



SERENA®

ALM

Installation and Configuration Guide

Serena Proprietary and Confidential Information

Copyright © 2011–2013 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.

Trademarks

Serena, TeamTrack, StarTool, PVCS, Comparex, Dimensions, Prototype Composer, Mariner and ChangeMan are registered trademarks of Serena Software, Inc. The Serena logo, Version Manager and Mover 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., 1850 Gateway Drive, 4th Floor, San Mateo, California, 94404-4061.

Product version: 4.0.1

Publication date: 2013-03-14

Table of Contents

Welcome to Serena ALM	1
Accessing the Documentation	9
What is Serena Release Manager?	10
What is Serena Development Manager?.....	10
Installation Prerequisites and Planning	12
Serena Release Manager Architecture	12
Preparing for the Installation	12
Installation Prerequisites	13
Database Requirements	14
Software Compatibility Requirements	15
System Requirements	15
Planning Checklists and Worksheets	15
Installation Planning for SBM Connector for HP Quality Center / ALM Integration ...	16
Installing and Configuring Serena Requirements Manager	16
Installing and Configuring the Serena Dashboard	16
Serena ALM Installation	17
Installation Overview	17
Installation Order	17
Installation and Configuration Checklist	18
Before Running the Installer	20
Running the Installer	22
Importing and Promoting Serena ALM	26
Importing the Serena ALM Solution	26
Creating an Environment for Serena ALM	28
Which Snapshots Should I Promote?.....	29
Promoting the Release Manager Snapshots.....	31
Promoting the Development Manager Snapshots	33
Configuring the Release Manager Process Applications	37
Setting Authentication for Added Endpoints.....	37
Promoting the Snapshots Again to Resolve References	37
Configuring SSO in RESTgrid Widgets	37
Publishing and Deploying the Release Manager Process Apps.....	38
Configuring the Development Manager Process Applications	39
Updating the Development Manager Endpoints	39
Updating the Connection Details in the Workflows.....	41
Republishing the Process Apps	44
Deploying the Development Manager Process Apps	44

Serena ALM Configuration	48
Applying Licenses	48
Configuring the Admin User for Release Manager	48
Creating an Administrative User	49
Configuring the Administrative User Privileges	49
Enabling Serena Release Manager Project Roles	50
Creating the Service User for Development Manager	51
Using the Serena ALM Configurator.....	53
Configuring Access to the Configurator.....	53
Invoking the Serena ALM Configurator	53
Entering Information in the Serena ALM Configurator	53
ALF Client Connection Information.....	54
SBM Application Connection Information	54
BCR Provider Connection Information	55
RFC Provider Connection Information	56
Dimensions CM Client Connection Information	56
Serena Release Automation Client Connection Information	57
Serena Release Automation (Nolio) Client Connection Information	58
ZMF Client Connection Information	58
Accessing the Standard SBM User Interface	59
Release Manager Configuration	60
Serena Release Manager Runtime Communication	60
Adding Your Application Names in Serena Release Manager	61
Adding Your Server Names in Serena Release Manager	61
Enabling Scheduling of Deployment Tasks	62
Managing Release Manager Users	63
Managing Release Manager User Interface Reports	65
Changing Existing Reports	66
Creating New Reports	66
Naming Convention and Reference Rules	66
List of Reports	68
Privileges Required for UI Reports	71
Managing Release Manager Notifications	71
Development Manager Configuration	75
Creating Users and Assigning Privileges	75
Creating and Assigning the Development Manager Users	75
Configuring Agile Properties.....	76
Dimensions CM for Serena ALM Configuration	78
Dimensions CM for Release Manager Configuration	78
Dimensions CM Communication Configuration Overview	78
Configuring Communication on the Dimensions CM Server.....	79

Specifying Dimensions CM ALF Event Configuration Information	80
Specifying Selection Criteria for Dimensions CM Events and Objects	81
Configuring Dimensions CM Communication in Release Manager	81
Configuring Objects in Dimensions CM	82
Configuring the Dimensions CM Global Stage Lifecycle	82
Managing Dimensions CM Users	82
Configuring Dimensions CM Projects and Streams	83
Available Selection of Requests and Baselines	84
Dimensions CM for Development Manager Configuration	84
Configuring the Dimensions CM Server for Development Manager	85
Configuring Dimensions CM for ALF Events	85
Assign Build Engineer Role	86
Assign Development Engineer Role as Delegation Candidate	88
Assign Development Engineer Role to TASK Lifecycle Transitions	88
Setting Privileges for the Service User	90
Configuring Dimensions CM for Products and Components	92
ChangeMan ZMF for Serena ALM Configuration	93
ZMF Communication Configuration Overview	93
Configuring ZMF Communication on the Mainframe	93
Configuring the Notification URL	94
Configuring the SERNET HTTP Server	94
SERSERV Prerequisites	95
Installing SERSERV	97
Verifying the Installation of SERSERV	101
SERSERV Runtime Considerations	101
Configuring a ZMF Proxy User ID	103
Configuring TSO User IDs and Permissions	104
Configuring ZMF Approvers	104
Configuring Objects in ChangeMan ZMF	104
Configuring ZMF Communication in Serena ALM	105
Specifying ALF Event Manager Connection Information for ZMF	105
Specifying Client-Specific Information for ChangeMan ZMF	106
Configuring the Release Manager ZMF Web Page Widgets	107
Configuring the Development Manager ZMF Web Page Widgets	108
Release Automation for Serena ALM Configuration	109
Release Automation Communication Configuration Overview	109
Release Automation Architecture	109
Release Automation (Nolio) Architecture	110
Configuring Communication with Release Automation	110
Configuring Communication on the Release Automation Server	110
Configuring Release Automation Communication in Release Manager	111

Configuring Objects in Serena Release Automation	112
Configuring Communication with Release Automation (Nolio)	114
Configuring Communication on the Release Automation (Nolio) Server	114
Configuring Release Automation (Nolio) Communication in Release Manager	116
Configuring Objects in Release Automation (Nolio)	118
Provider Configuration	121
Release Manager Provider Overview	121
Telling Release Manager Which Providers to Use	121
Configuring Access to Requests for Change.....	122
Designating the Details for Each RFC Provider	123
Configuring Access to Business Change Requests	124
Designating the Details for Each BCR Provider	124
Configuring Access to Development Change Requests	125
Designating the Details for Each DCR Provider	125
Configuring Access to Deployment Units	128
Release Manager DU Provider Details	129
Configuring Objects in Serena Business Manager	129
Configuring Objects in Serena Service Manager.....	130
Development Manager Provider Overview	131
Telling Development Manager Which Providers to Use	131
Development Manager Dev Task Provider Details	132
Vault Request Configuration	138
Vault Request Purpose	138
Vault Request Architecture	139
The Vault in a Vault Request	139
Definitive Media Library (DML)	139
Vault Template Purpose	139
Release Manager Upgrade Scenarios.....	140
Serena Release Manager Typical Upgrade	140
Serena Release Manager Manual Upgrade.....	140
Upgrading Serena Release Manager Manually	141
Development Manager Installation and Upgrade Scenarios.....	147
Configuring a Stand-Alone Development Manager Installation	149
Hiding the Requirements Manager-related Forms	149
Removing Requirements Manager-related Columns from Report Definitions	150
Removing Requirements Manager-related Columns from Reports	150
Removing the Viewers Role from the Requirements Manager Project	151
Adapting an Existing Stand-alone Requirements Manager Installation for use with Development Manager	151
Setting the Correct Endpoint for rtmService	152

Unhiding the Development Manager-related Forms	152
Enabling the Distribute transition for the Approval Process Application Workflow	152
Adapting Development Manager to Hide Products and Components	153
Hiding Form Fields for Products and Components	153
Removing the Viewers Role from the Component Lifecycle Project	153
Serena Release Manager Customization	154
Customizing Release Manager Workflows	154
Modifying Release Types and Stages	155
Summary of Adding a Stage	156
Example of Adding a Stage	157
Adding Provider Connections.....	171
Creating a Class for Your Provider	172
Creating Properties Files for Your Providers.....	172
Building and Packaging	175
Telling Serena Release Manager to Use This Provider	175
Configuring Release Manager to Use a Different Port.....	176
Checklist for Changing the Port Number	177
Changing the Port on Which the Common Web Server Runs	177
Configuring a Non-Default Web Server Port in the Process Apps	178
Changing the Web Services to Point to a Different Port	180
Activating Environment Association to Release Packages	180
Customizing the SSM Integration	181
Creating the Reports Used for the SSM Integration	182
Changing the Integration in SSM	183
Configuring the SSM Feed for the Release Train Calendar	183
Changing the Instance of SSM that Release Manager Uses.....	184
Setting Maximum Associations for Release Manager Objects	184
Customizing the User Interface	185
Troubleshooting Release Manager	186
Troubleshooting Overview	186
Information from the User Interface	186
Error Messages	186
Activity Log	186
Activity Page	186
History	187
Information from the ALM Configurator	187
Information from Log Files	187
Execution Log File	187
Installer Log Files	187
Symptoms and Solutions	188

RLM Installer Errors	188
Snapshot Promotion Errors	188
Cannot log into Serena Release Manager	190
Cannot Log into Serena ALM Configurator	190
User Interface Display Issues	192
Create Release Fails with a Check Uniqueness Error	194
Unable to update landing page layout	194
Matches Not Found for Selections	194
No Change Package Data Displayed in ZMF Deployment Tasks	195
Release Package Deployment Fails	195
Slow Response Time	196
Troubleshooting Development Manager	198
Installation Issues	199
Upgrading Issues	202
Development Control Issues	206
Configuration Issues	210
Serena ALM Workflow Reference	224
Release Manager Workflows	224
Workflow Relationships	224
Release Train Workflow	226
Application Release Workflow	227
Release Package Workflow	228
Start and Development States	228
Integration State	229
Staging and Production States	230
Exceptions State	231
Deployment Task Workflows	231
Automation Deployment Task Workflow	231
Manual Deployment Task Workflow	232
Vault Deployment Task Workflows	233
Deployment Process Template Workflow	235
Environment Workflow	236
Vault Request Workflow	237
Vault Template Workflow	238
Development Manager Workflows	238
Configuration File Reference	239
Configuration Files on the Serena ALM Server	239
Files in the Classes Folder	239
Files in Other Folders	241
Configuration Files on the Dimensions CM Server	242
Configuration Files on the Release Automation (Nolio) Server	242

Welcome to Serena ALM

Thank you for choosing Serena Orchestrated Application Lifecycle Management (ALM). Serena Orchestrated ALM automates end-to-end application development by helping IT organizations efficiently capture all demand from their stakeholders, manage changes to requirements and software configurations, and deploy releases with confidence. Serena has helped thousands of IT organizations make dramatic improvements to their application development, including greater visibility, faster time to market, higher stakeholder satisfaction, and lower development costs.

Serena Orchestrated ALM is powered by Serena Business Manager (SBM), a powerful platform that allows you to define and automate your organizational workflows. With SBM, you can ensure that all aspects of your development and delivery process are tracked and automated, using fully configurable lifecycles that are modeled after your own processes. The Serena Orchestrated ALM solutions integrate the workflow and automation power of SBM with your critical business systems, including your configuration and demand management systems.

Serena ALM includes the following solutions:

- **Serena Release Manager**, the orchestrated solution for application release management that enables you to plan, manage, and automate the deployment of applications into test, pre-production, and production environments across mainframe and distributed systems with start-to-finish traceability and end-to-end visibility.
- **Serena Development Manager**, the orchestrated solution that enables you to plan, control, and automate all your development processes from definition to quality assurance and release approval with start-to-finish traceability and end-to-end visibility across distributed environments. This includes an optional **Agile Planning** component that enables you to use Agile methodology when managing your development.
- **Serena Requirements Manager**, the orchestrated solution for application requirements management that brings requirements initiation, review, and approval to your end-to-end design, development, and quality process.

Audience and Scope

This document is intended for personnel who are responsible for installing and configuring Serena Release Manager or Serena Development Manager. For documentation on installing and configuring Serena Requirements Manager, see *Serena Requirements Manager Installation Guide*.

Each product in the individual integrated suites have 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 Readmes for Serena Release Manager and Serena Development Manager for the latest updates and corrections for this document.

Procedures and figures in this documentation are examples based on the default implementation of Serena ALM. The details may not match your implementation exactly, but provide a reference to get you started.

[Accessing the Documentation \[page 9\]](#)

Accessing the Documentation

You can access all documentation and demonstrations for the current release of the Serena ALM suite through the Serena ALM Help system.

When you click the **Help** link in Serena ALM, the online Help appears. You can navigate the contents, search, and view the glossary for information on installing, configuring, and using the

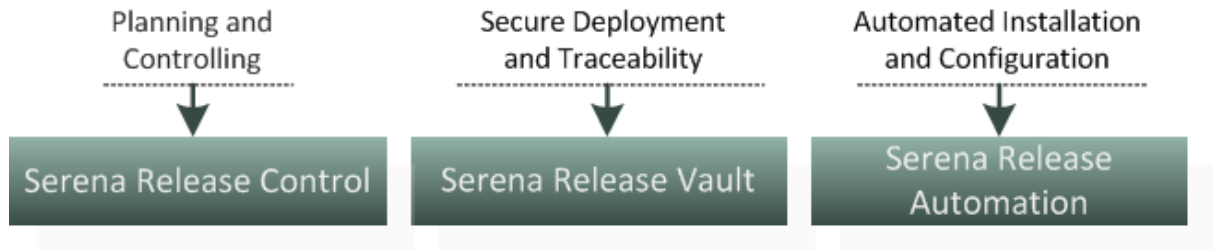
product. To view and download readme files and PDFs for all supported releases, select **Demos All Doc** in the online Help menu.

Documentation is also available on the Serena Support website.

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 Manager**, 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** and **Serena Release Automation, powered by Nolio**, automate application installation and configuration tasks.
 - Handle a high volume of tasks
 - Cut deployment time and cost
 - Reduce deployment errors
 - Handle 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*.

What is Serena Development Manager?

Serena Development Manager is a component of the Serena Application Lifecycle Management (ALM) product line. Serena Development Manager enables you to control your key software development efforts, track source code changes, define application projects, manage source code testing, and release approvals.

It consists of:

- **Serena Business Manager (SBM)**, a product that enables you to create process applications that pull and manage information and work-tracking items from different sources. It can be used to manage business processes, including access to other Web services and third-party applications.
- **Serena Development Control**, the component of Serena Development Manager that integrates SBM with the other products in the suite.

It will also include one or more the following optional components:

- **Serena Dimensions CM**, a software source control, configuration and management tool. This is the developer's primary tool. It enables you to manage your source versions and builds for Development Manager.
- **Serena ChangeMan ZMF**, a comprehensive and fully integrated solution for Software Change Management systems in z/OS environments. It provides reliable and streamlined implementation of software changes from development into production. ChangeMan ZMF manages and automates the application life cycle, protects the integrity of the code migration process, and results in higher quality delivered code to any test environment and to the production environment.
- **Serena SBM Connector for HP Quality Center / ALM**, a set of web services that is used in orchestration applications. This Orchestrated workflow also uses SBM and QC Connector web services. It enables you to use HP Quality Center / ALM for defects and to test issues.

You may also be installing the following components of the Serena ALM Suite in conjunction with Development Manager:

- **Serena Requirements Manager**, A Serena Orchestrated ALM Product that enables you to manage and approve requirements that are then used to generate development change requests in Development Manager.
- **Serena Release Manager**, a Serena Orchestrated ALM Product that enables you to plan, manage, and automate the deployment of applications into test, pre-production, and production environments across mainframe and distributed systems with start-to-finish traceability and end-to-end visibility.
- **Serena Dashboard**, a reporting tool that enables you to produce comprehensive graphs, metrics, and views of your ALM processes from different data sources. It provides a set of out-of-box graphs and metrics and enables you to customize them or create your own.

For an overview of using Serena Development Manager, see the *Serena Development Manager Getting Started Guide*.

Installation Prerequisites and Planning

This section tells you what you must do before installing and configuring Serena ALM.

[Serena Release Manager Architecture \[page 12\]](#)

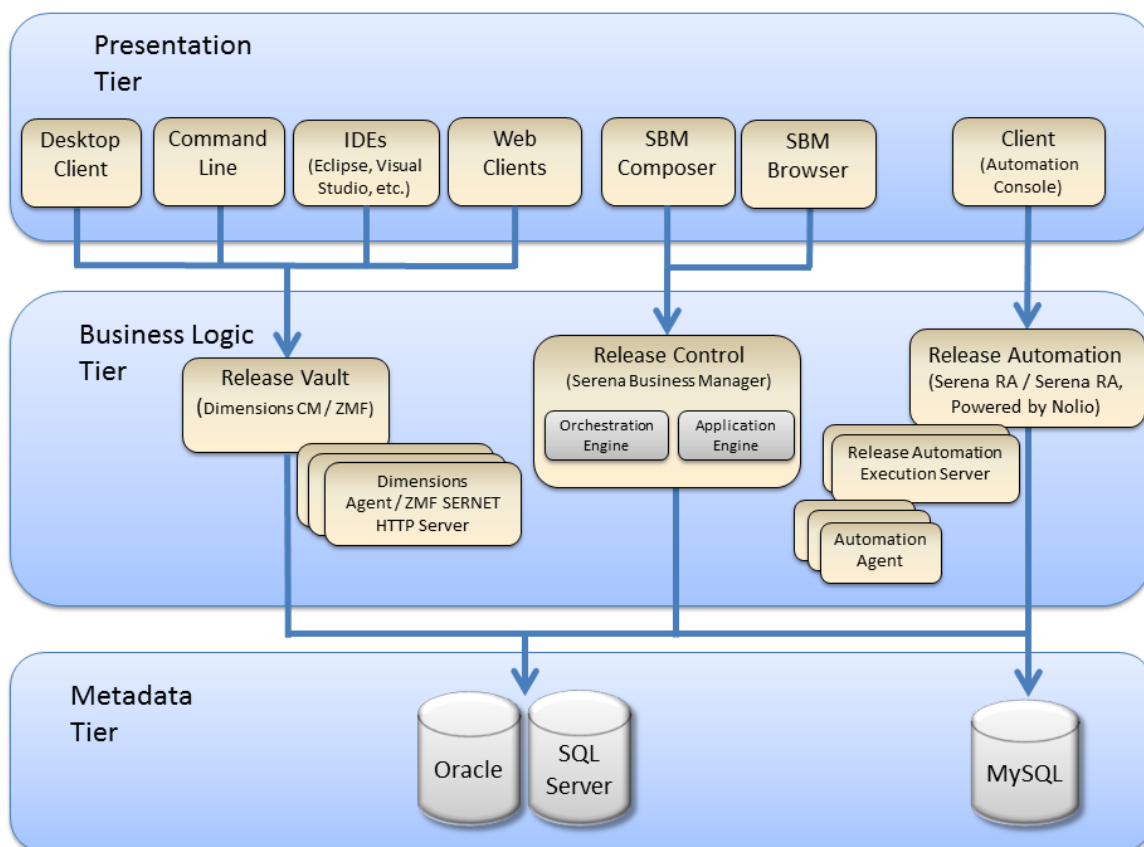
[Preparing for the Installation \[page 12\]](#)

Serena Release Manager Architecture

Through the Serena ALM user interface and its underlying SBM infrastructure, the integrated power of Release Control, Development Control, and Requirements Control are brought together.

The Serena Release Manager portion of the Serena ALM architecture is shown in the following figure. Serena Development Manager and Serena Release Manager share the SBM, Dimensions CM, and ChangeMan ZMF components. Each organization's implementation may vary depending on which Serena and non-Serena products and solutions participate as the foundation of and providers of information into the ALM processes.

For more information on the Serena ALM suite, see *Introduction to Serena ALM*.



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 \[page 13\]](#)
- [Database Requirements \[page 14\]](#)
- [System Requirements \[page 15\]](#)
- [Software Compatibility Requirements \[page 15\]](#)
- [System Requirements \[page 15\]](#)
- [Planning Checklists and Worksheets \[page 15\]](#)
- [Installation Planning for SBM Connector for HP Quality Center / ALM Integration \[page 16\]](#)
- [Installing and Configuring the Serena Dashboard \[page 16\]](#)

Installation Prerequisites

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

IMPORTANT! Serena ALM requires SSO to be enabled.

Preparing for Serena Release Manager 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 Manager**

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

- **Dimensions CM**

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

IMPORTANT! This release ships with a new version of the common tools installer. This common tools version is not backwards compatible with releases of Dimensions CM prior to 12.2.2. If your configuration uses a version of Dimensions CM prior to 12.2.2 that will be co-located with the Serena ALM installation, it is strongly recommended that you upgrade it before running the installer for this release of Serena ALM.

- **Serena License Manager**

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

- **Common Supporting Files**

The installer for Serena ALM 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.

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.

Preparing for Serena Development Manager Installation

Before you install the Serena Development 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*.

- **Connector for HP Quality Center**

If you are installing the Development Manager Connector for HP Quality Center, you will need to install:

- HP Quality Center, version 10 or later
- The Quality Center modules. For details, see *Installing and Registering Quality Center Modules in the Development Manager Connector for HP Quality Center / ALM Implementation Guide*

- **Dimensions CM**

If you are using Dimensions CM as a Configuration Management provider, Development Manager requires Serena Dimensions CM.

NOTE If upgrading from an earlier version of Dimensions CM, you may need to carry out some changes to the process model. Please contact Serena Support for details.

