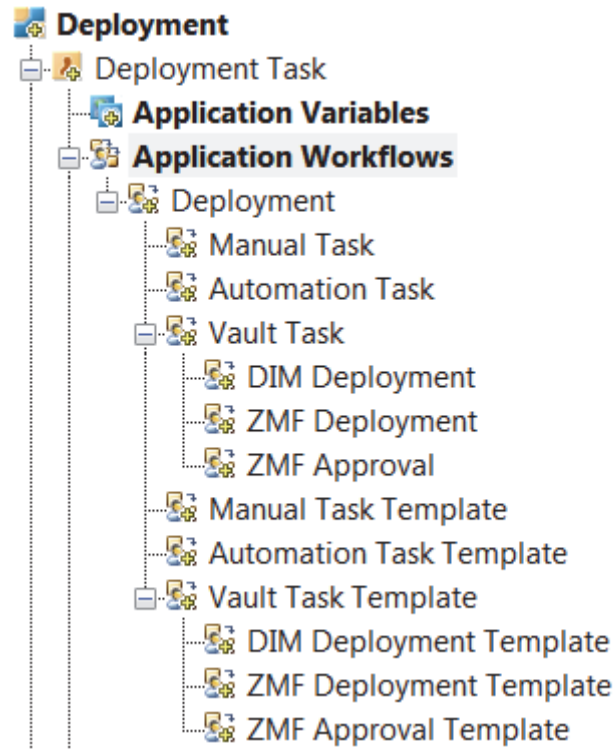


Vault Deployment Task Workflows

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

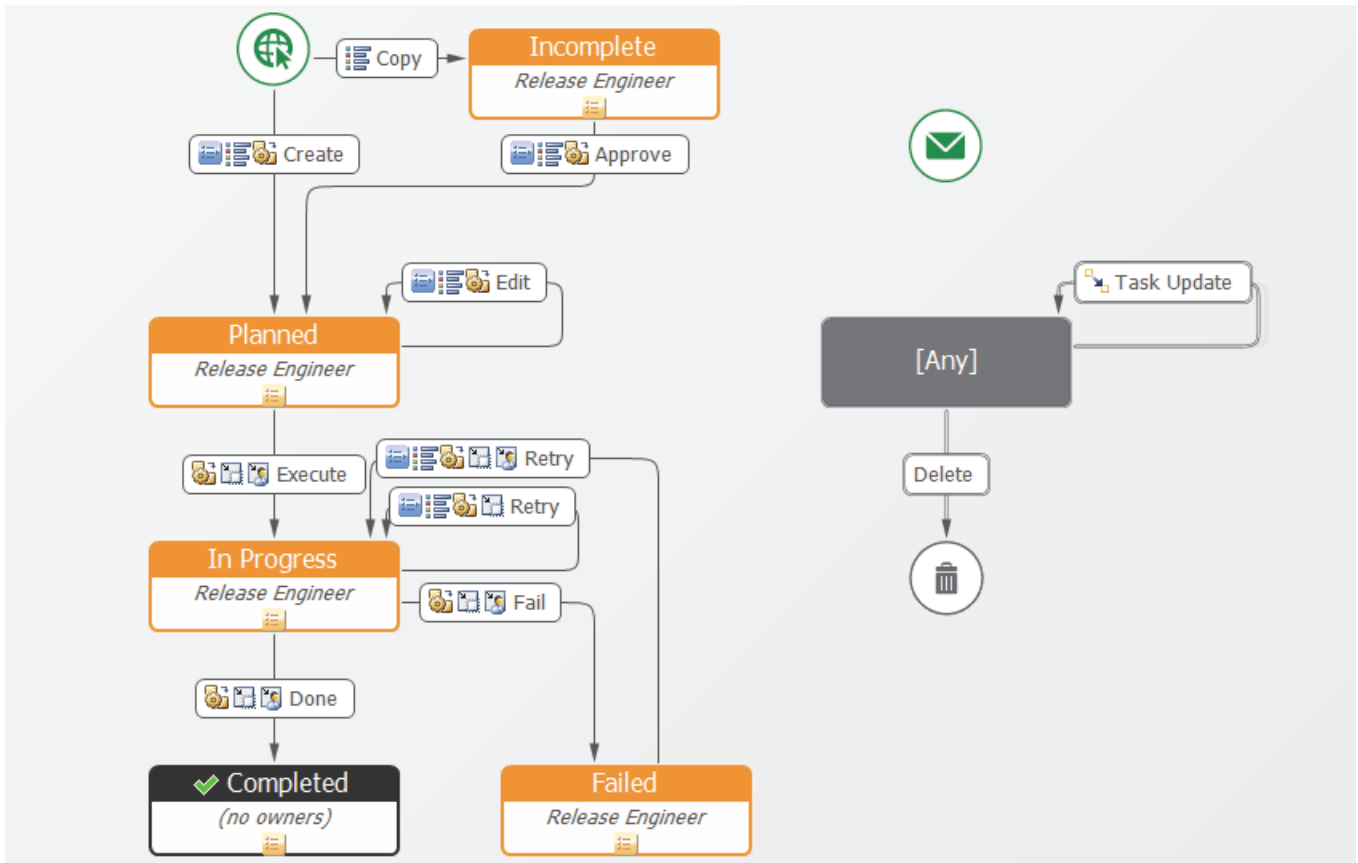