- **ChangeMan ZMF**

If you are using Changeman ZMF as a Configuration Management provider, Development Manager requires Changeman ZMF.

- **Serena License Manager**

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

- **Common Supporting Files**

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

Database Requirements

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

- **Serena Business Manager**

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

- **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 Microsoft SQL Server, Oracle, MySQL, and Derby.

- **Serena Release Automation, powered by Nolio**

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

NOTE For details of required database versions, see the **Supported Platforms** tab for the respective product at:

<http://support.serena.com/roadmap>

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 ALM 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 ALM	Your SBM requirements address the requirements for Serena ALM.
Serena Dimensions CM	<ul style="list-style-type: none"> See the Dimensions CM Release Plan page for the supported version. See the Dimensions CM Readme
Serena Release Automation	"System Requirements" in <i>Serena Release Automation Installation and Administration Guide</i> .
ChangeMan ZMF (Optional)	"System Requirements" in <i>Serena ChangeMan ZMF Installation Guide</i> .
Quality Center	Ensure that you are running a supported version of Quality Centre, as indicated in the Serena Release Plan. See Supported Products and Versions [page 15] .

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 [Installation and Configuration Checklist \[page 18\]](#).

Installation Planning for SBM Connector for HP Quality Center / ALM Integration

For prerequisites for the Quality Center integration, see the *Development Manager Connector for HP Quality Center / ALM Implementation Guide*. For HP Quality Center ALM consult your product documentation.

If you are using the Development Manager Connect for Quality Center, you will need to have HP Quality Center installed, and download and install the Quality Center modules on the SBM Server before running the Development Control Installer. For details, see Installing and Registering Quality Center Modules in the *Development Manager Connector for HP Quality Center / ALM Implementation Guide*. There are also some post-installation steps you will need to carry out.

Installing and Configuring Serena Requirements Manager

Serena Requirements Manager is a separately licensed option. For prerequisites, installation and configuration details, see the *Serena Requirements Manager Installation Guide*.

See [Development Manager Installation and Upgrade Scenarios \[page 147\]](#) for information on how to install and configure Serena Requirements Manager in relation to Serena Development Manager.

Installing and Configuring the Serena Dashboard

The Serena Dashboard is a separately licensed option. For pre-requisites, installation and configuration details, see the *Serena Dashboard Installation and Configuration Guide*

Serena ALM Installation

This section leads you through an installation of all of the components of Serena ALM.

[Installation Overview \[page 17\]](#)

[Installation Order \[page 17\]](#)

[Installation and Configuration Checklist \[page 18\]](#)

[Before Running the Installer \[page 20\]](#)

[Running the Installer \[page 22\]](#)

[Importing and Promoting Serena ALM \[page 26\]](#)

Installation Overview

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

You should install or verify existing installation settings for each of the systems that will participate in your Serena ALM implementation. After you have completed installing or verifying each of the systems, you must complete the system configuration to make Serena ALM ready to use.

Serena Release Manager components include:

- SBM
- Serena Release Control
- A Serena release vault, such as Dimensions CM or ChangeMan ZMF
- Serena Release Automation

Serena Development Manager components include:

- SBM
- Development Control
- A configuration management system, such as Dimensions CM or ChangeMan ZMF
- HP Quality Center

Installation Order

The installation should proceed in the following order given unless you have the experience and necessity to do otherwise.

If you already have an installation of the systems that you plan to use with Serena ALM, you do not need to install a new system. However, you must ensure that you are running a supported version as indicated on the Serena Support website roadmap and download pages for your version of Serena ALM.

1. **Serena License Manager** must be installed and configured so that you can license and run Serena Business Manager and Dimensions CM. For evaluation purposes, you can use a temporary license in lieu of licensing through the Serena License Manager.

2. If you plan to use a **a shared database**, install it before installing the rest of the products. For example, for evaluation purposes, you could install the Serena Oracle runtime database provided with Serena Dimensions CM. Point each product in the suite to this shared database as you install them.
3. **Serena Business Manager** must be installed and configured before you can install Serena Development Control or Serena Release Manager. You must use Single Sign On (SSO) for Serena Release Manager, and you should install SBM and the SSO server. Before you install Serena Business Manager, please see the “Pre-installation Checklist” section of the *Serena Business Manager Installation and Configuration Guide*.
4. **Serena Dimensions CM** must be installed and configured before you can install Serena Development Control and before you can use Serena Release Manager functions that rely on Dimensions CM. 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) during the installation of Dimensions CM and point Dimensions CM to the SSO server you installed for SBM.

5. **Serena ALM** must be installed using the provided installer. The Serena ALM installer installs Serena Development Manager, Serena Release Manager, and Serena Requirements Manager components.

Installation and Configuration Checklist

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

Step	Action
1: Installation	<p>Complete the prerequisites for the Serena ALM installation and then run the Serena ALM installer.</p> <p>See:</p> <ul style="list-style-type: none"> • Before Running the Installer [page 20] • Running the Installer [page 22]
3: Process Apps	<p>Import the Serena ALM SBM solution and promote the snapshots of the process apps.</p> <ul style="list-style-type: none"> • Import the Serena ALM solution, which contains snapshots of the process apps and other supporting data. • 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. <p>See Importing and Promoting Serena ALM [page 26].</p>

4: Serena ALM Process Apps	<p>Configure the process apps in SBM for Serena Release Manager and/or Serena Development Manager.</p> <ul style="list-style-type: none"> • If needed, select the SSO Authentication option in forms with RESTgrid widgets. • Publish and deploy the process apps to upgrade the structure of the RESTGrid widgets. <p>See Configuring the Release Manager Process Applications [page 37] and/or Configuring the Development Manager Process Applications [page 39].</p>
5: Serena ALM System Configuration	<p>Configure Serena ALM licensing and service or administrative users as follows:</p> <ul style="list-style-type: none"> • Apply licenses as needed. • Create a service or administrative user and set all privileges for that user to the Serena ALM objects, such as projects, reports, and auxiliary tables. For example, almadmin. • Enable roles for Serena ALM projects and verify that Serena ALM is activated. • Configure required objects in SBM for Serena Release Manager and/or Serena Development Manager. These may include but aren't limited to the following: <ul style="list-style-type: none"> ◦ SBM users, reports, and notifications ◦ Release Manager applications and environments (servers) (specific to your organization; add to the respective auxiliary tables) <p>See Serena ALM Configuration [page 48].</p>
6: Dimensions CM for Serena ALM	<p>Configure Dimensions CM communication for both Serena Release Manager and Serena Development Manager.</p> <p>See Dimensions CM for Serena ALM Configuration [page 78].</p>
7: ZMF for Serena ALM	<p>Configure ChangeMan ZMF communication for both Serena Release Manager and Serena Development Manager.</p> <p>See ChangeMan ZMF for Serena ALM Configuration [page 93].</p>
8: Serena Release Automation for Release Manager	<p>Configure Serena Release Automation communication for Serena Release Manager.</p> <p>See Release Automation for Serena ALM Configuration [page 109].</p>

**9: Providers for
Serena ALM**

Configure the providers as needed. This may include but isn't limited to the following configuration:

- For Serena Release Manager:
 - Serena Service Manager (SSM) request for change (RFC) properties (for example: `itsm.properties` and `providers.properties`)
 - Serena Business Manager (SBM) business change request (BCR) properties (for example: `bcr.properties` and `providers.properties`)
 - Serena Business Manager (SBM) development change request (DCR) properties (for example: `sbm-issues.properties` and `providers.properties`)
 - Dimensions CM deployment unit (DU) properties (for example: `dm_qlarius.properties` and `providers.properties`)
 - ChangeMan ZMF deployment unit (DU) properties (for example: `rlmzmf_packages.properties` and `providers.properties`)
 - Objects in SBM and SSM process apps if either or both are used to provide RFC, BCR, or DCR information for Release Manager.
- For Serena Development Manager:
 - Dimensions CM development unit properties (for example: `dm_qlarius.properties` and `providers.properties`)
 - ChangeMan ZMF development unit properties (for example: `dvmzmf_packages.properties` and `providers.properties`)

See [Provider Configuration \[page 121\]](#) for details.

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

TIP Clear your browser cache if Serena ALM has been run from your browser on this machine before to ensure that your user interface is displaying the most current shell elements. See [Clearing the Web Browser Cache \[page 193\]](#)

Before Running the Installer

Before running the ALM Suite installer to install Development manager, please read the following sections:

- [Installation Issues \[page 199\]](#)
- [Upgrading Issues \[page 202\]](#)
- [Development Manager Installation and Upgrade Scenarios \[page 147\]](#)

Upgrading

If you are upgrading from an existing version of Serena Release Manager, see [Release Manager Upgrade Scenarios \[page 140\]](#).

If you are upgrading an existing version of Development Manager or Requirements Manager, run the ALM Suite installer and refer to [Development Manager Installation and Upgrade Scenarios](#). [page 147]

IMPORTANT! If upgrading from ALM 4.0 to ALM 4.0.1.

Modify the registry before running the installer. If the following registry hive is missing:

HKLM\Software\Serena\Dimensions\12.2.0\Install\

and the following hive exists:

HKLM\Software\Serena\Dimensions\12.1.0\Install\

Use export & import to "copy" the hive from 12.1.0 to 12.2.0:

1. Export HKLM\Software\Serena\Dimensions\12.1.0\
2. Open the resulting REG file in a text editor, such as Notepad.

Replace all "12.1.0" entries with "12.2.0" and save the changes.

Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Serena\Dimensions\12.2.0]

[HKEY_LOCAL_MACHINE\SOFTWARE\Serena\Dimensions\12.2.0\Install]

"INSTALL_JavaHome"="C:\\Program
Files\\Serena\\Solutions\\..\\..\\Common\\jre\\6.0"

"INSTALL_TomCatDir"="C:\\Program
Files\\Serena\\Solutions\\..\\..\\Common\\tomcat\\6.0"

"INSTALL_CTVersion"="1.4.1.0"

3. Import this edited REG file into the Windows Registry.

This will result in an additional hive with "12.2.0".

HP Quality Center

If you are using the Development Manager Connect for Quality Center, you will need to have HP Quality Center installed, and download and install the Quality Center modules on the SBM Server before running the ALM Suite Installer. For details, see Installing and Registering Quality Center Modules in the *Development Manager Connector for HP Quality Center / ALM Implementation Guide*. There are also some post-installation steps you will need to carry out.

You will also need to select a custom install and select the Quality Center Connector, as this option is not selected by default.

Remote Desktop Installation

When installing Development Control via Remote Desktop access to a Windows 2003 system, installation may fail if you include the Connector to Quality Center option. To avoid this issue, run Remote Desktop in Console or Administration mode by running one of the following commands:

```
mstsc /admin
```

or

```
mstsc /console
```

Specifying Configurator SSO Support in the ALM Installer

It is advisable to only select the option Configurator SSO Support and configure the SSO details on the first occasion that you run the ALM Suite Installer. When you run the installer on subsequent occasions, you should choose a custom install and deselect this option. If you configure these details more than once, SSO may not function correctly

Also you should not use *localhost* for the server name but specify the name of the actual server.

Running the Installer

To install Serena ALM:

1. Download the full installation package from the Serena Support Web site, My Downloads page, for example ALM4.0.1_b0186_win32.zip or ALM4.0.1_b0186_win64.zip. Extract the files.
2. On the SBM Server, run the Serena ALM installer executable, ALM.exe or ALM 64-bit.exe depending on your server type.

You may be asked to install Windows Installer 4.5, in which case:

- a. Click **Install**.
- b. When asked if you want to reboot your machine, reply **Yes**.

The **Serena ALM Welcome** page appears.

3. Click **Next**.

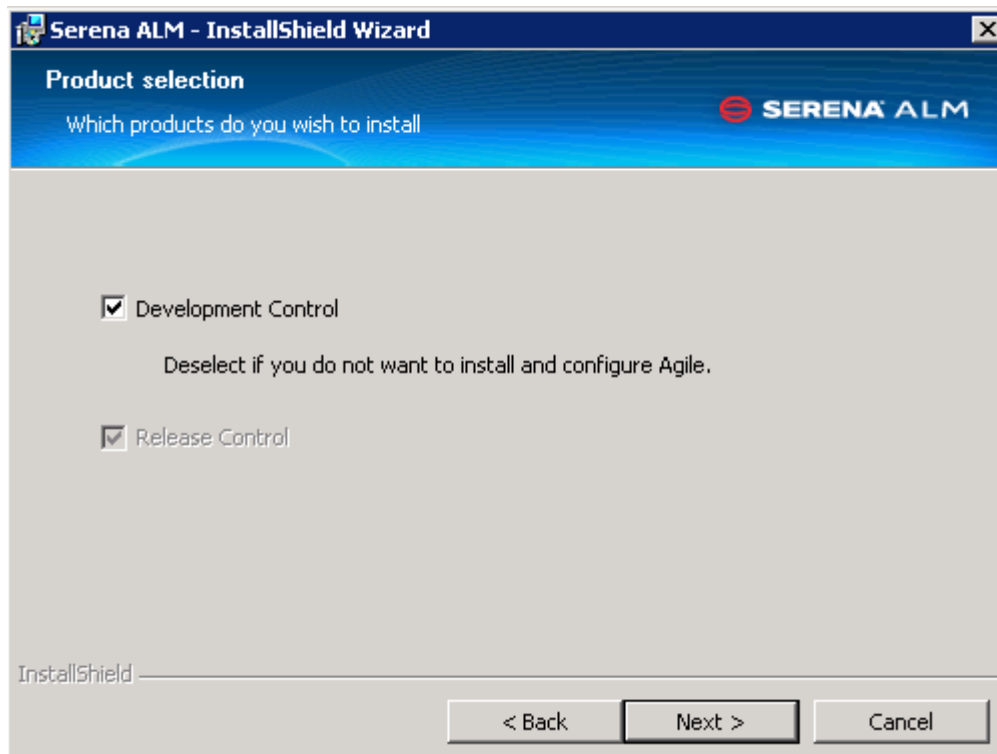
The **License Agreement** page appears.

4. Confirm and click **Next**.

The **Destination Folder** page appears.

5. Optionally click **Change** to change the target location for the installation.
6. Click **Next**.

The **Product selection** page appears.



7. If you want to install the Agile features of Serena Development Control, select **Development Control**. Selecting this option means that the shell and database schema for the Agile features of Development Manager will be installed. (Note that you cannot deselect **Release Control**. Whether or not you have access to Release Control will be determined by which process apps you select, as described in [Which Process Apps Should I Promote and Deploy \[page 29\]](#).)
8. Click **Next**.

The **Custom Setup** page appears. The options shown on this page depend on whether or not Dimensions CM is installed on the same sever as SBM.

9. Click to select or deselect components to install. An X on the installation tree entry icon indicates the component is not selected. To select any deselected components, expand the entry and select one of the feature installation options.

- **Agile Configuration:** Install & configure the agile component of Development Manager.

NOTE If you selected Development Control on the previous page, the option **Agile Configuration** will be selected, but you can deselect it. If you did **not** select Development Manager, this option will be deselected and you will not be able to select it.

- **ALM Foundation Services:** Common services to enable the Serena ALM solution.
 - **Configurator SSO Support:** Configure the connection to the Serena Single Sign-on server to enable login to the Configurator through SSO.
 - **Connectors | Quality Center Connector:** Integrate Development Control with external tools such as Quality Center.
 - **Providers:** Common Framework for all ALM Suites for configuration of supported Serena and non-Serena tools.
- **ALM Workbench:** Common role-based User Interface across the ALM Suites to optimize user experience.
- **ALM Process Application:** SBM-based process application snapshot. The snapshot must be promoted to an SBM runtime environment.

10. Click **Next**.

The **Configuration Details** page appears.

The installer indicates whether a valid version of SBM and the Serena common Tomcat Web server are installed. It installs Tomcat if needed.

NOTE The Serena ALM 4.0.1 installation includes a new version of the Serena Common Tomcat Web server. This version requires Dimensions CM 12.2.2 and is not backwards compatible with previous releases of Dimensions CM.

If your configuration co-locates Serena ALM with Dimensions CM 12.2.1.1 or older, Dimensions CM must be upgraded before installing Serena ALM 4.0.1.

The Serena ALM 4.0.1 installer will upgrade the Serena Common Tomcat Web server if the older version is detected. This upgrade moves the pre-existing Tomcat files into a backup location, tomcat6_pre4.0.1, and then installs the new common tools.

11. Click **Next**.

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

CAUTION! You should configure the Configurator SSO Support only once. If you run the installer again to install additional options, you must deselect this option. Otherwise, SSO may not function correctly. Also you should not use localhost for the server name but should specify the name of the actual server.

- a. Specify the host name and port for an existing installation of a Serena SSO server.
 - b. If you want the connection to use HTTPS, select **Secure (HTTPS) Connection**.
 - c. Click **Next**.
12. If you selected the **Agile Configuration** option, the **Connection details** page appears.

The screenshot shows the 'Serena ALM - InstallShield Wizard' window with the 'Connection details' tab selected. The title bar reads 'Serena ALM - InstallShield Wizard'. Below the title bar, the tab is labeled 'Connection details' and the subtitle is 'Details of SBM Database Server'. The 'SERENA ALM' logo is in the top right corner. The main area contains the following fields and controls:

- Database Server type:** Two radio buttons, 'Oracle' (selected) and 'SQL Server'.
- Host:** Text box containing 'sbm_dbhost'.
- Oracle SID:** Text box containing 'dim12'.
- Oracle Home:** Text box containing 'c:\oracle\ora11g'.
- SBM Database user:** Text box containing 'sample'.
- Password:** Text box with masked characters (dots).
- Test Connection:** A button located at the bottom right of the main area.
- Navigation buttons:** At the bottom of the window are three buttons: '< Back', 'Next >', and 'Cancel'.

- a. Enter the details of an existing SBM database in which to install the database schema for the Agile component of Development Manager.

- b. Click **Test Connection**.

If you receive a message that you successfully connected to the database, the **Next** button activates.

- c. Click **Next**.

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

13. Click **Install**.

The **Installing Serena ALM** page appears informing you files are being copied.

If you have chosen to install the Quality Centre connector, after a short time, a **Welcome to the InstallShield Wizard for Serena Connector for HP Quality Centre** appears.

- a. Click **Next**.
- b. Accept the License agreement and click **Next**.
- c. Select the version of SBM that you have installed.
- d. Click **Next**.
- e. On the **Ready to Install Program** page, click **Install**.
- f. Some files are copied and then an **InstallShield Wizard Completed** page appears.
- g. Click **Finish**.

The **Installshield Wizard Completed** page appears with an installation summary.

This informs you what components have been installed for Serena ALM and their location.

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

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

%Temp%\Install_SUITE_comp.log or

%Temp%\Install_SUITE_comp64.log

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

..\Program Files\Serena\Solutions

com.serena.alm.sbm.shell.4.0.1.zip

\solution (contains the solution file)

\war (contains the war files)

The installer does the following automatically:

- Copies the war files to the Tomcat Web server and restarts Tomcat.
- Puts the user interface files in the appropriate SBM folder and imports them into the database.
- Copies the ALM Solution pack solution file to the solutions folder for the Serena Common JBoss.

15. To check that ALM is installed into Tomcat, enter the URL:

http://<servername>:8080/alm/config/alm

in a web browser.

This should display the login page, or if you are already logged in, the ALM Configurator.

You can also enter the URL:

`http://<servername>:8080/almopd/static/`

in a web browser to check the ALM Foundation Services for Products and Components.

Importing and Promoting Serena ALM

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

NOTE It is a prerequisite for SBM to be installed with Single Sign-on (SSO). 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.

PRIVILEGES The SBM user you are using for the import and promotion tasks must have privileges to deploy and undeploy applications, import solutions, and view and edit the Serena ALM process applications. Use the SBM Application Administrator and the SBM Application Repository to grant these privileges. The user *admin* in the SBM Sample database has these privileges by default.

"Privilege Examples for Publishing and Deploying" in the *Serena Business Manager Application Repository Guide* gives an overview of the privileges needed to work with process apps.

Importing the Serena ALM Solution

The Serena Release Manager application is part of the Serena ALM suite, which is packaged as an SBM solution. Included in the solution are the following:

- Process apps for Serena ALM
- 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 SBM 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 as available. For example:

ALM_Solution_Pack 4.0.1.0.186

5. Click **Import**.

The **Import Solution** dialog box appears, explaining which snapshots 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**. The following process app snapshots should be listed:

- ALM Projects
- Application Release
- Deployment
- Dev Change Requests
- Dev Control Orchestrations
- Dev Package
- Dev Tasks
- Environment
- Global Process App (eval)
- Products and Components
- Quality Center Connector
- Release Package
- Release Train
- ReleaseTemplate
- Requirements Approval Management
- RLM_AUX
- RM to RQM Connector
- Vault_Request

Creating an Environment for Serena ALM

If you don't have a suitable existing SBM application engine server environment to promote the process applications into, you should create a new environment. After you create the environment, you must set up a BPEL engine and SBM event manager for that environment.

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

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

If you are putting your Serena ALM process apps into an existing environment, continue to [Which Process Apps Should I Promote and Deploy? \[page 29\]](#).

Create the environment and supporting target servers as follows:

- [Creating the Environment \[page 28\]](#)
- [Specifying the BPEL Server for SBM Orchestrations \[page 28\]](#)
- [Specifying the SBM System Event Manager \[page 29\]](#)

Creating the Environment

To create the environment:

1. From the SBM Application Repository Solutions content pane, click **Environments**.
2. Click **New**.

The **New Environment** dialog box appears.

3. Fill out the form for the new environment as follows:
 - a. Enter a name and description. For example, ALM Environment.
 - b. In the **Composer** field, select **Enable Deployment**.
 - c. Under **Application Engine Server**, enter a name and description. For example: RLM Application Engine Server.
 - d. In the URL, change the server to the host name for your application engine server and specify the port number for the server. For example:

`http://sbmaehost:80/gsoap/gsoap_ssl.dll?sbmininternalservices72`

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

Specifying the BPEL Server for SBM Orchestrations

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

To set up the target server for SBM orchestrations:

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

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

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

- a. In the **Type** field, select **BPEL Server (JBPM)**.
- b. Name the BPEL server. For example: SBM Orchestration Server.
- c. Enter the URL using the host name for your orchestration engine server as the hostname.

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

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

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

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

Specifying the SBM System Event Manager

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

To set up the SBM system event manager:

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

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

3. Fill out the form to create the event manager server as follows:
 - a. In the **Type** field, select **System Event Manager**.
 - b. Name the event manager server. For example: SBM Event Manager.
 - c. Enter the URL using the host name for your event manager host as the hostname. For example:
`http://EventManagerHost:8085/eventmanager/services/ALFAdmin`
 - d. Click **Test Connection** to test the connection.
 - e. Click **OK**.

NOTE

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

Which Snapshots Should I Promote?

The following sections describe how to promote and deploy the required process apps for the implementation of Serena ALM that you will be using.

Because there are shared process apps between Release Manager, Development Manager, and Requirements Manager, you will not necessarily promote and deploy all of the process app snapshots contained in the solution file. The tables below indicate which ones are required for your combination of the orchestrations. If you are using Development Manager, refer to the

Development Manager table. If you are using Release Manager, refer to the Release Manager table. Refer to both tables if using Release Manager and Development Manager.

Release Manager

Solution	Required	Process App	Notes
RLM, DVM, RQM	X	ALM Projects	
RLM	X	Release Train	
RLM	X	Application Release	
RLM	X	Release Package	
RLM	X	Deployment	
RLM	X	Release Template	
RLM	X	Environment	
RLM, DVM	X	RLM AUX	
RLM, DVM	X	Vault Request	

KEY:

X = Required

Solution:

- RLM = Release Manager
- RQM = Requirements Manager
- DVM = Development Manager

Development Manager

Solution	Required	Process App	Notes
DVM, RLM, RQM	X	ALM Projects	
DVM	X	Dev Packages	
DVM	X	Dev Change Requests	3, 4
DVM	X	Dev Tasks	
DVM	X	Dev Control Orchestrations	2,3,4
DVM + HP QC		Quality Center Connector	
DVM, RLM	X	Vault Request	

DVM, RLM	X (shell)	RLM AUX	
DVM	X	Global Process App	1
RQM	X	Requirements Approval Management	
RQM		RM to RQM Connector	
OPD	X	Products and Components	

KEY:

X = Required

Solution:

- RLM = Release Manager
- RQM = Requirements Manager
- DVM = Development Manager
- HP QC = HP Quality Center integration. For details, see the *Serena Development Manager Connector for HP Quality Center Guide*

NOTES:

1. The Global Process App imports sample Groups and Users to the new environment. If you do not want these users and groups, you can exclude this option when you promote this process app. For details, see [Promoting the Development Manager Snapshots \[page 33\]](#).
2. You must create the ReleaseRequestService endpoint before promoting this snapshot. See [Updating the Development Manager Endpoints \[page 39\]](#).
3. ALM Projects must be promoted before these snapshots because they depend upon the DVMUtilService endpoint that is created by ALM Projects.
4. You must manually select the DVMUtilService endpoint when promoting these snapshots.

Consult the *Requirements Manager Installation Guide* if you are configuring Requirements Manager in addition to Development Manager or Release Manager.

Promoting the Release Manager Snapshots

You must promote each of the snapshots required for RLM according to the table in [Which Snapshots Should I Promote? \[page 29\]](#).

When promoting the snapshots, make sure to create and select the endpoints as needed, and make sure the endpoints are authenticated with Security Token.

To promote a snapshot:

1. In the SBM Application Repository navigation pane, click **Solutions** and then select the **Solutions** tab.
2. Select the solution and click **Open Snapshots**.
The list of snapshots appears.
3. Select a snapshot that you have not yet promoted and click **Promote**.

The **Summary** page appears.

4. Click the **Destination** field.

The **Destination** page appears.

5. Select the environment that you created for Serena ALM.
6. Click **Next**.

The **Entities** page appears.

7. Click **Next**.

The **Mapping** page appears.

8. Select any Source that does not have a Destination Endpoint and click the **Choose Destination Endpoint** button.
9. Select the endpoint from the list, or if the Destination Endpoint has not yet been defined for the selected Source, click **Create a new endpoint**.

To create and select an endpoint:

- a. Name the endpoint.

TIP Name the new endpoint the same as the Source Endpoint for which it is being created for ease of identification when selecting it as the Destination Endpoint later.

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

`http://<hostname>:<tomcat port>/alm/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 alm/services directory of the common Tomcat Web server.

Example endpoints with the default port number are shown in the following table.

Source	Destination Endpoint
RLMUtilService	http://almhost:8080/alm/services/RLMUtilService
ReleaseRequestService	http://almhost:8080/alm/services/ ReleaseRequestService

- c. In the **Authentication** field, select **Security Token**.
 - d. Click **Test Connection** to test the connection.
 - e. Click **OK**.
 - f. Select the Source Endpoint again and click **Choose Destination Endpoint**. Select the endpoint you just created.
10. Click **Done**.

The **Summary** page appears.

11. Click **Promote**.

The **Promotion Started** page appears.

12. Click **View Log** or **Show Activities** to see results.
13. Repeat the procedure for each snapshot.

PRIVILEGE Privileges for deleting or modifying the Serena ALM 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.

Promoting the Development Manager Snapshots

You must promote each of the snapshots required for DVM according to the table in [Which Snapshots Should I Promote? \[page 29\]](#).

IMPORTANT!

- You should promote ALM Projects first, because it creates the DVMUtilService endpoint that is required for Dev Control Orchestrations and Dev Change Requests.
- When promoting Dev Control Orchestrations and Dev Change Requests DVMUtilService endpoint is not selected by default, so it is necessary to select it on the Mappings page. See [Step 7 \[page 35\]](#).
- You must promote the RQM Requirements Approval Management Snapshot to create all necessary reports for this release of DVM.

To promote a snapshot:

1. Click **Solutions** in the navigation pane.
2. Select the solution and click **Open Snapshots**.

The list of snapshots appears.


3. Select a process app and click **Promote**.


The **Summary** page appears.

4. Click on the **Destinations** field.

Summary

This shows the choices you have made. Verify each item is correct before continuing. Select an item to change it.

Source: Dev Control Orchestrations 

Destination: - 

Entities: Please select a source and destination first

Mappings: Please select a source and destination first

The **Destinations** page appears.

5. Select the environment you created above.
6. Click **Next**.

The **Entities** page appears.

Promote Help

Entities Dev Change Requests (saved) from Development_Manager_Solution 1.2.0.31.0

Select the entities to promote to the destination.

Applications

- Dev Change Requests

Entities of 'Dev Change Requests'

Projects: All

Notifications: All

Notification Rules: Referenced

Reports: All

Auxiliary Table Data: All

Transition Group Restrictions: ☒

User Field Selections: ☒

☒ Merge conflicts

Return to Summary < Previous Next > Cancel

- Make sure the option **Merge Conflicts** is selected.
- For the *Dev Change Request* process app, if you are upgrading a previous version of Development Manager:


This has a predefined project that is created as an example. This project could cause a conflict during promotion, so if you do not need this project, or the promotion fails because of this project, you should select **None** for **Projects** for this process app. For details, see Error Promoting Dev Change Requests Snapshot (DEF217760).

- For the *Global Process App (eval)* only, If you do not want to import the sample groups and users into your environment, select:

Groups: None

Users: None

Promote

Help 

Entities Global Process App (eval) (saved) from Development_Manager_Solution 1.2.0.31.0

Select the entities to promote to the destination.

Applications

Global Entities

Global Entities

Groups:

None

Notifications:

All

Notification Rules:

Referenced

Reports:

All

Folders:

All

Users:

None

Auxiliary Table Data:

All

User Field Selections:

☒

☒ Merge conflicts

7. Click **Next**.

The Mapping page appears.

NOTE For the *Dev Control Orchestrations* snapshot:

- Make sure that **Authentication** is set to *Security Token* for the endpoints *DVMUtilService* and *VaultService*
 1. Select an endpoint and click **Edit**.
 2. In the **Authentication** field, select **Security Token**.
 3. Click **OK**.
- Make sure the correct endpoint is selected if you have already installed Requirements Manager:
 1. Click the **Choose Destination Endpoints** button.
 2. Select *rtmService* (not *rtmService_2*)

NOTE Some endpoints may be missing and need to be selected. See the issue Missing Endpoints (DEF221077, DEF225893).

8. Click **Done**.

The **Summary** page appears.

Promote Help ?

Summary
This shows the choices you have made. Verify each item is correct before continuing. Select an item to change it.

Source: CMConcept ✓

Destination: Default Environment ✓

Entities: No entities ✓

Mappings: All 3 process app endpoints are mapped ✓

Settings
Stop Promotion: if any error occurs ▼
☐ Verify endpoints before continuing
☐ Send email notification

[Schedule?](#)

Promote **Cancel**

Note: Make sure **Verify endpoints before continuing** is **not** checked.

9. Click **Promote**.

The **Promotion Started** page appears.

10. Click **Show Activities** or **View Log** to see the results.

NOTE When promoting the process apps, you may see *Unable to create field* warning messages. For example:

```
WARN -- Unable to create field 'alm_Originating Baseline' in table 'Dev Packages' because relational table 'Related Vault Baselines' is not defined.
```

Such messages will not affect the success of the promotions or the functioning of Development Manager, and can be ignored.

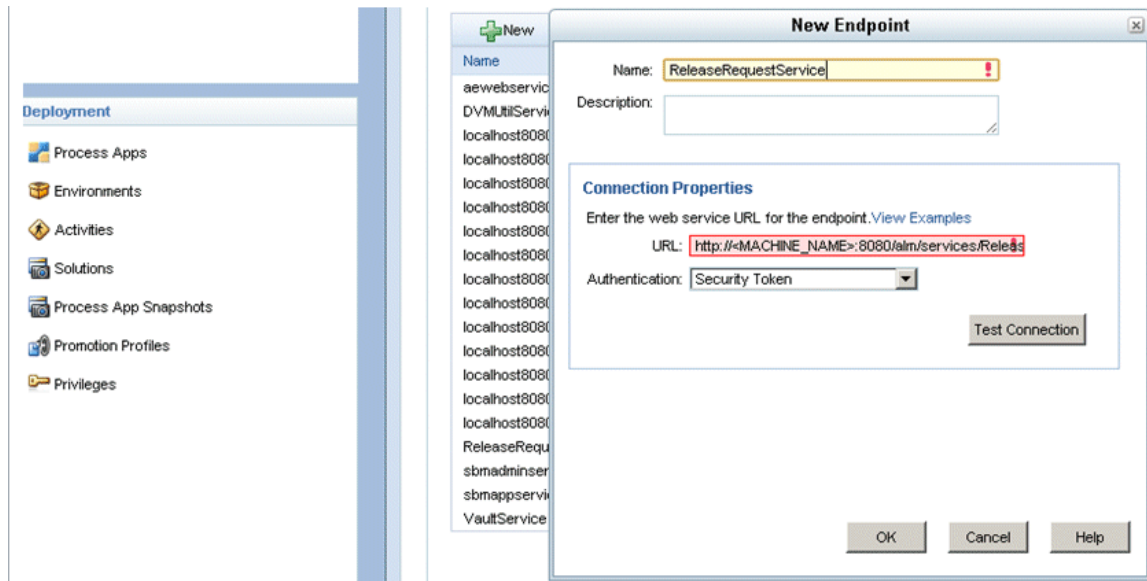
11. Repeat [Step 3 \[page 33\]](#) through [Step 10 \[page 36\]](#) for each snapshot.

Creating the ReleaseRequestService Endpoint

To create the ReleaseRequestService endpoint:

1. Open the ALM Configurator and select the **List Services** tab.
2. Note the URL for the ReleaseRequestService endpoint.

3. Open SBM Application Repository and navigate to Environments | <Your Environment> | Endpoint.
4. Create the ReleaseRequestService endpoint using the URL you noted above and selecting **Security Token** for **Authentication**.



Configuring the Release Manager Process Applications

This section guides you through configuring the SBM process apps for Serena Release Manager.

Setting Authentication for Added Endpoints

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

To update the automatically-generated endpoints:

1. Select the ALM Environment and then select the **Endpoints** tab.

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

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

Promoting the Snapshots Again to Resolve References

To resolve warnings for unresolved references in the snapshots, you must promote each of them again. Follow the steps in [Promoting the Release Manager Snapshots \[page 31\]](#), except you won't need to create and choose endpoints this time.

Configuring SSO in RESTgrid Widgets

If they are not already set automatically, you must select the **Use SSO authentication** check box for each RESTgrid widget to enable the security token to be included in the URL.

Serena Release Manager RESTgrid Widgets to Update

The form controls in which you must update the check box for Serena Release Manager are shown in the following table. The rest have the **Use SSO Authentication** option selected by default. 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 that uses RESTgrid widgets to see if **Use SSO Authentication** is selected.

Process App	Forms	Controls
Deployment	newAutomationTask, newAutomationTemplate, editAutomationTask, editAutomationTemplate, retryAutomationTask	gridApps, gridEnv, gridProc
	newReleaseAutomationTask, newReleaseAutomationTemplate, editReleaseAutomationTask, editReleaseAutomationTemplate, retryReleaseAutomationTask	gridApp, gridEnv, gridProc, gridComp

Publishing and Deploying the Release Manager Process Apps

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

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

To publish and deploy the Release Manager process apps in SBM:

For each of the process apps:

- Application Release
- Deployment
- Environment
- Release Package
- ReleaseTemplate
- Release Train
- Vault_Request
- RLM_AUX

1. In SBM Composer, do the following:

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

Documentation References

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

Configuring the Development Manager Process Applications

This section describes how to configure an out-of-the-box default example. The details may be different depending on how you are customizing Development Manager for your processes. When you are updating the process apps you need to check out the elements you are updating, and check them back in and deploy the process apps after making the updates.

Related Topics:

[Updating the Endpoints \[page 39\]](#)

[Updating the Connection Details in the Workflows \[page 41\]](#)

[Republishing the Process Apps \[page 44\]](#)

[Deploying the Development Manager Process Apps \[page 44\]](#)

Updating the Endpoints

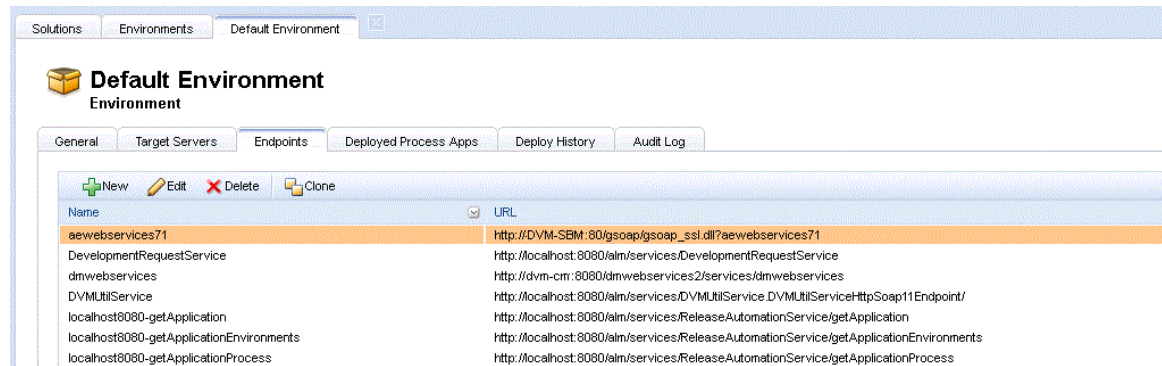
Some endpoints may need to be changed to reference the server name and port number you are using. Also, SOAP services that use RESTgrid widgets need to be updated to use Security Token authentication, so that SSO is used for these widgets.

NOTE Some other endpoints may need to be updated. See the issue [Missing Endpoints \(DEF221077, DEF225893\) \[page 211\]](#).

1. From the SBM Application Administrator navigation pane, click Environments.
2. Select the environment you used for the promotions.
3. Select the Endpoints tab.

4. Select aewebsservices71 and click the Edit button.

IMPORTANT! Ensure that you are deploying the original endpoint and not a duplicate. Duplicates will have an underscore and number appended to the end (for example rtmService_2)



5. Change the server and port information in the URL field to specify your SBM server.
6. In the **Authentication** field, select **Security Token**.
7. Click **OK**.
8. Repeat the same steps for
 - sbmadminservices72
 - sbmappsservices72
 - qcbug (if you are using Quality Center)

Deploy the Process Apps

Whenever you update an endpoint, you need to deploy the corresponding process apps. These are:

- aewebsservices71:
 - Dev Control Orchestrations
 - Dev Change Requests
 - Dev Tasks
 - Dev Packages
- ReleaseRequestService:
 - Dev Control Orchestrations
- DVMUtilService:
 - Dev Control Orchestrations
 - Dev Change Requests
 - ALM Projects
- VaultService:
 - Dev Control Orchestrations
- qcbug:

- QC Connector

Updating the Connection Details in the Workflows

The connection details for SBM in various places in the process apps may need to be updated to reference the server and login credentials that you will be using.

The existing settings are:

- User name: serviceuser
- Password: serviceuserpassword
- Host or Sever name: *localhost*, or blank (defaults to *localhost*)

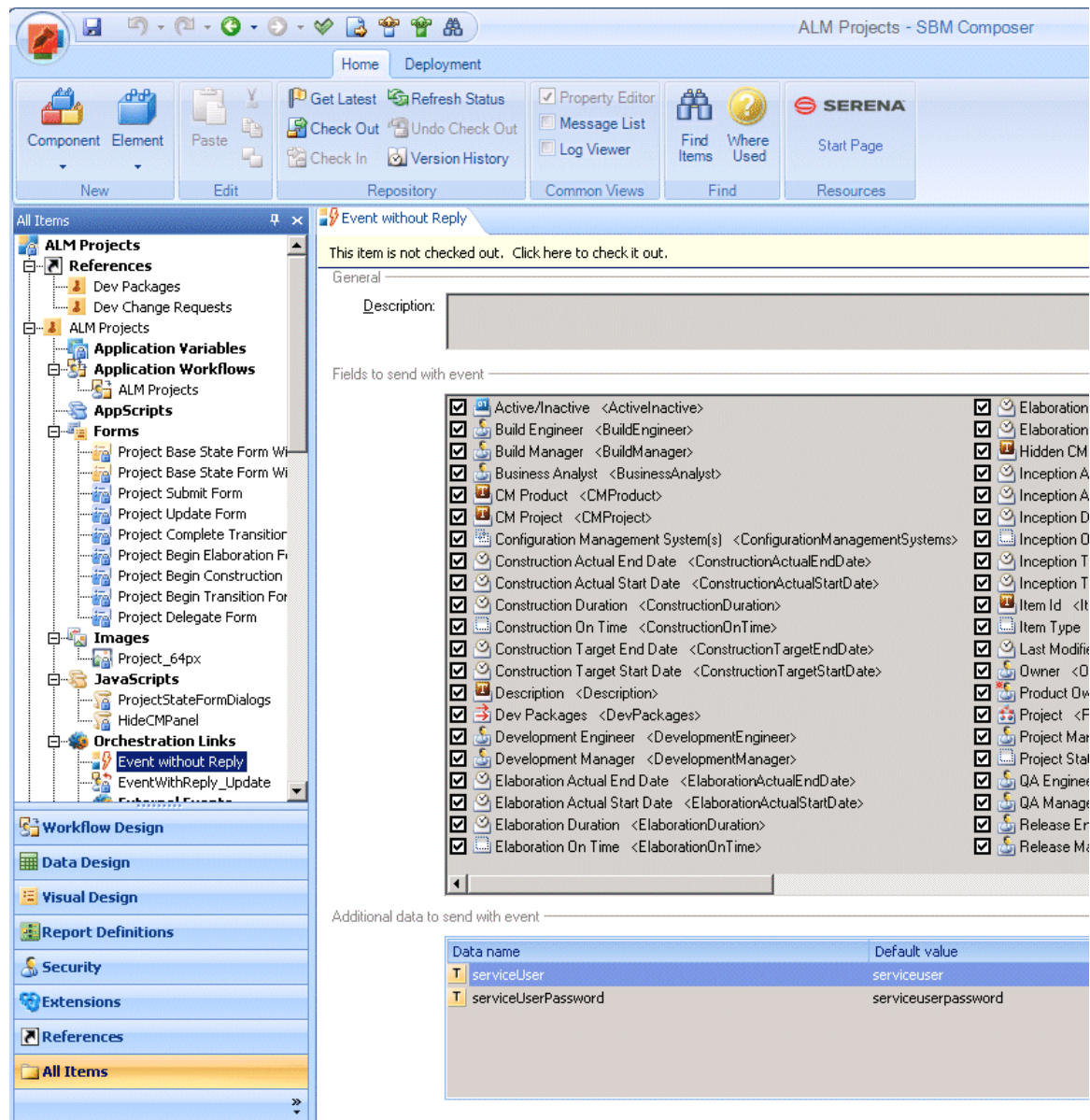
TIP If you are using the default username and password, for example for a test or evaluation configuration, you don't need to do this procedure.