Deployment Task Hierarchy

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



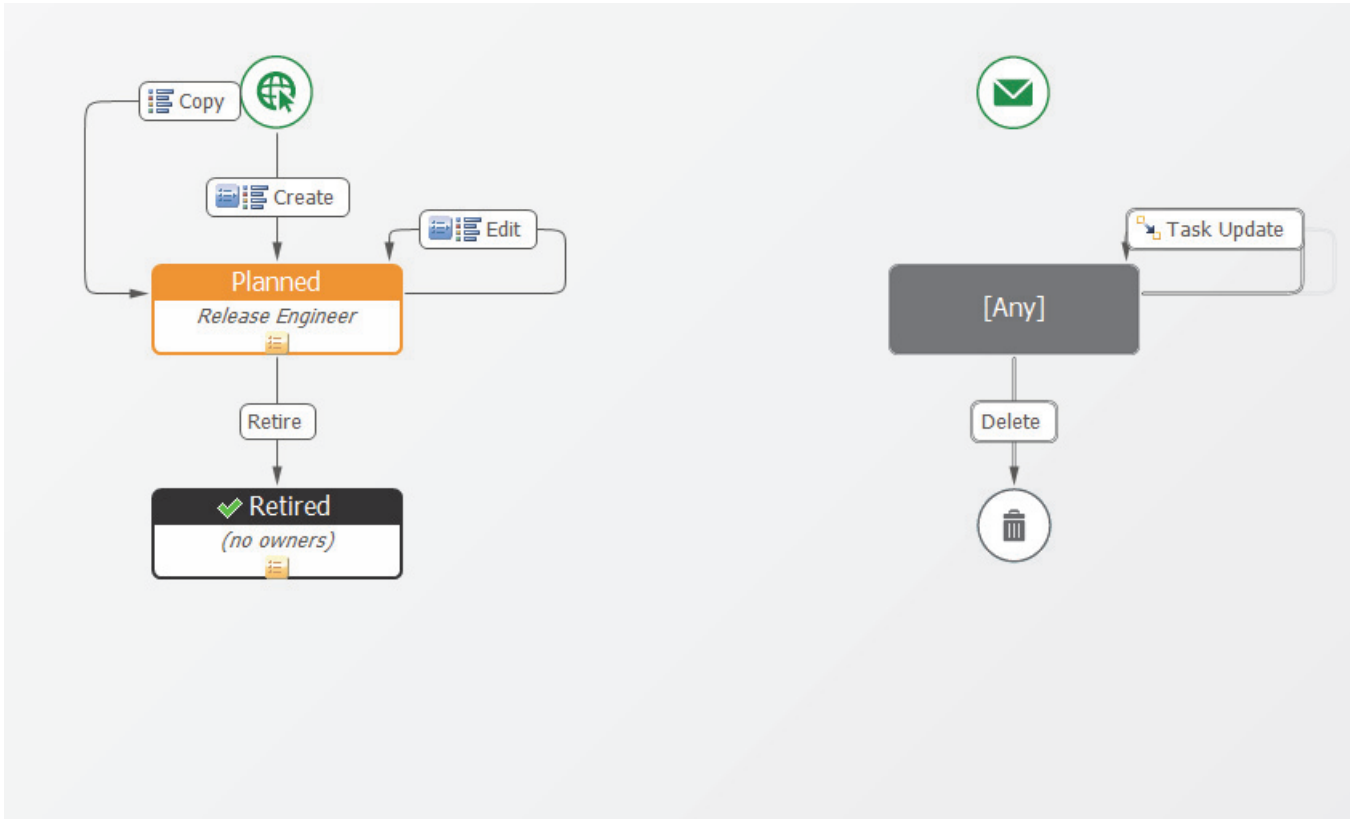
Main Vault Task Workflow