The values that you will need to enter are the user name and password for the connection user if these are different, and the name of the SBM server if this is not *localhost*.

You need to update the **Event without Reply** orchestration links and some of the orchestration workflows.

To update the Orchestration Links

1. Open SBM Composer.
2. Open a process app, for example **ALM Projects**.
3. In the Navigation pane, expand **All Items**.
4. Under **Orchestration Links**, select **Event without Reply**.



5. Edit the **Additional data to send with event** to change the **serviceUser** and **serviceUserPassword** to the values you are using
6. Repeat this for each of the orchestration workflows in each process app where there are connection details and each **Event without Reply** orchestration link that is present. Note that the fields appearing may be different for different workflows.

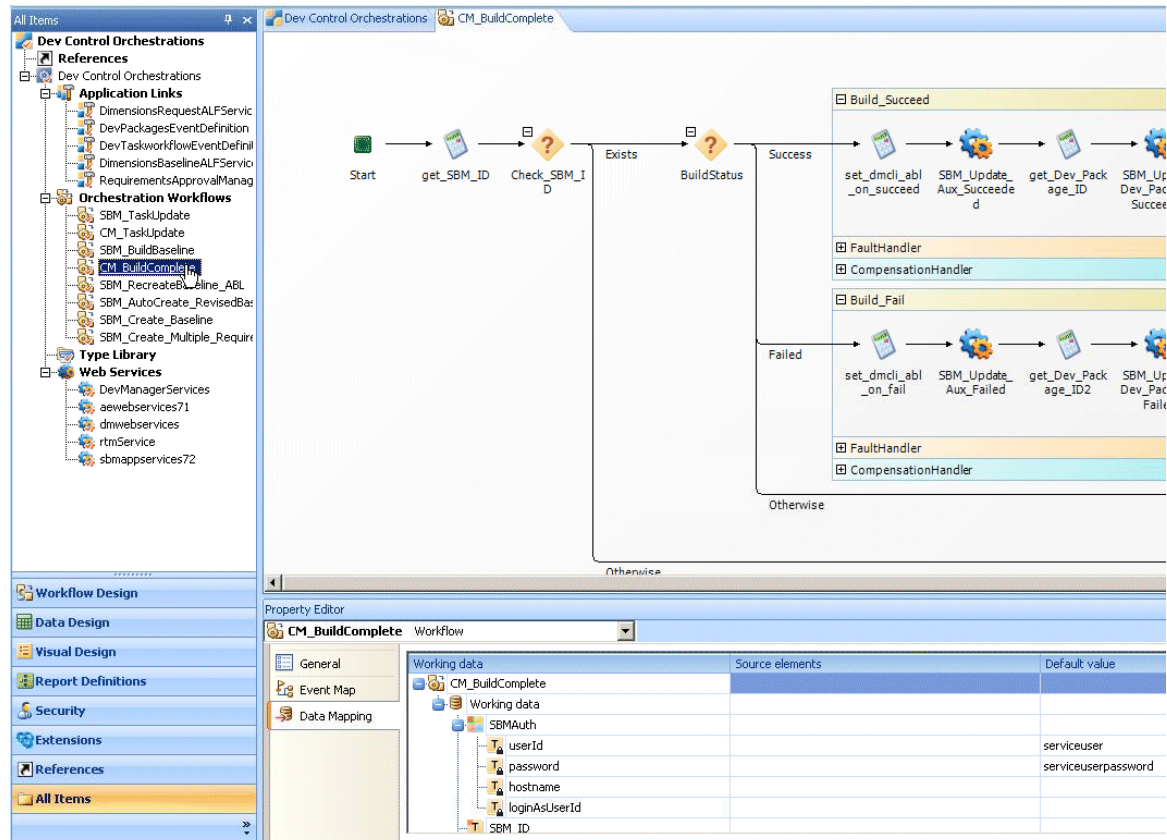
To update the Orchestration Workflows:

The orchestration workflows that may need updating are:

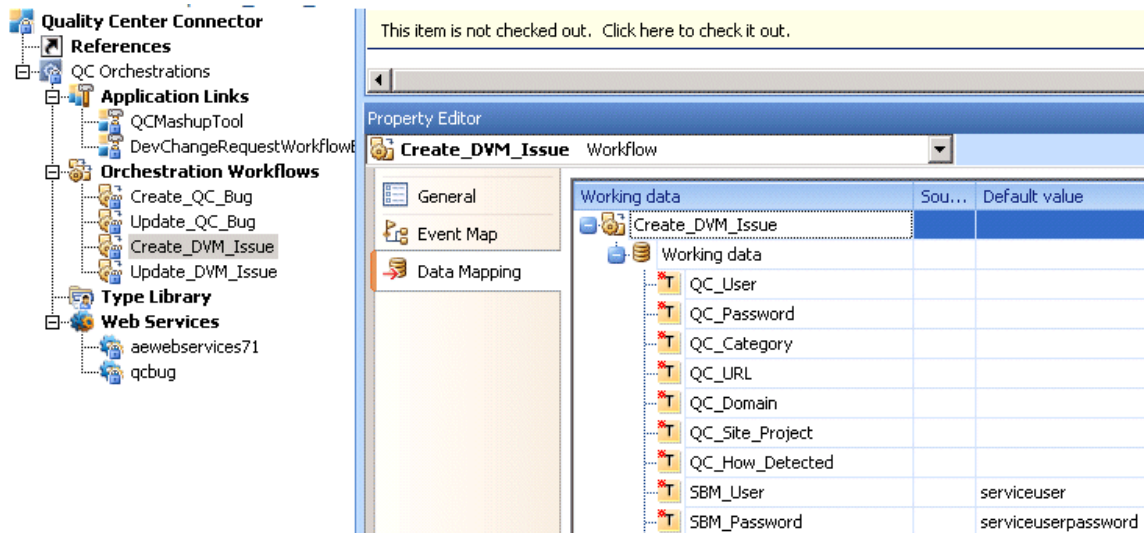
- Dev Control Orchestrations Process App
- CM_BuildComplete
- Quality Center Connector
- Create_DVM_Issue
- Update_DVM_Issue

1. Open the Dev Control Orchestrations process app.

2. Under **Orchestration Workflows**, select *CM_BuildComplete*
3. Select **Data Mapping** in the Property Editor.
4. Under SBMAuth are fields for the authentication details for the SBM server
5. Under SBMAuth, change the default values to the actual server name (if not localhost) and user credentials you will be using:



- **userId**: The connection user. if not serviceuser
 - **password**: The password for the connection user
 - **hostname**: The name of the SBM server. (If this is blank, localhost will be assumed.)
6. Open the Quality Center Connector process app.
 7. Under **Orchestration Workflows**, select *Create_DVM_Issue*.



8. Update the values for **SBM_User**, and **SBM_Password** to the values for the connection user.
9. Repeat this for the *Update DVM_Issue* orchestration workflow.

Deploy each process app when you have completed the updates.

Republishing the Process Apps

Make sure you redeploy and republish any changes you have made to each of the process applications you have changed.

Deploying the Development Manager Process Apps

When deploying the Development Manager process apps you need to do so in the following order:

1. Global Process App (eval)
2. Requirements Approval Management
3. ALM Projects
4. Dev Change Requests
5. Dev Tasks
6. Dev Packages
7. Dev Control Orchestration
8. Products and Components
9. Quality Center Connector (if you are using HP Quality Center)
10. RM to RQM Connector (if you are using Requirements Manager)
11. Release Train (if you are using Release Manager)
12. Application Release (if you are using Release Manager)
13. Release Package (if you are using Release Manager)
14. Deployment (if you are using Release Manager)
15. Environment (if you are using Release Manager)
16. ReleaseTemplate (if you are using Release Manager)

17. RLM_AUX

18. Vault_Request (if you are using Release Manager)

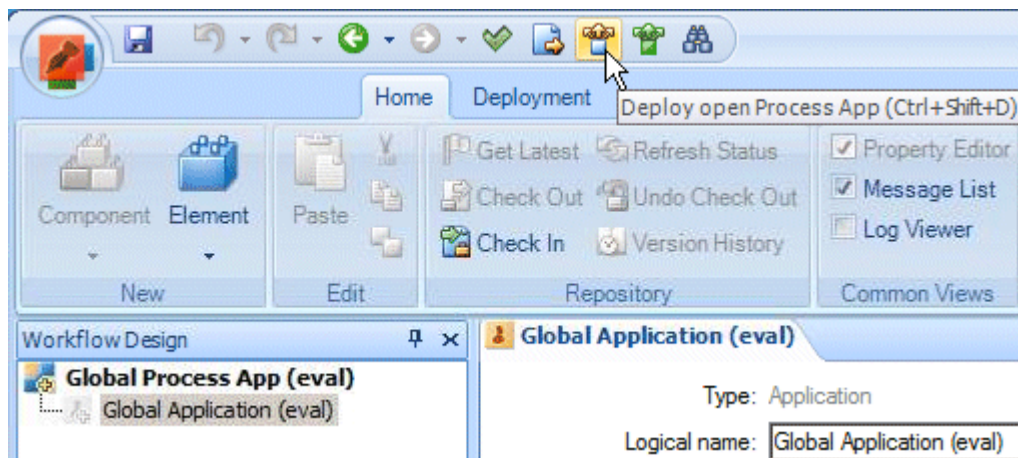
If you receive an error when deploying a process app it is likely that it contains references to another process app that needs to be deployed before it.

NOTE If you are installing a stand-alone version of Development Manager (you are not using Requirements Manager) perform the steps described in [Hiding the Requirements Manager-related Forms \[page 149\]](#) and [Removing Requirements Manager-related Columns from Report Definitions \[page 150\]](#).

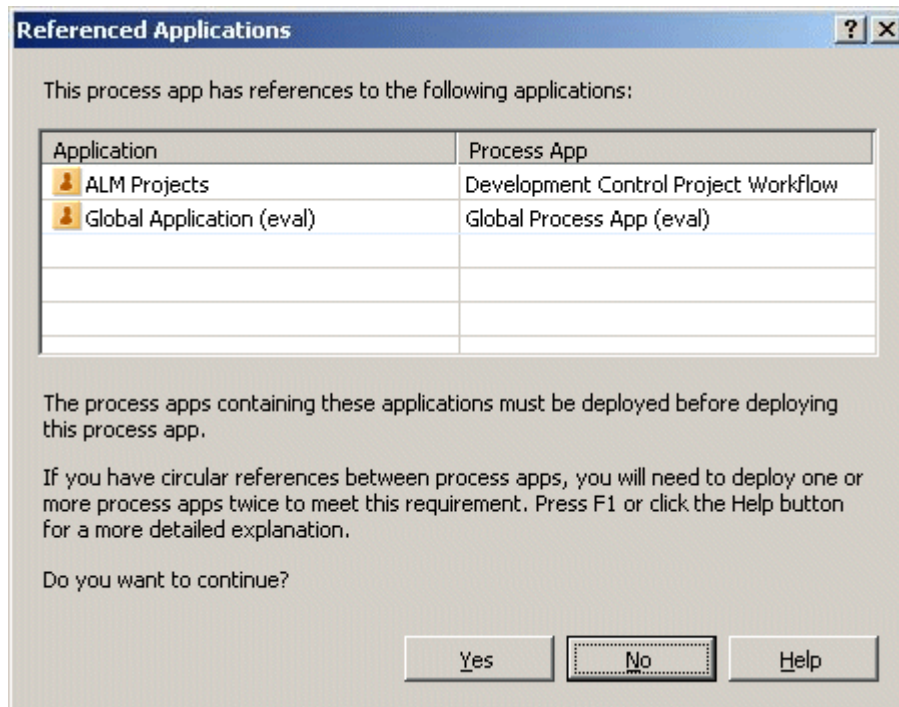
If you do not intend to take advantage of the products and components functionality of Development Manager, you may also wish to carry out the steps described in [Installing Development Manager Without Products and Components \[page 153\]](#) if you do not want them to appear in the Development Manager process apps.

To deploy a process app:

1. Open SBM Composer
2. Open the Process App.
3. Click the Deploy button on the toolbar.



You may receive a message informing you that certain process apps need to be deployed first.



If you have already deployed the applications mentioned, click **Yes**.

4. Click **Deploy**.
5. In the Deploy Process App menu, select the environment you created above.

You may see a message informing you that some endpoints have not been mapped.

If this occurs:

- a. Click the **Options** button.
 - b. For each endpoint in your process app that has a blank value, click in the field and select the necessary endpoint (which by default should have the same name) from the list.
6. Click Deploy.
 7. To view the status of the deployment, see the message list.

You can also use the Deploy and Quick Deploy buttons on the toolbar.

8. Check in the elements of the process app by right-clicking it and selecting **Check In**. Click OK in the Check In dialog box.

CAUTION! If you deploy a process app that contains a Web service, and the defining .wsdl file contains an incorrect service location for the environment to which you deployed, you need to edit the endpoint in Application Repository to point to the correct service location and then redeploy the process app so the changes can take effect. (You may also want to get the latest changes back to SBM Composer.)

If you then clone the environment and deploy the same process app to the cloned environment using default settings, the new deployment will not use the existing endpoints that you edited; it will instead create new endpoints with similar names. These new endpoints will contain the service location from the defining .wsdl file, so you need to edit them (as you did in the original environment) to point to the correct service location for the Web service. (Alternatively, you can edit the deployment in the **Deploy Options** dialog box in SBM Composer to use the correct endpoints and delete the duplicate ones.) In either case, you need to redeploy the process app to the cloned environment so these changes can take effect.

For further details of working with process apps, see the SBM Composer Guide.

Serena ALM Configuration

This section tells you how to configure and administer objects in Serena ALM. You must complete the required configuration in Serena ALM before you can begin using Serena ALM.

[Applying Licenses \[page 48\]](#)

[Configuring the Admin User for Release Manager \[page 48\]](#)

[Creating the Service User for Development Manager \[page 51\]](#)

[Using the Serena ALM Configurator \[page 53\]](#)

[Accessing the Standard SBM User Interface \[page 59\]](#)

[Release Manager Configuration \[page 60\]](#)

[Development Manager Configuration \[page 75\]](#)

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 ALM suite of products.

- **Serena ALM Solutions:** Use Serena License Manager to apply licenses for Serena Release Control and Serena Development Manager.
- **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:** Obtain license keys from Serena Support.
- **Serena Release Automation, Powered by Nolio:** Obtain license keys from Serena Support.

Documentation References

- Documentation on applying licenses in SBM for Serena Release Manager and Serena Development Manager is in the *Serena Business Manager Licensing Guide* in "Licensing Serena Business Manager".
- Documentation on applying licenses for Dimensions CM is in the *Serena Dimensions CM Administrator's Guide* in "Administering Your Licenses and the License Server".
- Documentation on applying licenses for ChangeMan ZMF is in the *SER10TY User's Guide*.
- Documentation on applying licenses for Serena Release Automation is in the *Serena Release Automation User's Guide* in "License Roles".
- Documentation on applying licenses for Serena Release Automation, Powered by Nolio, is in the *Serena Release Automation, Powered by Nolio, Installation and Administration Guide* in "Updating Serena Release Automation License".

Configuring the Admin User for Release Manager

Before you can use Serena Release Manager, you must configure the user under which Serena Release Manager executes its service calls through Single Sign-on (SSO).

Before you can use Serena Development Manager, you must configure a special service user that signs on as a service account.

To configure the Release Manager administrative user:

1. Create an administrative user.
2. Set all privileges for the administrative user to the Serena Release Manager objects, such as projects, reports, and auxiliary tables. For example, almadmin.
3. Enable roles for Serena Release Manager projects and verify that Serena Release Manager is activated.

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.

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 almadmin. Edit almadmin to give the additional unique privileges needed for Serena Release Manager.

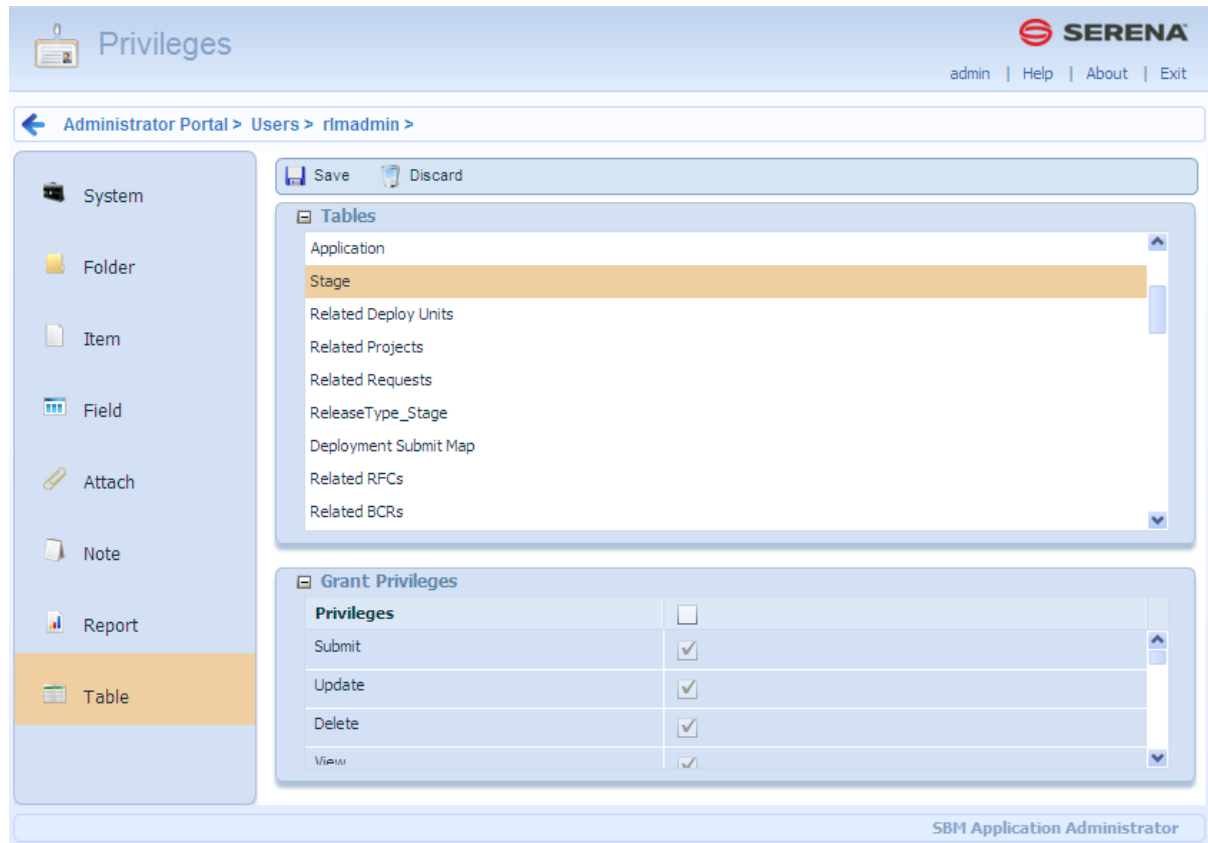
NOTE A matching administrative user ID must be set up in Dimensions CM, because single sign-on is required for the communication to work. A similar TSO user ID is required if you are implementing ChangeMan ZMF with Serena Release Manager.

Configuring the Administrative User Privileges

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

Example

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



Documentation References

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

Enabling Serena Release Manager Project Roles

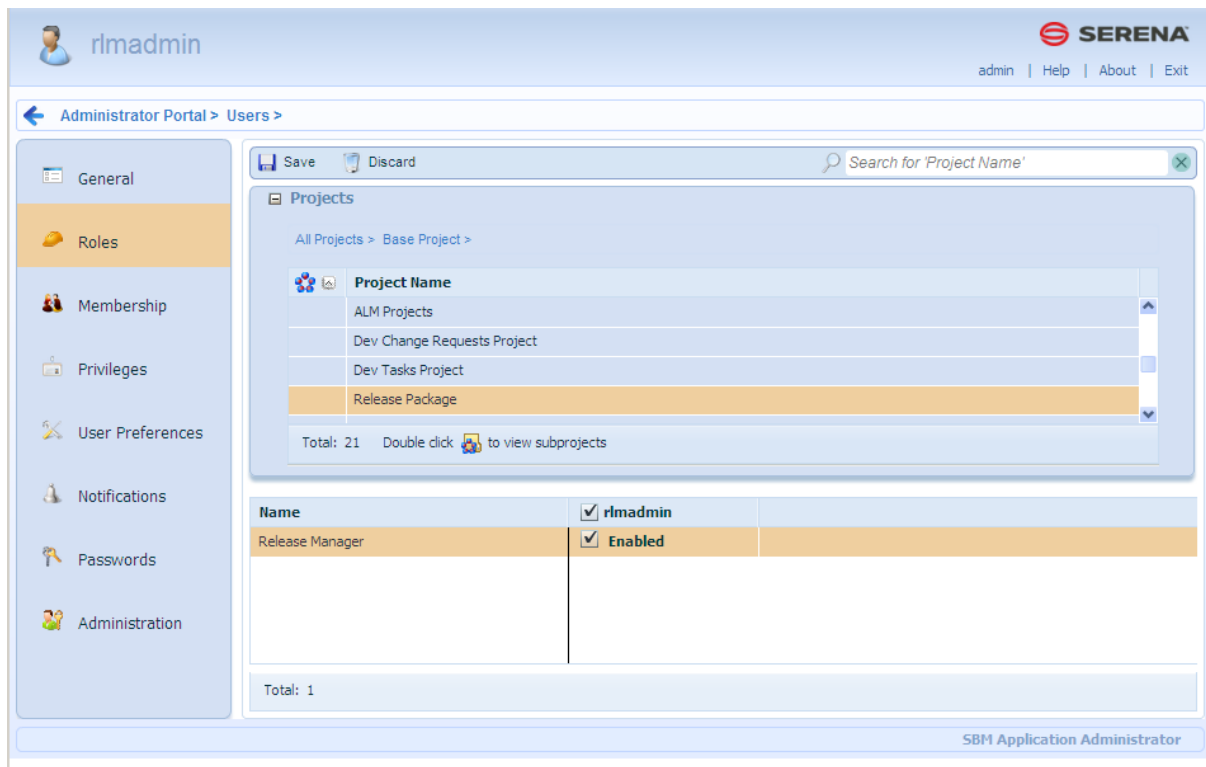
The administrative user whose login credentials are used for the Release Manager orchestrations must be enabled for all the roles that may have ownership within each of the Serena Release Manager projects.

To enable the role for each Serena Release Manager project:

1. In SBM Application Administrator, edit the administrative user. For example, almadmin.
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 Manager project and role.

Example

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



Documentation References

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

Creating the Service User for Development Manager

You must create and assign a specific user as owner for the login credentials for the Development Manager orchestrations that connect SBM to Dimensions CM and/or ChangeMan ZMF. This user must be in SBM and in Dimensions CM and/or ChangeMan ZMF.

NOTE By default, the Serena Development Manager orchestrations are configured to use *serviceuser* as the user and *serviceuserpassword* as the password. The Service user is unique to Development Manager and is used to run services as a service account. Serena Release Manager uses SSO to run services, and does not embed the user name and password in the orchestrations.

If you have promoted and deployed the *Global Process App* process app, you will not need to create and assign the user *serviceuser* in SBM if this is the service user you are using for Development Manager. You will, however, need to do this in Dimensions CM as described in [Setting Privileges for the Service User \[page 90\]](#).

To create the service user:

1. Log into the Administrator Portal.
2. Click **Users** in the main window.
3. Add a new user specifying the following properties:
 - **Login ID:** The default used by the process apps is *serviceuser*. You can specify a different user name, in which case you will need to update this in the process apps.
 - **Password:** The default used by the process apps is *serviceuserpassword*. You can specify a different password, in which case you will need to update this in the process apps.
 - **Product Access:** Set this to *Managed Administrator*.

- **System Privileges:** Set this to *All*.

4. Select **Roles**.
5. Assign all the necessary roles in each project for the service user

For each project:

- a. Select the desired project in the Projects area.
- b. In the bottom pane, select the desired role and select the checkbox under the user to enable the role.
 - For Development Manager, select the role of Administrator for each project.
 - For the Component Lifecycle Project, select the roles of Component Manager and Component Authority.

Do this for the following Development Manager projects:

- ALM Projects
- Products and Components
- Dev Change Requests
- Dev Packages
- Dev Tasks
- Requirements Approval Management
- RM to RQM Connector

6. Select **Membership** and add the user to the *Everyone* group.
7. Save your changes.

You need to give the service user all privileges to the Serena Development Manager objects, such as projects, reports, and tables.

1. Select the service user.
2. Select Edit.
3. On the Privileges tab, select the Table tab and select all privileges for the following tables:
 - Products
 - Baselines
 - Quality Connector
 - Requirements
 - RM Server Connections
4. On the Administration tab, select the tables tab, and select all privileges for the tables.
5. Select the System tab below.
6. Select **Global Administration** and click OK.

Using the Serena ALM Configurator

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

If you run the Serena ALM Configurator, the client connection properties files are configured through your entries in the ALM Configurator forms. Properties files with names that have the suffix `connection.properties` are configured through the ALM Configurator. Other files must be manually configured.

Configuring Access to the Configurator

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

To update the authorized list of users:

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

```
..\Program Files\Serena\common\tomcat\6.0\webapps\alm\WEB-INF\classes
```

2. Open the `rlm.properties` file.
3. Set the properties for the connection as follows:

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

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

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

4. Restart the Serena common Tomcat service.

Invoking the Serena ALM Configurator

To invoke and log in to Serena ALM Configurator:

1. Enter the Serena ALM Configurator URL. For example:

```
http://almhost:8080/alm
```

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

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

Upon successful login, your Serena ALM Configurator appears.

Entering Information in the Serena ALM Configurator

You can use the Serena ALM Configurator to configure the client connections.

To add or update the client connection properties:

1. In the Serena ALM Configurator, select the **CONFIG** page.
2. Enter the connection information in the forms for each of the following tabs:
 - **ALF**: Application Lifecycle Foundation for communication between systems
 - **SBM**: Serena Business Manager for the core system as well as deployment change request (DCR) or other provider

- **BCR:** Business Change Request provider (default SBM Incidents)
- **RFC:** Request for Change provider (default SSM, Serena Service Manager)
- **Dimensions CM:** Release vault, deployment unit (DU) and optionally DCR provider
- **Release Automation:** for installing and configuring into target environments
- **Release Automation (Nolio):** powered by Nolio, for installing and configuring into target environments
- **ZMF:** ChangeMan ZMF: Release vault and DU provider

The information is saved in separate files in the Serena Release Manager common Tomcat Web server classes folder.

3. Restart the Serena Common JBOSS and IIS Admin Service services.
4. Restart the Serena common Tomcat service.

ALF Client Connection Information

Specify the connection to Application Lifecycle Framework (ALF), which is used by Serena Release Automation, powered by Nolio, and by Srena Development Control to communicate with Serena Release Manager.

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

To update the ALF client connection information:

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

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

Example

alf-client-connection.properties

```
ALF_EVENTMANAGERURL = http://sbmhost:8085/eventmanager/services/
    ALFEventManagerDocLit
AE_USERID = almadmin
AE_PASSWORD = almadmin_test
```

SBM Application Connection Information

Specify the instance of SBM if you are using SBM as a provider for Development Change Requests (DCRs). These must reside in the same SBM system as Serena Release Manager, as some of the

functions implemented in Serena Release Manager use the Web services to implement the calls and use the provider to determine the instance of SBM.

IMPORTANT! The `sbm-client-connection.properties` file must have valid connection information to enable the creation of objects in Serena Release Manager. If this connection is not configured properly, you will receive a checkuniqueness error.

To update the SBM connection information:

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

Parameter	Value
SBM_APPWS_URL	URL to the SBM server that provides the DCRs, in the form of: <code>http://<sbm_hostname>:<port>/gsoap/gsoap_ssl.dll?sbmappservices72</code>
sbm.service.user	User ID with access to the appropriate SBM projects and tables controlled by Serena Release Manager. This user ID must exist in SBM.
sbm.service.password	Password for the SBM user ID.

Example

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

`sbm-client-connection.properties`

```
SBM_APPWS_URL = http://almhost:80/gsoap/gsoap_ssl.dll?sbmappservices72
sbm.service.user = almadmin
sbm.service.password = almadmin_test
```

BCR Provider Connection Information

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

To update the BCR provider connection information:

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

Parameter	Value
bcr.ws.app.url	URL to the SBM server that provides the BCRs, in the form of: <code>http://<sbm_hostname>:<port>/gsoap/gsoap_ssl.dll?sbmappservices72</code>
bcr.user	User ID with access to the appropriate SBM projects and tables controlled by Serena Release Manager. This user ID must exist in SBM.
bcr.password	Password for the SBM user ID.

Example

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

bcr-connection.properties

```
bcr.ws.app.url=http://alomhost:80/gsoap/gsoap_ssl.dll?sbmappservices72
bcr.user=aladmin
bcr.password=aladmin_test
```

RFC Provider Connection Information

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

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

To update the RFC provider connection information:

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

Parameter	Value
itsm.ws.app.url	URL to the SBM server that provides the RFCs, in the form of: http://<sbm_hostname>:<port>/gsoap/ gsoap_ssl.dll?sbmappservices72
itsm.user	User ID with access to the appropriate SBM projects and tables controlled by Serena Release Manager. This user ID must exist in SBM.
itsm.password	Password for the SBM user ID.

Example

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

itsm-connection.properties

```
itsm.ws.app.url=http://alomhost:80/gsoap/gsoap_ssl.dll?sbmappservices72
itsm.user=aladmin
itsm.password=aladmin_test
```

Dimensions CM Client Connection Information

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

To update the Dimensions CM client connection information:

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

Parameter	Value
DIM_WS_URL	URL to the server where the Dimensions CM Web services are installed, in the form of: http://<cm_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.
DIM_SERVICEUSER	User ID used by DVM to sign on to the Dimensions CM service.
DIM_SERVICEUSERPASSWORD	Password used by DVM to sign on to the Dimensions CM service.

Example

dm-client-connection.properties

```

DIM_WS_URL = http://dimcm_host:8080/dmwebservices2/services/dmwebservices/
DIM_DBNAME = cm_typical
DIM_DBCONN = Dim12
DIM_SERVER = dimcm_host
DIM_SERVICEUSER = serviceuser
DIM_SERVICEUSERPASSWORD = serviceuserpassword

```

Serena Release Automation Client Connection Information

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

To update the Release Automation client connection information:

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

Parameter	Value
ra.server.url	The URL that points to Web services on the Serena Release Automation execution server, in the form of: http://<ra_hostname>:<port>/serena_ra/rest/deploy/
ra.server.username	The Serena Release Automation execution server user name.
ra.server.password	The Serena Release Automation execution server password.
useSso	Set this to true. Serena Release Manager uses SBM Single Sign On to login to Serena Release Automation.

Example

ra-client-connection.properties

```
ra.server.url=http://srahost:9095/serena_ra/rest/deploy/
ra.server.username=rlmadmin
ra.server.password=rlmadmin
useSso=true
```

Serena Release Automation (Nolio) Client Connection Information

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

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

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

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

Example

nolio-client-connection.properties

```
nolio.ws.openapi.uri=http://sranolio_host:8080/datamanagement/ws/OpenAPIService
nolio.username=superuser
nolio.password=suser
```

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

ZMF Client Connection Information

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

To update the ChangeMan ZMF connection information:

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

Parameter	Value
ZMF_WS_URL	URL to the ALM server where the ALMZMF Web services are installed, in the form of: http://<alm_hostname>:<port>/almzmf/services/ZMFPackageServices/
ZMF_SERVER_HOSTADDRESS	Hostname or IP address for the z/OS mainframe server where the ZMF started task, or ZMF application, is running.
ZMF_SERVER_HOSTPORT	Port number for the z/OS mainframe server.
ZMF_SERVER_PROXYID	Mandatory proxy user ID used to log into the z/OS mainframe server on behalf of a SBM user to ZMF. See Configuring a ZMF Proxy User ID [page 103] .
ZMF_SERVER_PROXY_PASSWORD	Password for the proxy user ID.
SERNET_HTTPSERVER	URL to the z/OS mainframe SERNET HTTP server instance, in the form of: http://<sernet_hostname>:<port> This is required to populate the ZMF UI widgets in Release Manager. See Configuring the SERNET HTTP Server [page 94] .

Example

zmf-client-connection.properties

```
ZMF_WS_URL = http://almhost:8080/almzmf/services/ZMFPackageServices/
ZMF_SERVER_HOSTADDRESS = zmf_host
ZMF_SERVER_HOSTPORT = 5035
ZMF_SERVER_PROXYID = ALMMAN
ZMF_SERVER_PROXY_PASSWORD = ALMPWD1
SERNET_HTTPSERVER = http://zmf_host:5083
```

Accessing the Standard SBM User Interface

Serena ALM 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 ALM standard SBM user interface:

1. Login to the Serena ALM Web client.
2. Remove the shell parameter from the URL.

For example, if your Serena ALM URL is:

```
http://almhost/tmtrack/tmtrack.dll?shell=alm
```

your Serena ALM standard SBM user interface URL would be:

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

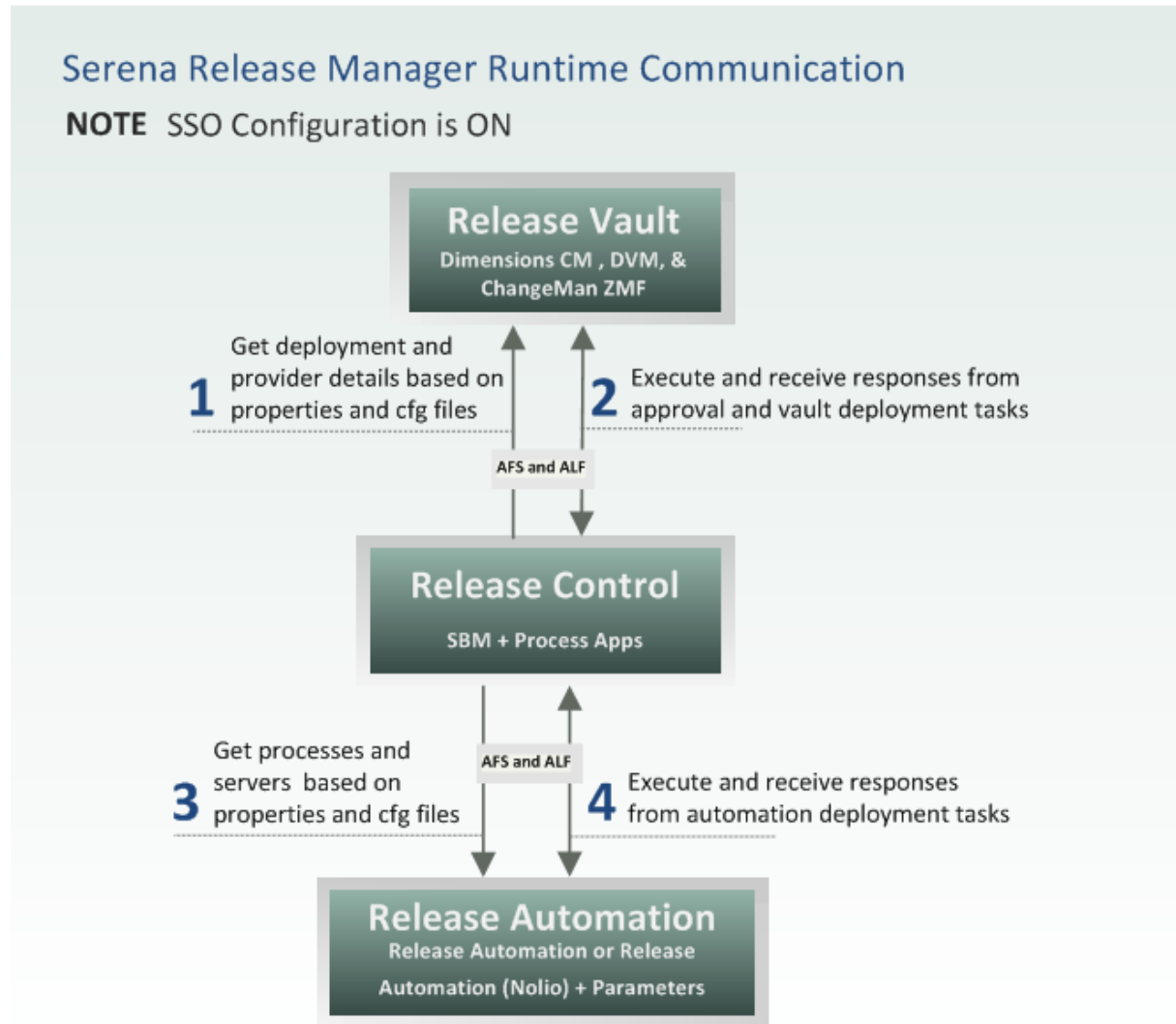
Release Manager Configuration

This section tells you how to configure and administer objects in Release Manager.

Serena Release Manager Runtime Communication

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

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



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 or Serena Release Automation (Nolio) 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.

Adding Your Application Names in Serena Release Manager

Before you start releasing applications using Serena Release Manager, you must pre-populate Serena Release Manager 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 Manager.

What Can You Change in Serena Release Manager?

You can change the following application information:

- Add application name and description.
- Update application name and description.

What is the Impact?

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

How Do You Change It?

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

To change the Application table entries:

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

Documentation References

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

Adding Your Server Names in Serena Release Manager

Before you start using environments in Serena Release Manager, you must pre-populate Serena Release Manager 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 Manager.

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 Manager 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 Manager objects, but you may choose to extend and customize this functionality. For example, you could customize the system to use the environment server configuration to store parameters in specific field values, which could then be passed to Serena Release Automation.

What Can You Change in Serena Release Manager?

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 SBM Application Administrator Guide* in "About Auxiliary Data".

Enabling Scheduling of Deployment Tasks

If you want to be able to schedule execution time of either vault or automation deployment tasks, you must first configure notifications and escalations in SBM.

To enable scheduling of deployments tasks:

1. In the SBM Configurator, enable the mail server as follows:
 - a. On the navigation pane, select **Mail Services**.
 - b. In the **Notification Server** tab, select **Enable Notification Server**.
 - c. Click **Apply**.
2. In the SBM Administrator Portal, configure the **Schedule automation deployment task** notification for the **Automation Task** workflow as follows:

- a. On the Process apps/applications tree, select **Deployment**.
 - b. Under **Workflow name**, double-click **Deployment** to expand the workflow name tree.
 - c. In the expanded tree, select **Automation Task**.
 - d. In **Notifications**, select **Schedule automation deployment task**.
 - e. In the navigation pane, select **Subscriptions**.
 - f. For the Serena Release Manager administrative user ID that is defined in the SBM tab of the ALM Configurator, select **Notify** and **Allow to Subscribe**.
 - g. In the navigation pane, select **Escalation**.
 - h. In **Delay Parameters**, select **Delay**.
 - i. In **Send when interval between**, select **Schedule execution time**.
 - j. In the navigation pane, select **General**.
 - k. Click **Save** to save the configuration settings.
3. In the SBM Administrator Portal, configure the **Schedule DimCM deployment task** notification for the **Vault Task** workflow as follows:
- a. On the Process apps/applications tree, select **Deployment**.
 - b. Under **Workflow name**, double-click **Deployment** to expand the workflow name tree.
 - c. In the expanded tree, select **Dim Deployment Task**.
 - d. In **Notifications**, select **Schedule DimCM deployment task**.
 - e. In the navigation pane, select **Subscriptions**.
 - f. For any user who will submit automation deployment tasks, such as your Release Managers and Release Engineers, select **Notify** and **Allow to Subscribe**.
 - g. In the navigation pane, select **Escalation**.
 - h. In **Delay Parameters**, select **Delay**.
 - i. In **Send when interval between**, select **Schedule execution time**.
 - j. In the navigation pane, select **General**.
 - k. Click **Save** to save the configuration settings.

Managing Release Manager Users

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

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

Adding Users and Groups

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

What Can You Change in Release Manager?

User and group changes are done in native SBM.

What Can You Change in SBM?

You can change the following user information:

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

What is the Impact?

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

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

- Enable roles in projects for any workflows in which this role may be assigned ownership.
- Change privileges of roles that are assigned to states.
- You may need to change or add related roles that are needed in Dimensions CM for performing the necessary actions, as they won't be in the default process model. To define roles, see "Users and Roles" in the *Dimensions CM Process Modeling User's Guide*.

How Do You Change It?

You should manage users according to the SBM documentation.

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

Documentation References

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

Example

The default roles defined in Serena Release Manager are Release Manager and Release Engineer. Typical release management roles are shown in the following table.

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 [page 65] in some organizations.

Role Name	Description
Change Manager	The role that is responsible for the IT Operations for the systems where the pre-production and production release environments reside. A change manager approves deployment into pre-production and production environments.
Development Manager	The role that is responsible for and approves development activities for a release. A development manager would typically be consulted during the release management process and provide approval on the content of a release package.
Installation Manager	<p>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 Manager and is responsible for deployment of request packages into environments.</p> <p>This role may also be the Build Manager [page 64] in some organizations.</p>
QA Manager	The role that is responsible for and approves testing activities for a release.
Release Engineer	<p>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.</p> <p>This role may be the Installation Manager [page 65] or Build Manager [page 64] in some organizations.</p>
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 Manager User Interface Reports

Reports help keep release management stakeholders informed of status, history, and other release information. Serena Release Manager provides a default set of reports that are used to populate the user interface panes.

In addition to reports provided, you can configure custom reports to provide the information you need for your organization's Release Manager. The default reports configured for Serena Release Manager are shown in the following sections.

What Can You Change in SBM?

You can change the following report information:

- Add reports
- Modify reports
- Delete reports
- Manage access to reports

What is the Impact?

- If you add a report, you must give users access to the report. Any users that do not have access or the assigned role will not see the report when they log in to the UI.
- If you do not use the proper naming convention, the report will not be listed in the drop-down list. For the report to appear in the UI and to be listed in the drop-down menu that filters the view, you must use one of the required report prefixes.
- If you change report criteria, the results that appear in the UI will change.