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.



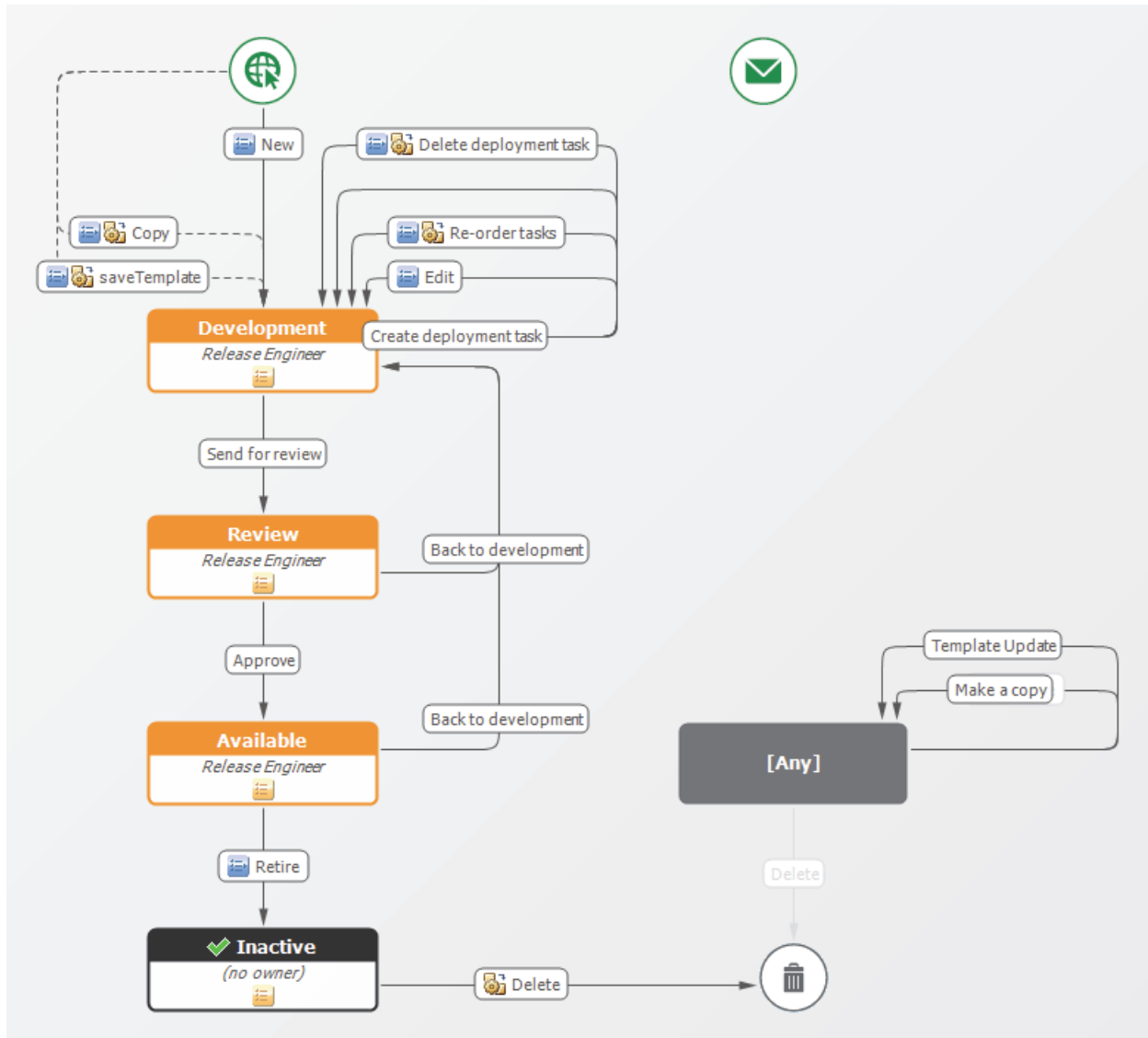
Main Vault Task Template Workflow

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.