- You can change existing reports to add columns in your UI, but you should be careful when changing the title or reference name. If you change either of these, the report may no longer appear in the **View All** drop-down list or in the work area view itself, or underlying behaviors may be impacted. Make sure you understand the use of the report you are changing before you implement a change.
- If you delete a report that populates the UI, the name of the report no longer appears in the **View All** drop-down menu.

How Do You Change It?

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

You can create new reports for the UI if you need a report that represents something other than the default reports included in Serena Release Manager.

Changing Existing Reports

You can change existing reports to add columns, but you should not change the title or reference name. If you change either of these, the report may no longer be listed in the **View All** drop-down menu or the results may not appear in the work area pane.

To change the reports:

1. From the native SBM interface, select the process app for the report you want to change.
2. Change the report to add or remove fields that you want to appear or to change the selection criteria for the report data.

The changes you make to the report automatically appear in the UI if you save with the same name and reference name.

CAUTION! Do not change the title or reference name of the report. If you need a different report name, see [Creating New Reports \[page 66\]](#).

Creating New Reports

You can create new reports and register them in the UI if you need a report that represents something other than the default reports included in Serena Release Manager.

Reports that appear in the UI must be created with a specific title and reference name naming convention. The UI logic parses the name to determine whether the report should appear in the page and associated drop-down list, and on which page it should appear.

The UI logic uses the first node or nodes to determine the process app and associated page for the report and the second or middle node to determine the display name.

To create the reports:

1. From the native SBM interface, select the process app for the report you want to add.
2. Create the report with a specific name for the process app under which you want it to appear.
3. Make sure the users who need to see the report have the correct role and privileges.
4. Register the report.

Naming Convention and Reference Rules

The SBM reports that you create for Serena Release Manager must have title and reference names that follow a specific naming convention, and they must be referenced in the associated JavaScript file according to specific reference rules. These naming conventions and rules are documented as follows:

- [Report Naming Convention \[page 67\]](#)
- [Report Reference Rules \[page 67\]](#)

Report Naming Convention

Syntax

<category title>.<view title>.<view description>

or

<category-specific-prefix>.<view title>.<view description>

Spaces should be included in <view title> as needed, since this is the display name for the report in associated drop-down menus.

<view description> is optional, and is not displayed in the shell UI in the default implementation of Serena Release Manager. This is useful for documentary purposes, and could be used for tooltips in custom or future releases of the UI.

Example

DevTasks.View.In Progress.All Tasks That Are Currently In Progress

In this example,

<category-specific-prefix> = DevTasks.View

<view title> = In Progress (visible in the shell UI)

<view description> = All Tasks That Are Currently In Progress

Report Reference Rules

- The report name used in the UI is the report title. For example, the reports shown in the **Release Trains | View All** menu and the corresponding report names are as follows:

The screenshot displays the Serena Business Manager web interface. At the top, there's a navigation bar with tabs: 'Inbox', 'Release Trains', 'Application Releases', 'Release Packages', 'Deployment Tasks', and 'environments'. Below this, a 'Views' dropdown menu is open, showing options: 'All', 'Approved', 'Completed', 'Failed', 'In Progress', 'Mine', 'Planning', and 'Review'. The 'In Progress' option is highlighted. The main content area shows a table with columns: 'Item ID', 'Submitter', 'Official release name', 'Description', 'Release type', and 'State'. The table contains one row with 'Item ID' RTHAX, 'Submitter' M Admin, 'Official release name' QML_Q3_S01, 'Description' Major, and 'State' Planning. Below the table, there's a 'SERENA BUSINESS MANAGER' header and a 'Welcome, ALM Admin' message. On the left, there's a 'Reports' sidebar with 'Basic Tasks' (Show me my Reports, Find Reports, Create Report) and 'Advanced Tasks' (Browse Application Reports, Browse Multi-View Reports, Browse Multi-Table Reports, Browse Built-In Reports, Browse System Reports, Access SLA Reports). The main area shows a 'Release Train New Project (Release Train)' table with columns: 'Title', 'Privilege Category', 'Created', 'Author', 'Last Modified', 'Last Modifier', and 'Last Run'. The table lists several reports, with 'Release Trains View All' highlighted in yellow. Other reports include 'Release Trains View Approved', 'Release Trains View Completed', 'Release Trains View Failed', 'Release Trains View In Progress', 'Release Trains View Mine', 'Release Trains View Planning', and 'Release Trains View Review'.

Title	Privilege Category	Created	Author	Last Modified	Last Modifier	Last Run
Inactive Release Trains	Guest	01/09/2012	Serena Business Manager	07/20/2012	Administrator	01/09/2012
Release Trains In Qant	Guest	05/11/2011	Administrator	07/20/2012	Administrator	07/26/2012
Release Trains View All	Guest	06/22/2012	Serena Business Manager	07/15/2012	Administrator	07/27/2012
Release Trains View Approved	Guest	06/22/2012	Serena Business Manager	07/20/2012	Administrator	07/15/2012
Release Trains View Completed	Guest	06/22/2012	Serena Business Manager	07/20/2012	Administrator	12/20/2011
Release Trains View Failed	Guest	06/22/2012	Serena Business Manager	07/20/2012	Administrator	12/20/2011
Release Trains View In Progress	Guest	07/15/2012	Serena Business Manager	07/20/2012	Administrator	07/15/2012
Release Trains View Mine	Guest	06/22/2012	Serena Business Manager	07/20/2012	Administrator	07/16/2012
Release Trains View Planning	Guest	06/22/2012	Serena Business Manager	07/20/2012	Administrator	07/15/2012
Release Trains View Review	Guest	07/15/2012	Serena Business Manager	07/20/2012	Administrator	07/15/2012

- The report itself should not include filtering or sorting. The reference name syntax provides the information used by the UI code to filter and sort the reports.

- Reports may be referenced by the reference name, and may appear as "category view".
- Category Definitions Categories are defined in `..\Serena\SBM\Application Engine\template\shell\alm\config\views.jsvar`.

Each category can have the following defined:

name: a code-name, to be used internally

title: to be shown as a section-button caption

url: the page URL to be shown; currently, the same page with parameters is displayed

description: documentary only

report_prefix: a prefix for reports to be shown as a views. Category title or name is considered a prefix if it was not configured exactly.

demand_permission: the privileges to be checked for access the category

List of Reports

The default reports for the Serena Release Manager UI are shown in the following table.

Objects	Report Title	Description
Inbox	RLM.MyWork.Inbox.RelTrain.AppRel	Release trains assigned to the current user.
	RLM.MyWork.Inbox.DepTask.DepTemplate	Deployment tasks assigned the current user.
	RLM.MyWork.Inbox.Env.RelPackage	Release packages assigned the current user.
Release Trains	Release Trains.View.All	All release trains in active status.
	Release Trains.View.Mine	Release trains assigned to the current user.
	Release Trains.View.Planning	All release trains in the Planning state.
	Release Trains.View.Approved	All release trains in the Approved state.
	Release Trains.View.In Progress	All release trains in the In Progress state.
	Release Trains.View.Review	All release trains in the Review state.
	Release Trains.View.Completed	All release trains in the Completed state.
	Release Trains.View.Failed	All release trains in the Failed state.

Objects	Report Title	Description
Application Releases	Application Releases.View.All	All application releases in active status.
	Application Releases.View.Mine	Application releases assigned to the current user.
	Application Releases.View.Planning	All application releases in the Planning state.
	Application Releases.View.In Progress	All application releases in the In Progress state.
	Application Releases.View.Review	All application releases in the Review state.
	Application Releases.View.Completed	All application releases in the Completed state.
Release Packages	Release Packages.View.All	All release packages in active status.
	Release Packages.View.Mine	Release packages assigned to the current user.
	Release Packages.View.Ready for UAT	All release packages in the Ready for UAT state.
	Release Packages.View.UAT	All release packages in the UAT state.
	Release Packages.View.Ready for Production	All release packages in the Ready for Production state.
	Release Packages.View.Production	All release packages in the Production state.
	Release Packages.View.Ready for Deployment	All release packages in the Ready for Deployment state.
	Release Packages.View.Failed Testing	All release packages in the Failed Testing state.
	Release Packages.View.Failed Deployment	All release packages in the Failed Deployment state.

Objects	Report Title	Description
Deployment Tasks	Deployment Tasks.View.All	All deployment tasks in active status.
	Deployment Tasks.View.Mine	All deployment tasks assigned to the current user.
	Deployment Tasks.View.Planned	All deployment tasks in the Planned state.
	Deployment Tasks.View.In Progress	All deployment tasks in the In Progress state.
	Deployment Tasks.View.Failed	All deployment tasks in the Failed state.
Environments	Environments.View.All	All environments in active status.
	Environments.View.Mine	All environments assigned to the current user.
	Environments.View.Commissioned	All environments in the Commissioned state.
	Environments.View.Decommissioned	All environments in the Decommissioned state.
Deployment Process Templates	Release Templates.View.All	All deployment process templates in active status.
	Release Templates.View.Mine	All deployment process templates assigned to the current user.
	Release Templates.View.Available	All deployment process templates in the Available state.
	Release Templates.View.Failed	All deployment process templates in the Failed state.
Vault Requests	Vault Requests.View.All	All vault requests in active status.
	Vault Requests.View.Mine	All vault requests assigned to the current user.
	Vault Requests.View.Ready for Delivery	All vault requests in the Ready for Delivery state.
Vault Templates	Vault Request Templates.View.All	All vault templates in active status.
	Vault Request Templates.View.Mine	All vault templates assigned to the current user.
	Vault Request Templates.View.Available	All vault templates in the Available state.

Privileges Required for UI Reports

For users to see a particular UI report's results when they select a page in Serena ALM, they must have one of the following item privileges within the related process app:

- View All Items
- View Item if Owner
- View Item if Secondary Owner
- View Item if Submitter

Managing Release Manager Notifications

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

Serena Release Manager provides a default set of notifications, and you can configure these as needed to support your organization. The notifications configured by default for Serena Release Manager are shown in the following table.

Workflow	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

Workflow	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 (and sub-workflows)	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

Workflow	Notifications
Release Template (Deployment Process Template)	RT - Any Release Template changes owner
	RT - Any Release Template changes state
	RT - Any Release Template changes to inactive
	RT - Any Release Template I submitted changed state
	RT - Any Release Template I submitted changed to inactive
	RT - Any Release Template is submitted
	RT - I become the owner of any Release Template
Rlm Aux (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
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

Workflow	Notifications
Vault Request	VR - Any Vault_Request changes owner
	VR - Any Vault_Request changes state
	VR - Any Vault_Request changes to inactive
	VR - Any Vault_Request I submitted changed state
	VR - Any Vault_Request I submitted changed to inactive
	VR - Any Vault_Request is submitted
	VR - I become the owner of any Vault_Request
Vault Template	VT - Any Vault_Request changes owner
	VT - Any Vault_Request changes state
	VT - Any Vault_Request changes to inactive
	VT - Any Vault_Request I submitted changed state
	VT - Any Vault_Request I submitted changed to inactive
	VT - Any Vault_Request is submitted
	VT - I become the owner of any Vault_Request

What Can You Change in Release Manager?

Notification changes are done in native SBM.

What Can You Change in SBM?

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

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

You can change the following notification information:

- Add notifications
- Modify notifications
- Subscribe users to notifications

What is the Impact?

If you add a notification, you should subscribe users to it so that the notification is sent to them when the associated event occurs.

How Do You Change It?

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

Documentation References

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

Development Manager Configuration

This section tells you how to configure and administer objects in Development Manager.

There are a number of different possible scenarios, depending on whether you are upgrading an existing version of Serena Development Manager, and whether you will be also using Serena Requirements Manager. These situations require different configuration steps to be carried out. For details see [Development Manager Installation and Upgrade Scenarios \[page 147\]](#).

Creating Users and Assigning Privileges

You need to create users and assign any necessary privileges on the SBM server. You will also need to create and assign the necessary users in Dimensions CM and/or ChangeMan ZMF.

Creating and Assigning the Development Manager Users

The out-of-the-box installation of Development Manager includes a set of users that have the appropriate roles in the process apps assigned. If you have promoted and deployed the *Global Process App*, you will not need to create and assign these users. These are:

- Mira: Project Manager
- Matt: Business Analyst
- Kaci: Product Owner
- Eva: Component Manager
- Josh: Development Manager
- Emma: Development Engineer
- Amy: Build Manager
- Alexander: Build Engineer
- Dinesh: QA Manager
- Ashley: QA Engineer
- Dave: Release Manager
- Oscar: Release Engineer
- serviceuser: Administrator

If you are setting up other users, you need to create them and assign the corresponding roles in each of the projects for which those roles exist: Each role in each project must have at least one user assigned to it for the applications to function.

The projects are:

- Dev Tasks Project
- ALM Projects
- Dev Change Requests Project
- Dev Package Project

- Component Lifecycle Project

If you are using the Agile feature of Development Manager, you will need to assign users the roles:

- Product Owner
- Scrum Master

In the project ALM Projects.

When you create your own users, you will need to assign the correct roles for those users. For example, You would carry out the following steps for the Development Manager user:

1. Log into SBM Administrator as an administrative user who has privileges to update users by entering the following URL in a browser

`http://<hostname>/tmtrack/tmtrack.dll?shell=alm`
2. Click the arrow next to your user name at the top right of the UI, and select **Administrator Portal**
3. Click **Users** in the main window.
4. Click Add, and enter the details for the user, selecting the option **Regular User.:**
5. Click **Save**.
6. Double-click the name of the user.
7. Select **Roles**.
8. Double-click *Base Project*.
9. Select a project and select the required role in the list below. For example for the Development Manager user, select Development Manager.

Also select the *User* role.

10. Repeat the previous step for each of the projects where the role is listed.
11. Click **Save**.

Setting Privileges for the ALM_USER_SETTINGS Table

All users that will use ALM Suite UI need to have privileges for ALM_USER_SETTINGS auxiliary table.

1. Under Groups, select the Everyone group.
2. Select Tables, and select the ALM_USER_SETTINGS table.
3. Select Submit, Update and View under Grant Privileges.

Documentation References

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

Configuring Agile Properties

If you are using the Agile features of Development Manager, you need to set some values in the configuration properties file. This is located in:

`<tomcat home>\webapps\alm#agile\WEB-INF\classes\config.properties`

Edit the file and:

1. Set the parameters:

- `ws.url`
- `ws.ext.url`
- `sbm.webadminjsoncallurl`

to the correct values for your SBM server. For example:

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

```
ws.ext.url=http://localhost:80/gsoap/gsoap_ssl.dll?sbmappservices72
```

```
sbm.webadminjsoncallurl=http://localhost:80/tmtrack/tmtrack.dll?
```

2. Set the parameters:

- `sbm.service.user=<sbm.service.user>`
- `sbm.service.password=<sbm.service.password>`

To the correct values for the service user and password you are using. For example:

```
sbm.service.user=serviceuser
```

```
sbm.service.password=serviceuserpassword
```

When you have completed the changes, Restart Tomcat.

Dimensions CM for Serena ALM Configuration

This section tells you how to configure and administer objects for communication and integration with Dimensions CM. You must complete the following configuration before you begin using Serena Release Manager or Serena Development Manager.

[Dimensions CM for Release Manager Configuration \[page 78\]](#)

[Dimensions CM for Development Manager Configuration \[page 84\]](#)

Dimensions CM for Release Manager Configuration

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

[Dimensions CM Communication Configuration Overview \[page 78\]](#)

[Configuring Communication on the Dimensions CM Server \[page 79\]](#)

[Configuring Dimensions CM Communication in Release Manager \[page 81\]](#)

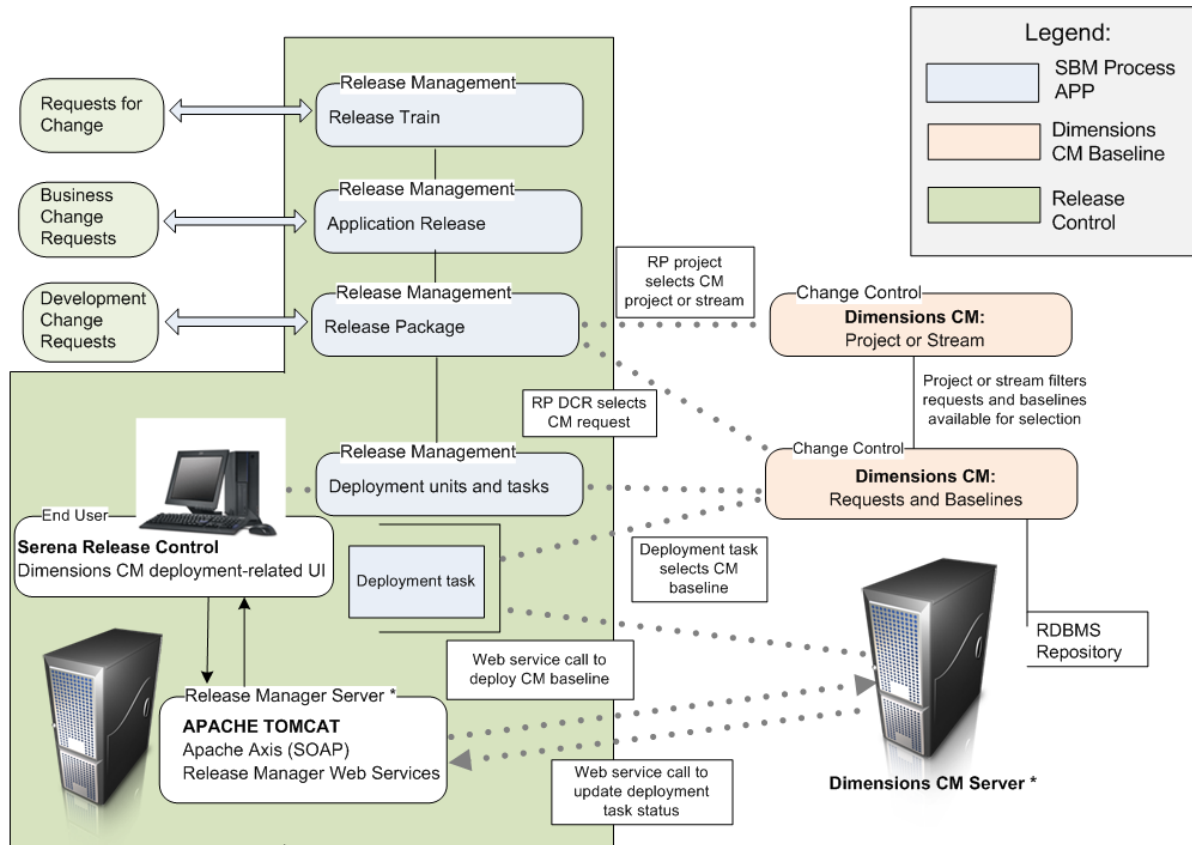
[Configuring Objects in Dimensions CM \[page 82\]](#)

Dimensions CM Communication Configuration Overview

You must configure Dimensions CM communication on the Dimensions CM server and on the Serena ALM server to activate the integration.

If you have Serena Development Manager (DVM) installed as part of the Serena ALM suite, your implementation of Dimensions CM is shared between Serena Release Manager and Serena Development Manager. Information that applies to Dimensions CM here also applies to Serena Development Manager. For information on installing and configuring Serena Development Manager, see *Serena Development Manager Installation and Configuration*.

The architecture that supports the Dimensions CM integration with Serena Release Manager is shown in the following figure.



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

Configuring Communication on the Dimensions CM Server

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

- Specify ALF event configuration information in the `dm.cfg` file.

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

For details, see the following:

- [Specifying Dimensions CM ALF Event Configuration Information \[page 80\]](#)
- [Specifying Selection Criteria for Dimensions CM Events and Objects \[page 81\]](#)

Specifying Dimensions CM ALF Event Configuration Information

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

To update the ALF events configuration for Dimensions CM:

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

```
C:\Program Files\Dimensions 12.2\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 DimensionsUnderOrchestratedApps
```

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

3. Restart the Dimensions CM Listener service.

Example

dm.cfg

```
#####
# ALF Events Configuration
#####
DM_ALF_ENDPOINT http://sbmhost:8085/eventmanager/services/ALFEventManager
DM_ALF_USER almadmin
DM_ALF_PASSWORD almadmin_pswd
DM_ALF_EVENT_CONFIG %DM_DFS%alf_events_config.xml
DM_ALF_PRODUCT_INSTANCE DimensionsUnderOrchestratedApps
```


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 [Provider Configuration \[page 121\]](#).

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\12.2\CM\dfs
2. Update the ALF_EVENTS_CONFIG.XML file to specify your Dimensions CM database name, project name, baseline type, and deploy event.
3. Restart the Dimensions CM Listener service.

Example

ALF_EVENTS_CONFIG.XML

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

Configuring Dimensions CM Communication in Release Manager

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

must specify the connection information through the Serena ALM Configurator **Dimensions CM** page.

Configuring Objects in Dimensions CM

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

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

- [Configuring the Dimensions CM Global Stage Lifecycle \[page 82\]](#)
- [Managing Dimensions CM Users \[page 82\]](#)
- [Configuring Dimensions CM Projects and Streams \[page 83\]](#)
- [Available Selection of Requests and Baselines \[page 84\]](#)

Configuring the Dimensions CM Global Stage Lifecycle

To make the deployment stages in Serena Release Manager 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 Manager stages for major, minor, and emergency release types. You can also configure Serena Release Manager 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" section.
- Complete documentation on configuring the deployment areas in Dimensions CM is in the *Dimensions CM Process Modeling User's Guide* in the "Area Definitions" section.