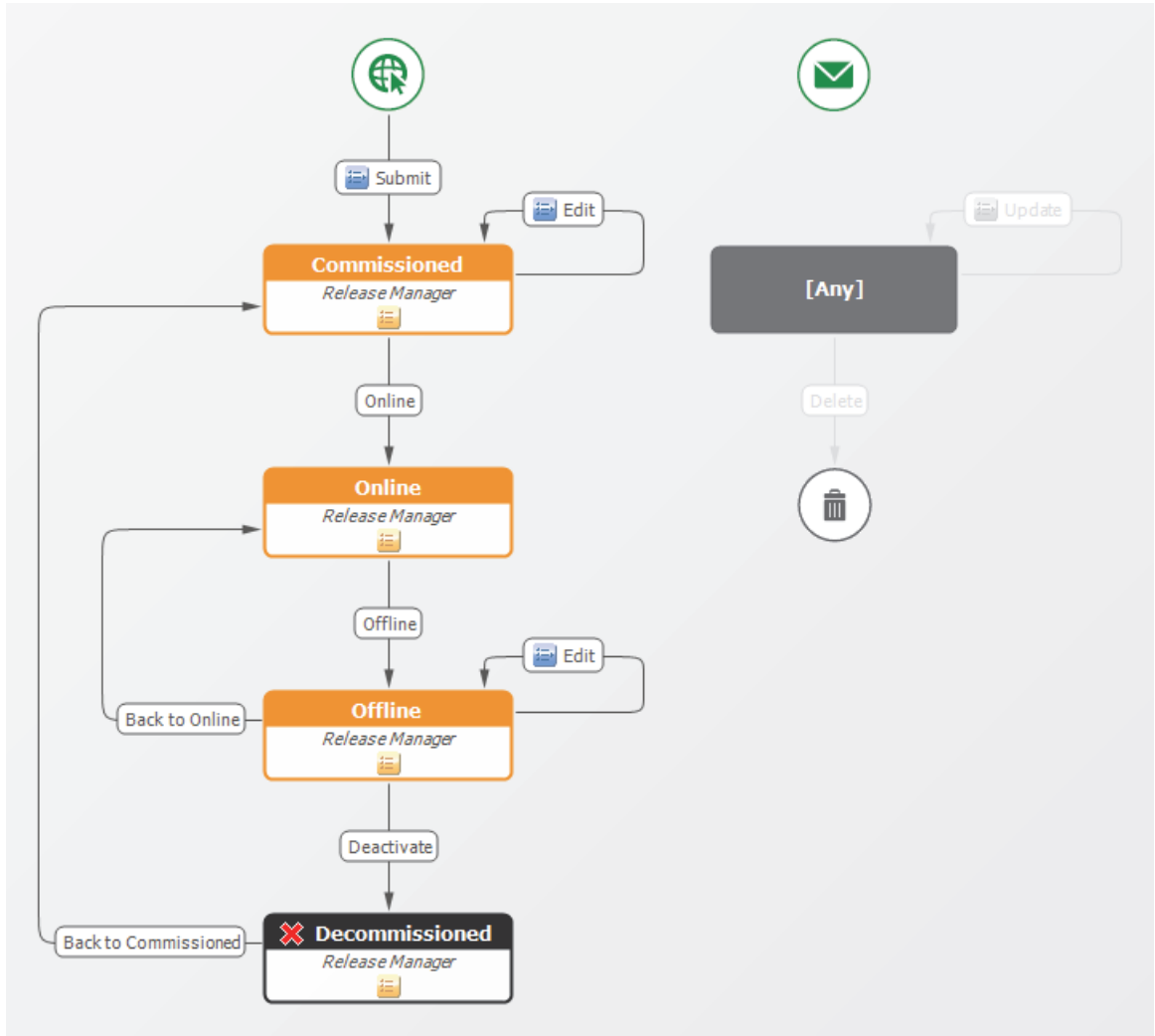
Deployment Process Template Workflow

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.



Index

- A**
 - activity log 136
 - ALF event notification
 - configuring for release automation 55
 - configuring for ZMF 51
 - ALF variables
 - updating for Dimensions CM 43
 - Apache Tomcat
 - See Tomcat Web server
 - verifying Web Services install 28
 - application releases
 - workflow, full 148
 - applications
 - names, adding 58
 - relating to application releases 58
 - Serena Release Automation, accessing 72
 - automations
 - workflows 146
- B**
 - BCR
 - providers, implementing 119
 - BPEL engine
 - for process app, setup 35
 - browser cache, clearing 138
 - business change request provider
 - specifying 78
- C**
 - cache, clearing 138
 - calendar
 - extending 133
 - reports 100
 - ChangeMan ZMF 52
 - change packages, specifying 52
 - configuring on the mainframe 46
 - deployment units, filtering 84
 - notify parameter 47
 - notifying ALF 47
 - selection criteria 52
 - conditions
 - solutions 137
 - configuration
 - quickstart 18
 - requirements 15
 - configuring
 - ALF event notification for release automation 55
 - ALF event notification for ZMF 51
 - BCR provider 78
 - DCR provider 79
 - RFC provider 77
 - connecting
 - Serena Release Automation to SBM 54
 - customization 97
- D**
 - dashboard
 - configuring 98
 - reports 100
 - databases
 - requirements, release automation 14
 - DCR
 - filtering, incidents 82
 - filtering, issues 82
 - providers, implementing 119
 - deployment
 - history 136
 - log files 137
 - results 136
 - deployment process templates
 - workflow, full 158
 - deployment tasks
 - workflows, full 153
 - deployment unit providers
 - specifying 83
 - deployment units
 - ChangeMan ZMF, filtering 84
 - Dimensions CM, filtering 84
 - development change request provider
 - specifying 79
 - dialogs
 - configuring 103
 - reports 103
 - Dimensions CM
 - configuring in Release Manager 45
 - configuring objects in 68
 - configuring on the server 43
 - deployment units, filtering 84
 - installation 26
 - requests, filtering 81

- selection criteria 44
- user ID 41
- users, configuring 69
- documentation, guide to 7
- DU
 - providers, implementing 119

E

- environments
 - associating to release packages 131
 - creating for process app 35
 - Serena Release Automation, accessing 72
 - workflow, full 159
- error conditions
 - messages 136
 - overview 136
 - sources of information 136
 - symptoms 137
- event manager
 - for process app, setup 36
- event notification
 - Serena Release Automation, configuration 52
- execution server
 - Serena Release Automation, specifying 56

F

- failures
 - activity log 136
 - error messages 136
 - history 136
 - log files 137
 - overview 136
 - solutions 137
 - sources of information 136
 - symptoms 137

G

- Global Stage Lifecycle
 - mapping stages to 68

H

- history 136

I

- inbox 101
 - configuring 101
- installation

- Apache Tomcat requirements 28
- common Web server 27
- Dimensions CM 26
- order 24
- overview 24
- preparing for 13
- prerequisites 14
- SBM 25
- Serena License Manager 24
- Serena Release Automation 29
- Serena Release Control 26
- supporting files 28
- verifying ZMF Web Services 28
- installation quickstart 18

L

- licenses
 - applying 30
- log
 - activity 136
 - files 137

M

- manuals, guide to 7
- messages
 - log 136
 - solutions 137
- My Inbox
 - reports 101

N

- notifications, e-mail
 - Serena Release Control, managing 62

O

- optimizing
 - response time 141
- order of installation 24

P

- port
 - conflicts 124
 - for Web server, changing 125
 - for Web server, default 27
 - for Web server, pointing to 126
 - for Web server, prior versions 126
 - for Web server, using non-default 124

- post-installation 46
 - ChangeMan ZMF 46
 - Dimensions CM 43
 - Serena License Manager 30
 - Serena Release Automation 52
 - Serena Release Control 33
 - Serena Release Control, Dimensions CM 43
- process and servers
 - Serena Release Automation, configuring 72
- processes
 - configuring 105
 - Serena Release Automation, accessing 72
- projects
 - Dimensions CM 70
 - in SBM 73
- projects, specifying 52
- promoting
 - snapshots, in SBM 36
- provider
 - for requests 79