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 Manager users do not need to log into the Dimensions CM client. The interaction most Serena Release Manager 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 Manager 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 Manager power users or Serena Release Manager administrators.

What Can You Change in Release Manager?

Dimensions CM object changes are done in Dimensions CM.

What Can You Change in Dimensions CM?

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

What is the Impact?

- When a user is signed on through single sign-on to either SBM or Dimensions CM and accesses the other client through the Web interface, that user is automatically logged into the other product.
- If you don't use single sign-on for the administrative user used for the Serena Release Manager communication to Dimensions CM, the Serena Release Manager integration to Dimensions CM will not work as designed.

How Do You Change It?

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

Documentation References

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

Configuring Dimensions CM Projects and Streams

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

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

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

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

What Can You Change in Release Manager?

Dimensions CM object changes are done in Dimensions CM.

What Can You Change in Dimensions CM?

You can change the following project and stream information:

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

What is the Impact?

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

How Do You Change It?

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

Documentation References

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

Available Selection of Requests and Baselines

If Dimensions CM is used as one of your development change request providers, requests are displayed for selection in Serena Release Manager 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 Manager 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 [Provider Configuration \[page 121\]](#).

Configuring Dimensions CM for Development Manager

This chapter describes the configuration of Dimensions CM specific to Development Manager.

[Configuring the Dimensions CM Server for Development Manager \[page 85\]](#)

[Configuring Dimensions CM for ALF Events \[page 85\]](#)

[Assign Build Engineer Role \[page 86\]](#)

[Assign Development Engineer Role as Delegation Candidate \[page 88\]](#)

[Assign Development Engineer Role to TASK Lifecycle Transitions \[page 88\]](#)

[Setting Privileges for the Service User \[page 90\]](#)

[Configuring Dimensions CM for Products and Components \[page 92\]](#)

Configuring the Dimensions CM Server for Development Manager

So that Serena Development Control can get the information from Serena Dimensions CM, you must configure the information for your specific environment.

Configuring Dimensions CM for ALF Events

So that Serena Development Control can receive information from Serena Dimensions CM, you must update the Dimensions CM configuration settings with the necessary Application Lifecycle Framework (ALF) server information.

This is connecting SBM with Dimensions CM through ALF. Serena Development Control will get the information through SBM.

To update the Dimensions CM configuration ALF event settings:

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

```
..\Program Files\Serena\Dimensions 12.2\CM
```

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

```
#####
```

```
# ALF Events Configuration
```

```
#####
```

```
DM_ALF_ENDPOINT http://<SBM_server>:<port#>/eventmanager/services/ALFEventManager
```

```
DM_ALF_EVENT_CONFIG %DM_DFS%alf_events_config.xml
```

```
DM_ALF_PRODUCT_INSTANCE DimensionsUnderOrchestratedApps
```

where

- DM_ALF_ENDPOINT is pointing to the correct SBM server and port number and ALF Event Manager

Example

dm.cfg entry:

```
#####
```

```
# ALF Events Configuration
```

```
#####
```

```
DM_ALF_ENDPOINT http://sbmhost:8085/eventmanager/services/ALFEventManager
```

```
DM_ALF_EVENT_CONFIG %DM_DFS%alf_events_config.xml
```

```
DM_ALF_PRODUCT_INSTANCE DimensionsUnderOrchestratedApps
```

Updating the ALF Events Configuration Settings

So that Serena Development Control can receive information from Serena Dimensions CM, you must update the ALF configuration settings with the necessary information.

To update the Dimensions CM configuration settings:

1. Navigate to the Dimensions CM installation directory. For example:
`..\Program Files\Dimensions\12.1\CM\dfs`
2. Update the ALF_EVENTS_CONFIG.XML file to specify your Dimensions CM database name, and project name.

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@*-Dim10</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>QLARIUS:*</Name>
      <!--Specify one or more object classes-->
      <Objects>
<!-- Specify a specific object class -->

```

After completing these changes, restart the Serena Dimensions Listener service.

Assign Build Engineer Role

In the default application, members of the Build team need to have the role of BUILD ENGINEER assigned. The Build Manager, AMY, and the Development Engineer EMMA need to be authorised to be delegated the role BUILD ENGINEER with Capability SECONDARY in Dimensions CM.

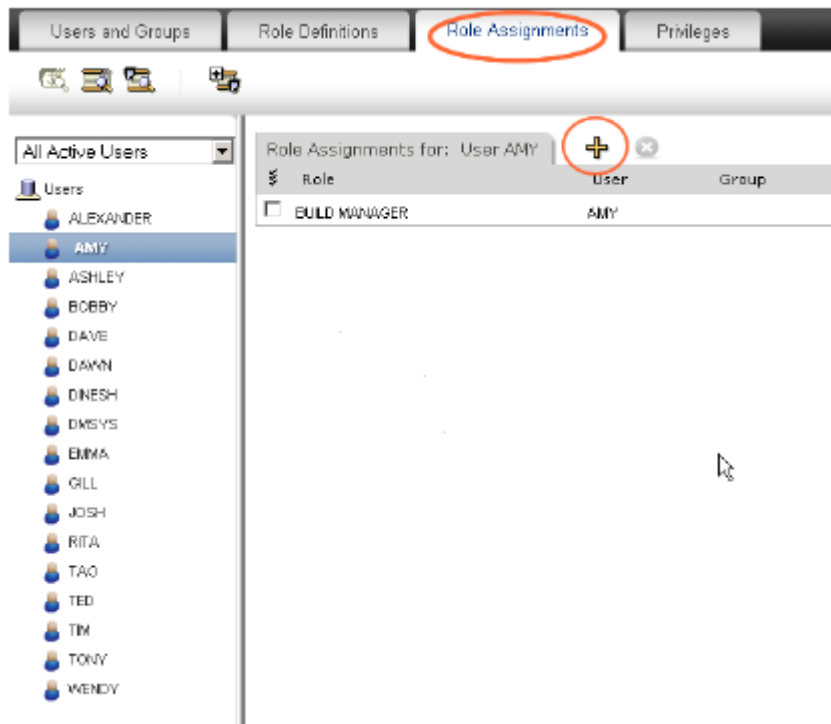
To do this:

1. Open the Dimensions CM Administration Console as a member of the Administration group, for example dmsys:

In a browser, enter for example the URL

`http://dvm_cmserver:8080/adminconsole`

2. Under **Users and Roles**, select **Role assignments | By user**



3. Select the user and select the Add button.

New Role Assignment OK Cancel Help
Keep open ☐

Roles can be assigned to users at Product or Design Part level, but can also be assigned to a specific design part variant and/or project. The following rules apply:

1. If no variant is specified, the assignment applies to all variants of the design part, except for any that have been separately assigned to other users.
2. If no project is specified, the assignment applies to all projects in the product, except for any that have been separately assigned for the design part to other users.
3. Specific role assignments against individual projects only apply to items.

Assignment Type: ☐ Real ☒ Delegation Candidate

Product Name: QLARIUS

User Or Group: AMY

☒ **Project Role:** BUILD ENGINEER

☒ **Design Part:** QLARIUS ... Variant:

☒ **Role Capability:** SECONDARY

Project Id:

Report: ☐

4. Set the following values:

- **Assignment Type:** Delegation Candidate.
- **Project Role:** BUILD ENGINEER
- **Design Part:** The top level design part for the project, for example QLARIUS

- **Role Capability:** Secondary

5. Click **OK**.

Assign Development Engineer Role as Delegation Candidate

In the default application, the Development Engineer, for example Emma, needs to be authorised to be delegated the role DEVELOPMENT ENGINEER with Capability SECONDARY.

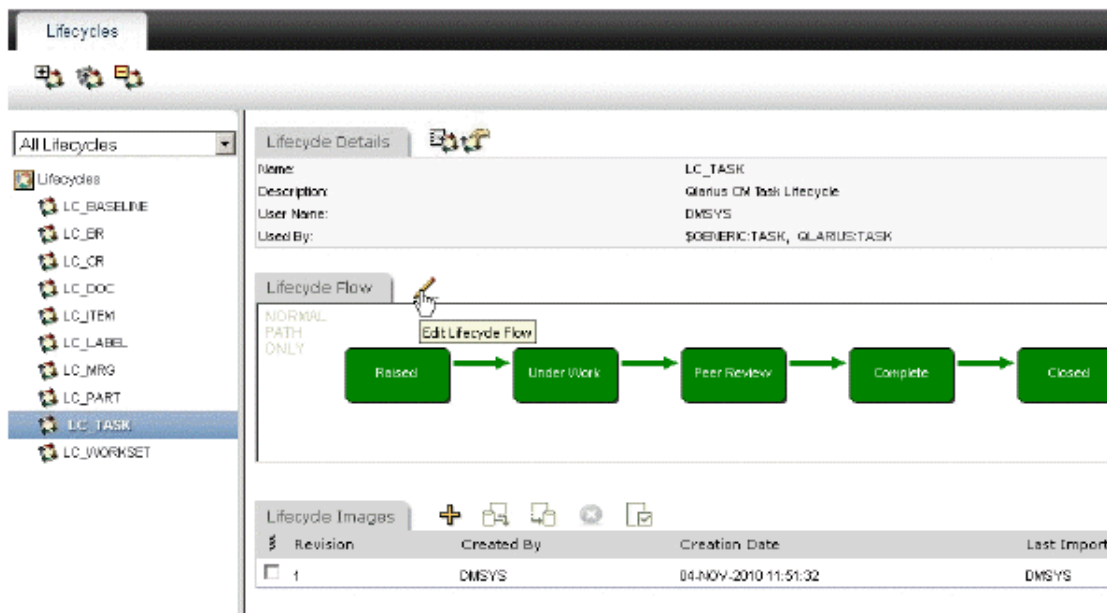
To do this:

1. Open the Dimensions CM Administration Console as a member of the Administration group.
2. Under **Users and Roles**, select **Role assignments | By user**
3. Select the user and select the **Add** button
4. Set the following values:
 - **Assignment Type:** Delegation Candidate.
 - **Project Role:** DEVELOPMENT ENGINEER
 - **Design Part:** The top level design part for the product, for example QLARIUS
 - **Role Capability:** Secondary
5. Click **OK**.

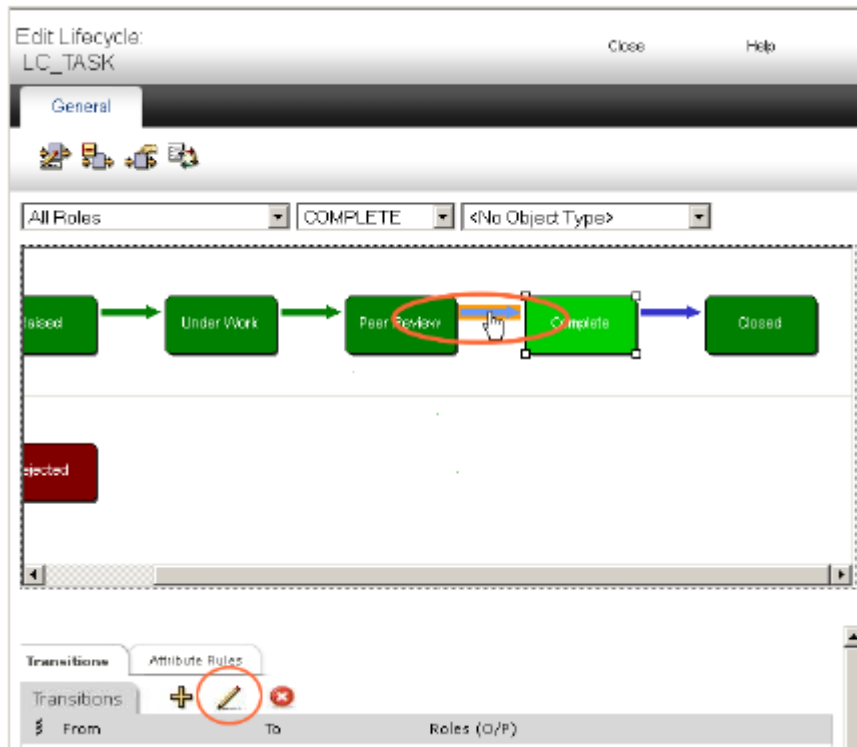
Assign Development Engineer Role to TASK Lifecycle Transitions

The transitions in the TASK lifecycle in Dimensions CM to *Complete* and *Closed* need to have DEVELOPMENT ENGINEER assigned as a Pending role.

1. Under Configuration Object Management, select Lifecycles.
2. Select LC_TASK.
3. Click the Edit button



4. Select the Complete state and select the transition arrow from Peer review to Complete

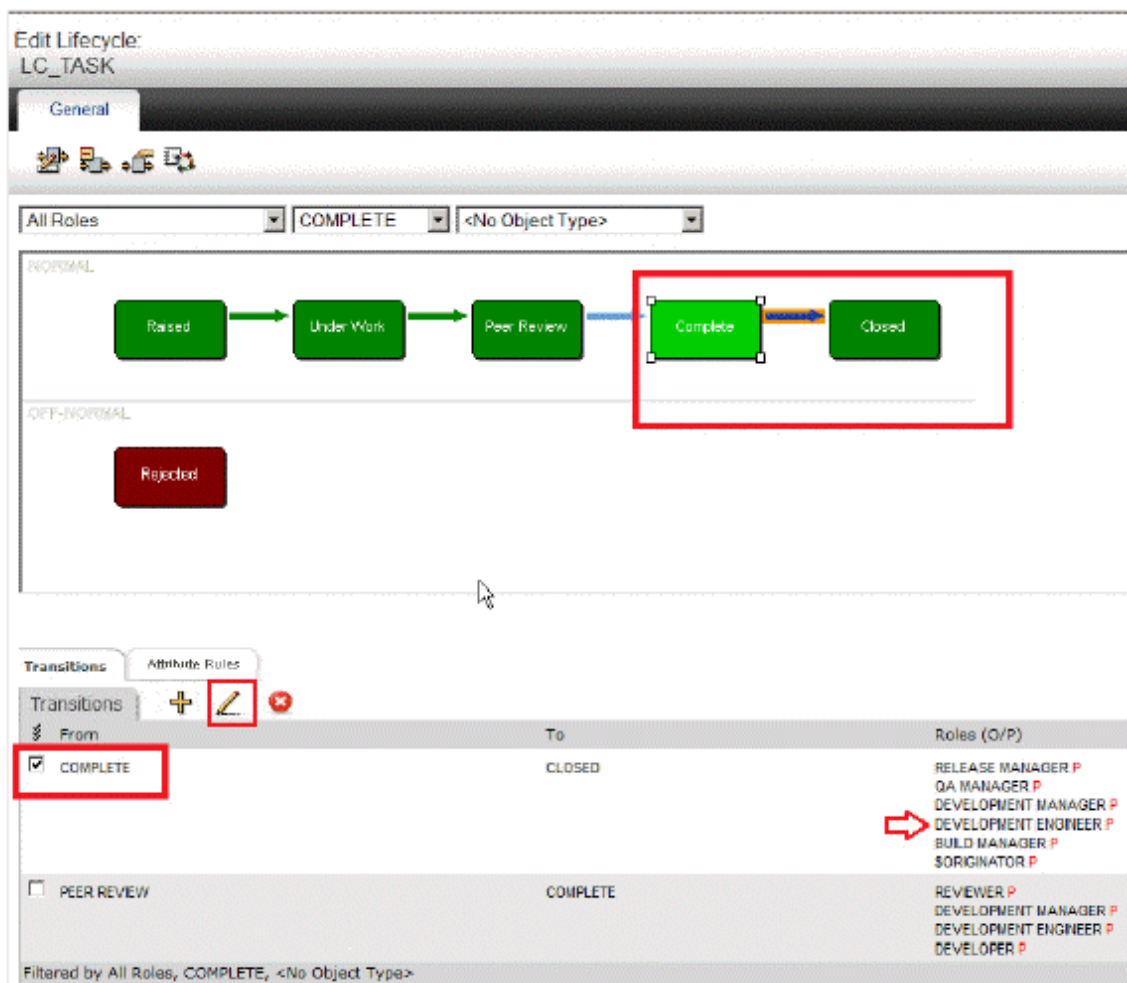


5. Click the Edit button on the Transitions tab.
6. Select DEVELOPMENT ENGINEER and select the Pending check box



7. Click OK.
8. Select the the transition arrow from Complete to Closed
9. Click the Edit button on the Transitions tab.
10. Select DEVELOPMENT ENGINEER and select the Pending check box

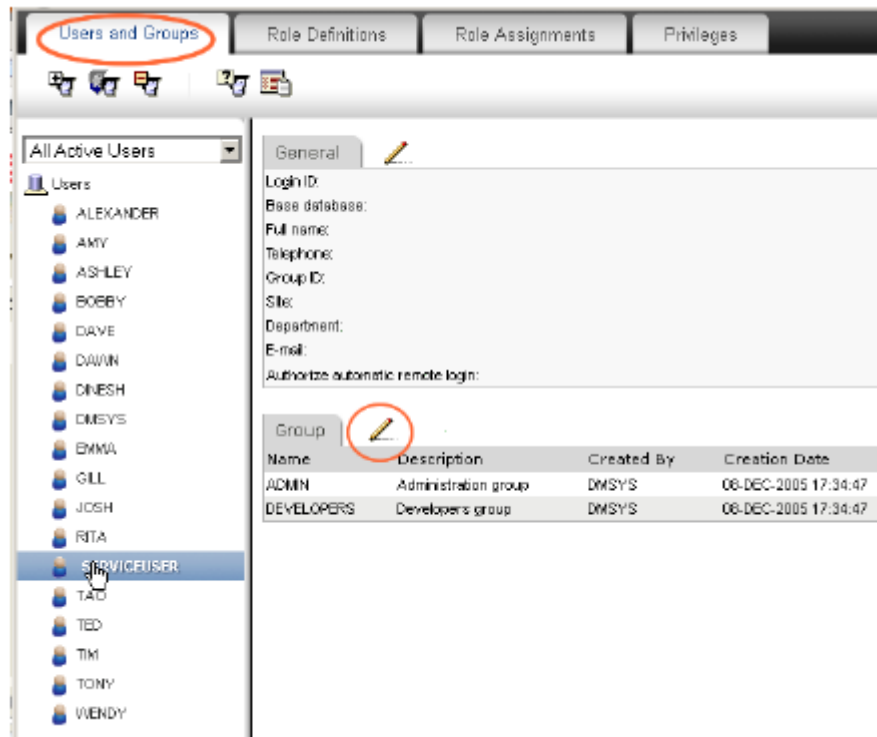
- Click OK.



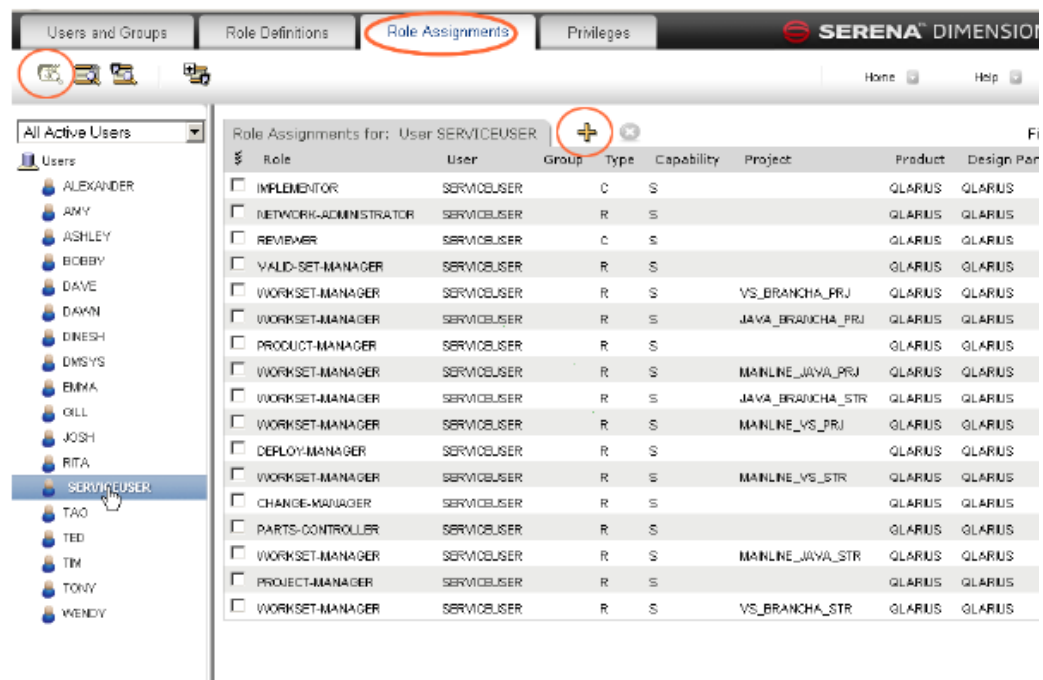
Setting Privileges for the Service User

The user that you are using in the process apps to connect with Dimensions CM, by default *serviceuser* needs to exist on the Dimensions CM server, and have the necessary roles and privileges to work with Development Manager.

- Open the Administration Console.
- Select Users and Groups.
- If the user does not exist, Click the Add button on the toolbar, and create a user with a name corresponding to the service user.
- Select the service user.



5. Click the Edit button and in the Edit User dialog, add the groups ADMIN and DEVELOPERS.
6. Click the Role Assignments Tab.
7. Click the Users button on the toolbar.
8. Select the user, for example SERVICEUSER.
9. Click the Add button.



10. Add the same role assignments as are assigned for user DMSYS.

Configuring Dimensions CM For Using Products and Components

If you are using products other than the QLARIUS product provided with the CM_TYPICAL sample database, there are a number of things you need to configure in Dimensions CM: These are:

- Ensure that any users who will link Products or Components in Development Manager with Design Parts in Dimensions CM are registered Dimensions users, for example, Eva the Component Manager in the sample ALM users.
- Ensure that the Dimensions CM design parts that you wish to link with Components exist in each new Dimensions CM product.
- Create a TASK request type in each new Dimensions CM product with the same LC_TASK lifecycle, and the synchronized attributes (SBM_ID, TITLE, ACTUAL_DEV_EFFORT, EST_DEV_EFFORT) as required by your Development Manager configuration.
- Assign the DEVELOPMENT ENGINEER role to appropriate users and/or groups who may own tasks within each new Dimensions CM product.
- Finally, ensure that the service user can create requests of type TASK in each new Dimensions CM product.

ChangeMan ZMF for Serena ALM Configuration

This section tells you how to configure and administer objects for communication and integration with ChangeMan ZMF.

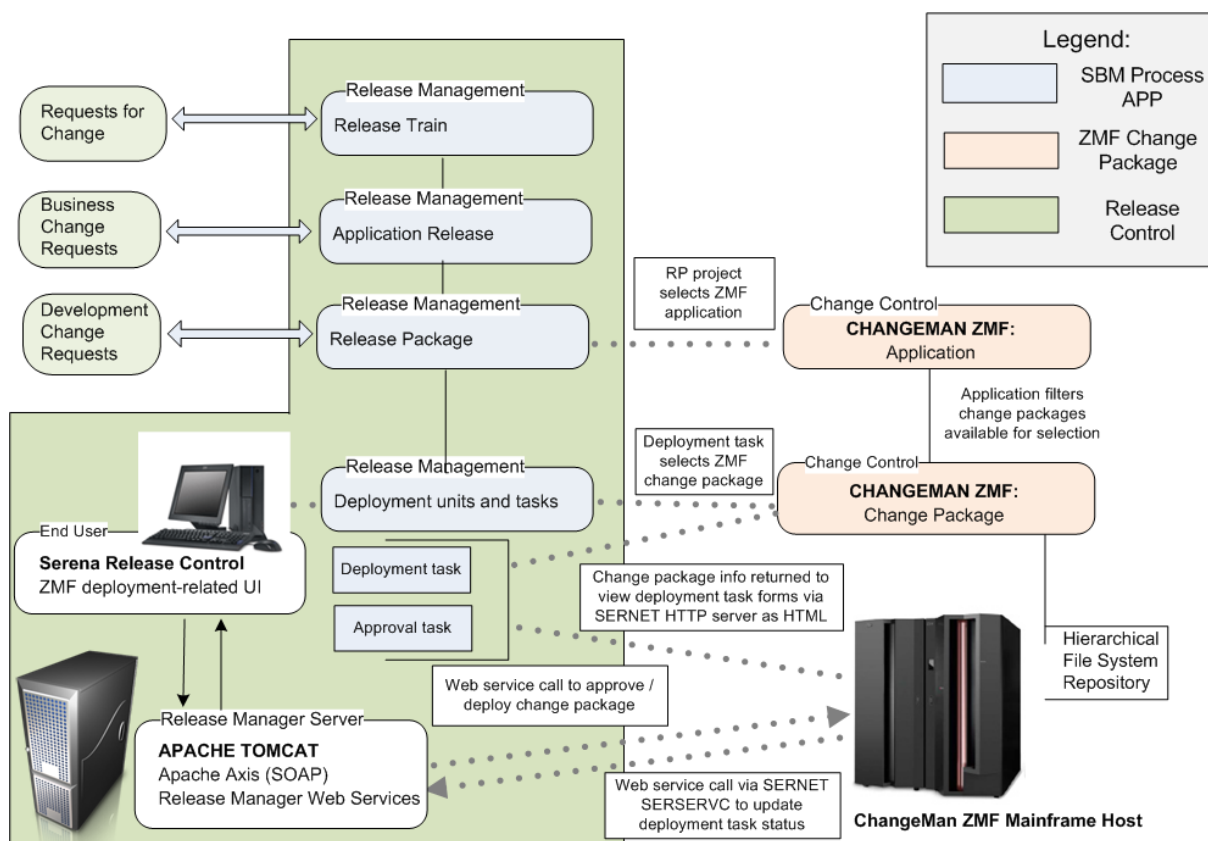
[ZMF Communication Configuration Overview \[page 93\]](#)

[Configuring ZMF Communication on the Mainframe \[page 93\]](#)

[Configuring ZMF Communication in Serena ALM \[page 105\]](#)

ZMF Communication Configuration Overview

You must configure ChangeMan ZMF communication on the z/OS mainframe and on the Serena ALM server to activate the integration. The architecture that supports the ZMF to Release Manager integration is shown in the following figure. The ZMF to Development Manager integration is not pictured, but has similar integration points.



Configuring ZMF Communication on the Mainframe

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

- Configure the NTFYURL; this is the URL Serena ALM uses to send information to SERNET through the server.

- Configure the SERNET HTTP server; this is the server Serena ALM uses to populate the UI widgets with ZMF information.
- Configure a proxy user ID for each mainframe host, or LPAR, that Serena ALM 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 ALM.

Configuring the Notification URL

You must give ChangeMan ZMF a way to tell Serena ALM 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 ALM when an ALF event is emitted from ChangeMan ZMF.

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

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

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

where hostname is the server name where Serena ALM 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='alm_host:8080/almzmfalf/services/ZMFALFEventRouter'
```