R

- release control
 - history 136
 - processes, configuring 105
 - release types, configuring 106
 - stages, configuring 106
 - upgrading 90
- release packages
 - associating environments with 131
 - workflow, full 149
- release trains
 - workflow, full 148
- release types
 - configuring 106
- reports
 - for dialogs and views 103
 - for My Inbox 101
 - for the calendar 100
 - for the dashboard 100
 - managing 62
- requests 81
 - configuring provider for 79
 - Dimensions CM, filtering 81
- requests for change provider
 - specifying 77
- requirements
 - databases, release automation 14
 - server 15
 - software 15
 - system 15
- response time
 - troubleshooting 141
- restrictions

- workflows 146
- results
 - activity log 136
 - log files 137
- RFC
 - providers, implementing 119
- roles
 - Dimensions CM user ID 41
 - examples 61

S

- SBM
 - Application Administrator, navigating to 40
 - configuring entities in 73
 - features, accessing 58
 - filtering, incidents 82
 - filtering, issues 82
 - installation 25
 - projects, configuring in 73
 - user interface, accessing 58
- Serena License Manager
 - installation 24
 - post-installation 30
- Serena Release Automation
 - applications, accessing 72
 - configuring objects in 71
 - connecting 55
 - connecting to SBM 54
 - environments, accessing 72
 - event notification configuration 52
 - execution server, specifying 56
 - installation 29
 - post-installation 52
 - process and servers, configuring 72
 - processes, accessing 72
 - server, specifying 52
 - servers, accessing 72
 - users, configuring 71
- Serena Release Control
 - calendar, configuring 100
 - dashboard, configuring 98
 - dialogs, configuring 103
 - importing the solution 34
 - installation 26
 - notifications, e-mail 62
 - post-installation 33
 - reports 62
 - roles, examples 61
 - upgrading 90
 - user interface, configuring 98
 - users, managing 60
 - views, configuring 103
- servers
 - names, adding 59

- recommended configuration 141
- relating to environments 59
- requirements 15
- Serena Release Automation, accessing 72
- time, optimizing 141
- shell
 - customizing 128
 - installing 39
 - upgrading 94
- shell UI, exiting 58
- snapshots
 - promoting, in SBM 36
- software compatibility 15
- solution, importing 34
- solutions
 - for failure conditions 137
- specifying
 - BCR providers 78
 - DCR providers 79
 - DU providers 83
 - RFC providers 77
- SSM
 - configuring RFC reports for 131
 - sending information to 131
- SSO
 - existing Dimensions CM installations 26
 - new Dimensions CM installations 26
- stages
 - configuring 106
- status
 - history 136
- streams
 - Dimensions CM 70
- system
 - requirements 15

T

- target servers
 - for process app, BPEL engine 35
 - for process app, event manager 36
- Tomcat Web server
 - changing the port 125
 - common, installed 27
 - default port 27
 - default port, prior versions 126
 - port conflicts 124
 - port, pointing to 126
 - using a non-default port 124
 - verifying Web Services
- troubleshooting
 - activity log 136
 - error messages 136
 - history 136
 - log files 137

- overview 136
- solutions 137
- sources of information 136
- symptoms 137
- Web services requirements
- tutorials, guide to 7

U

- UI elements
 - customizing 128
 - installing 39
 - missing 139
 - upgrading 94
- UI shell
 - installed 28
- upgrade instructions 90
- user interface
 - accessing SBM features 58
 - configuring 98
- users
 - Dimensions CM, configuring 69
 - Serena Release Automation, configuring 71
 - Serena Release Control, managing 60

V

- views
 - configuring 103
 - reports 103

W

- war files
 - installed 28
- Web server
 - See Tomcat Web server
- workflows
 - application releases, full 148
 - automations 146
 - deployment process templates, full 158
 - deployment tasks, full 153
 - environments, full 159
 - release package, production 152
 - release package, Production Deployment 152
 - release package, staging 152
 - release package, UAT 152
 - release packages, development 150
 - release packages, exceptions 153
 - release packages, full 149
 - release packages, integration 151
 - release packages, start 150
 - release trains, full 148

restrictions 146

Z

ZMF Web Services
 verifying install 28