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

Documentation References

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

Configuring the SERNET HTTP Server

The SERNET HTTP Server provides an HTML interface for ChangeMan ZMF information.

NOTE The original version of the SERNET HTTP Server was written for use by ARM (Application Release Manager), an earlier version of Serena Release Manager. It is now used by Serena Release Manager, Serena Development Manager, and Serena Dashboard.

IMPORTANT!

- For the UI widgets to populate the forms with information from ZMF, the ZMF forms must be updated so that the widgets point to the correct hostname. See [Configuring the Release Manager ZMF Web Page Widgets \[page 107\]](#).
- The URL to this server is specified in the `zmf-client-connection.properties` file, which can be updated using the Serena ALM Configurator. See [Provider Configuration \[page 121\]](#).

ChangeMan ZMF Version Support

SERNET HTTP Server supports multiple ChangeMan ZMF versions as follows:

- It was originally packaged with ChangeMan ZMF 5.6 as the lowest common release.
- It supports all releases from ChangeMan ZMF 5.6 through 7.1.3 and later.
- The HTTP server can communicate with multiple servers at different version levels of ChangeMan ZMF, such as 6.1 and 7.1, at the same time and on different LPARs or remote CPUs.
- Only one server is needed for multiple ZMF levels and remote servers.

Architecture and Behavior

The architecture and behavior of SERNET HTTP Server is as follows:

- It is a native z/OS mainframe HTTP server than runs under SERNET.
- It is written in REXX.
- It runs under IBM's Unix System Services (USS) on the host, through BPXBATCH.
- It is true middleware.
- It listens for HTTP URL calls.
- It does code page translation from ASCII to EBCDIC.
- It supports UTF-8 character encoding only.
- It makes XML calls to servers.
- It answers in HTML formatted Web pages.
- It can deliver raw XML or JSON objects.
- It can perform ALL ZMF green Web services.
- It can be tested stand-alone from a Web browser.
- For PACKAGE PARMS it performs transformations. For example, packageStatus is returned as DEV instead of 6.
- It normalizes date and time formats.
- It does not issue any TSOe calls.

Installation

SERNET HTTP Server is installed using a z/OS job that copies components from PDS to USS files.

SERSERV Prerequisites

SERSERV has the following general requirements:

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

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

NOTE SERSERV works with versions of ChangeMan ZMF that may not be certified to work with Serena ALM and may be installed independently of Serena ALM.

SERNET User ID

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

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

`/u/userid`

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

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

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

Verifying SERNET User ID Privileges

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

To verify privileges for an existing SERNET user ID:

1. Find the current SERNET user ID.
 - a. Use SDSF to examine a running ChangeMan ZMF task or the JESMSGLOG of the job output from a previous execution of ChangeMan ZMF.
 - b. At the top of the message log, usually next to the message
`$HASP373 STARTED`
 find message IEF695I. This message includes an ASSIGNED message line that identifies the SERNET user ID.
2. Retrieve the OMVS segment for the SERNET user ID.
 - a. At the TSO command line, issue the following command:
`LU userid OMVS`
 where `userid` is the SERNET user ID found above, such as `sernet`.
 - b. If no OMVS segment is returned, ask your systems programmer or security administrator to add one. (See [SERNET User ID \[page 96\]](#) for required privileges.)
 - c. If an OMVS segment exists for the user ID, verify that the UID is set to zero and note the Unix home directory (for example, `/u/sernet`). If either are missing, ask your security administrator to add it to the OMVS segment.

PRIVILEGE

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

Installing SERSERV

To install SERSERV perform the following steps.

1. Create the SERSERV runtime directory with a path name of the form
`/u/userid/serserv`
 where `userid` is your actual SERNET user ID. (In our examples, this is `sernet`.)
 - a. Select ISPF option 3.17 to invoke the UNIX directory list utility.
 - b. In the **Pathname** field of the **z/OS Unix Directory List Utility** panel, type the SERNET home directory path (for example, `/u/sernet`). Leave the **Option==>** prompt blank to request a display of directory contents and press Enter.
 The directory list for the SERNET home directory displays.
 - c. In the **Unix Directory List** panel for the SERNET home directory, type the **N (New)** line command at the root level of the file hierarchy. The **Filename** for this level is listed as a single period (`.`) and the **Type** is "Directory" (Dir).
 For example:

[illegible]

Press Enter.

- d. When the **Create New z/OS UNIX File** window displays, create a new directory called `serservc` with permission level 755. Required field values are:
- **Pathname:** Give the full path name for the new directory `serservc`.
For example: `/u/sernet/serservc`
 - **Permissions:** Type 755 to assign read, write, and execute permissions over the directory to the directory owner, the owner's security group, and all others.
 - **File Type:** Type 1 to identify the new data object as a directory.

For example:

```

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

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

Permissions . . 755  (Octal)

Link . . . . .                             +

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

```

Press Enter and then exit the utility with PF3.

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

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

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

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

- a. For the PORT parameter, change the sample port number to the actual IP port assigned for the exclusive use of the SERSERV HTTP server.
- b. In the PATH parameter, replace the sample home directory, /u/sernet, with the actual Unix home directory you created for SERNET.

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

- c. For the PROCLIB parameter, replace the sample value USER.PROCLIB with the name of your actual installation PROCLIB.
4. Customize the JCL for the SERSERVI install job. Member SERSERVI resides in the CNTL library where you unloaded the ChangeMan ZMF SERCOMC installation libraries.

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

```
//jobcard JOB , 'USS JOB', CLASS=A,
```

```

//          NOTIFY=userid
// *
//STEP1    EXEC PGM=IKJEFT01,DYNAMNBR=200,COND=EVEN
//SYSTSPRT DD SYSOUT=*
//HFSOUT   DD PATH='/u/sernet/serservc/stdout',
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//          PATHMODE=(SIRWXU,SIRWXG,SIRWXO)
//HFSERR   DD PATH='/u/sernet/serservc/stderr',
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//          PATHMODE=(SIRWXU,SIRWXG,SIRWXO)
//EMPTY    DD *
/*
//NEWLOG   DD PATH='/u/sernet/serservc/msglog',
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//          PATHMODE=(SIRWXU,SIRWXG,SIRWXO)
//NEWHDR   DD PATH='/u/sernet/serservc/headers',
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//          PATHMODE=(SIRWXU,SIRWXG,SIRWXO)
//MSGARCH  DD *
*****  ***** SerServ *----- Archive Restart ---*
/*
//NEWSRV   DD PATH='/u/sernet/serservc/serserv',
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//          PATHMODE=(SIRWXU,SIRWXG,SIRWXO)
//NEWTASK  DD PATH='/u/sernet/serservc/sertask',
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//          PATHMODE=(SIRWXU,SIRWXG,SIRWXO)
//SERSERV  DD DISP=SHR,DSN=CMNPRD.CMN.SE56.#000031.REX(SERSERV)
//SERTASK  DD DISP=SHR,DSN=CMNPRD.CMN.SE56.#000031.REX(SERTASK)
//SERCMD   DD DISP=SHR,DSN=CMNPRD.CMN.SE56.#000031.CLS(SERCMD)
//NEWCMD   DD DISP=SHR,DSN=USER.SYS1.CLIST(SERCMD)
//SERVA    DD DISP=SHR,DSN=USER.PROCLIB(SERSERVA)
//SERVP    DD DISP=SHR,DSN=USER.PROCLIB(SERSERV)
//SYSPRINT DD SYSOUT=*
. . .

```

- a. Copy the SERSERVI JCL sample to a work library for editing.
- b. Edit the job card as needed.
- c. In each occurrence of the PATH parameter that is supplied to various job steps in this job, change the sample home directory name, /u/sernet, to the actual name of the Unix home directory you defined for SERNET.

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

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

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

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

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

5. Run SERSERVI.

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

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

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

Verifying the Installation of SERSERVIC

To verify the installation of SERSERVIC, do the following:

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

```
/S SERSERVIC
```

2. Ping SERSERVIC locally to verify that it is operational.

- a. At the TSO command line, enter:

```
SERCMD PING
```

- b. You should receive the response ok.

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

- a. From any Web browser, type

```
http://ip:port/?PING
```

where

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

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

- b. You should receive the response ok.

SERSERVIC Runtime Considerations

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

Startup and Shutdown

Console Commands

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

/S SERSERV

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

/C SERSERV

IPL Startup

However, SERSERV is designed for high availability. When testing is complete, consider adding SERSERV to the list of started tasks that are brought up at IPL time.

Orderly Shutdown

It does no harm to cancel SERSERV with a console command, but orderly shutdown is the preferred method. To initiate an orderly shutdown of the server, type

SERCMD SHUTDOWN

at the TSO command line. You should receive the response ok.

During an orderly shutdown, SERSERV copies its message log (msglog), error log (stderr), and standard output (stdout) to SYSOUT * before terminating execution.

Network Synchronization

SERSERV requires the local network time to be synchronized with server time on the host. Time zone differences of an integer number of hours are acceptable. However, the minutes and seconds (mm:ss) on the local network clock may not differ by more than 59 seconds from the minutes and seconds on the host clock.

Verifying Host Clock Time

To verify that local network time is synchronized with server time on the host, do the following:

1. From a Web browser running on the local network, type

`http://ip:port/?TIME`

where

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

port is the port number assigned to the exclusive use of SERSERV
2. You should receive the response hh:mm:ss, which is the time on the mainframe where SERSERV is running.
3. Discard the hours and compare the minutes and seconds on the host with the minutes and seconds reported on your local network. If a difference greater than 59 seconds is found, your local network time must be synchronized to the host.

The mainframe time is considered correct because its clock is built in at manufacture and cannot be changed.

Running Multiple Instances of SERSERV

You can run multiple instances of SERSERV if needed. For example, you may want to run a development HTTP server and a production HTTP server. Each instance must have its own directory and a unique port.

To run another instance of the HTTP server:

- Follow the installation instructions in [Installing SERSERV](#) [page 97], substituting another name for this instance, such as SERSERVD, in place of SERSERV.
- Specify a different directory and port for this instance according to your company's installation standards.

Here are examples of JCL segments for a production system and a development system.

- Production SERSERV example:

```
//SERSERV JOB MSGLEVEL=1
//STARTING EXEC SERSERV
XXSERSERV PROC OUTC=H, * CLASS
XX          PORT=8188, * PORT
XX          PATH='/u/serstart/serserv', * PATH
XX          PROCLIB='USER.PROCLIB' * PROCLIB
```

- Development SERSERV example:

```
//SERSERVD JOB MSGLEVEL=1
//STARTING EXEC SERSERVD
XXSERSERVD PROC OUTC=H, * CLASS
XX          PORT=6157, * PORT
XX          PATH='/u/sernet/serserv', * PATH
XX          PROCLIB='USER.PROCLIB' * PROCLIB
```

Documentation References

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

Configuring a ZMF 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 ALM server.

The purpose of the trusted user ID is to allow users to automatically access ChangeMan ZMF through Serena ALM 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 ALM 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 ALM *trusted resource* is not related to the RACF user ID TRUSTED attribute.

Configuring TSO User IDs and Permissions

All Serena ALM 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 ALM; however, when using Serena ALM, 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 ALM uses.

NOTE Serena ALM 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 ZMF Approvers

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

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

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

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

Configuring Objects in ChangeMan ZMF

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

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

- Applications (Projects)
- Sites (Environments)

- Change Packages
- Approver lists
- Promotion levels
- Audit return code rules

What Can You Change?

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

What is the Impact?

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

How Do You Change It?

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

Documentation References

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

Configuring ZMF Communication in Release Manager

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

- Specify the connection information. You do this through the Serena ALM Configurator **ZMF** page.
- Specify ALF event manager information for ChangeMan ZMF in the Serena Release Manager common Tomcat Web server webapps\almzmfalf\WEB-INF\conf folder `alfzmf_resource.properties` file.
- Specify other ZMF client-specific information in the Serena Release Manager common Tomcat Web server classes folder `zmf-client.properties` file.

Specifying ALF Event Manager Connection Information for ZMF

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

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

To update the ZMF ALF connection information:

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

`..\Program Files\Serena\common\tomcat\6.0\webapps\almzmfalf\WEB-INF\conf`
2. Open the `zmfalf_resource.properties` file.

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

- 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 = almadmin
AE_PASSWORD = almadmin_test
AE_VERSION = SBM 2009 R3
```

Specifying Client-Specific Information for ChangeMan ZMF

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

To specify the ZMF response status information:

- Navigate to the Serena ALM common Tomcat Web server classes folder. For example:
`..\Program Files\Serena\common\tomcat\6.0\webapps\alm\WEB-INF\classes`
- Open the zmf-client.properties file.
- Set the properties for the connection as follows:

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

- Restart the Serena common Tomcat service.

Example

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

zmf-client.properties

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

Configuring the Release Manager ZMF Web Page Widgets

If you are using ChangeMan ZMF for z/OS mainframe release management, you must configure your Serena Release Manager hostname in the Deployment Task state form widgets that display ZMF change package information. Otherwise, the ZMF information will not appear because the value localhost will only work if you are running the browser on the SBM machine itself.

Change the forms in the SBM Composer as shown in the following figure and procedure.

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

ZMF Widget Form Updates

The screenshot illustrates the SBM Composer interface with the following steps and details:

- I. Select a ZMF form:** The left pane shows a tree of forms under 'Visual Design'. The 'viewVaultZMFDeployTask' form is selected.
- II. Select the ZMF info tab:** The 'ChangeMan ZMF Package Info' tab is selected in the central pane.
- III. Select a sub-tab:** The 'General Info' sub-tab is selected.
- IV. Select the widget:** The 'ZMFPackageGenInfoWebPage' widget is selected in the 'Form Tools' palette.
- V. Change localhost to your hostname:** The 'Property Editor' at the bottom shows the 'URL' property for the 'ZMFPackageGenInfoWebPage' widget. The URL is:


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

 The 'localhost' part of the URL is highlighted for replacement.

To update the hostname in the Release Manager ZMF form widgets:

1. In SBM Composer, open the **Deployment** process app from the **Repository**. Choose **Overwrite** if prompted.
2. In **Visual Design**, check out and update the viewVaultZMFDeployTask and viewVaultZMFApprovalTask forms as follows:
 - a. Select the form and then select the ZMF Info tab, such as ChangeManZMFPackageInfo.
 - b. For each of the sub-tabs, **General**, **Description**, **AuditReport**, **Sites**, **Instructions**, **Components**, and **User Info**, select the Web page widget, such as ZMFPackageGenInfoWebPage.
 - c. In the Property Editor, in the URL for the widget, change localhost to your server hostname.
3. Save and check in the forms.

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

Configuring the Development Manager ZMF Web Page Widgets

If you are using ChangeMan ZMF as a provider for dev tasks, you must configure the hostname for the SBM server in the Dev Tasks state form widgets that display ZMF change package information, otherwise, the ZMF information will not appear because the value localhost will only work if you are running the browser on the SBM machine itself.

Change the forms in the SBM Composer as shown.

To update the hostname in the ZMF form widgets:

1. In SBM Composer, open the **Dev Tasks** process app from the **Repository**. Choose **Overwrite** if prompted.
2. In **Visual Design**, check out and update the **Task Base State Form** form as follows:
 - a. Select the form and then select the **Configuration Management** tab.
 - b. For each of the sub-tabs, General, Description, AuditReport, Sites, Instructions, Components, and User Info, select the Web page widget, for example, GeneralInfoWebPageWidget.
 - c. In the Property Editor, in the URL for the widget, change localhost to your SBM server hostname.
3. Save and check in the forms.

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

Release Automation for Serena ALM Configuration

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

[Release Automation Communication Configuration Overview \[page 109\]](#)

[Configuring Communication with Release Automation \[page 110\]](#)

[Configuring Communication with Release Automation \(Nolio\) \[page 114\]](#)

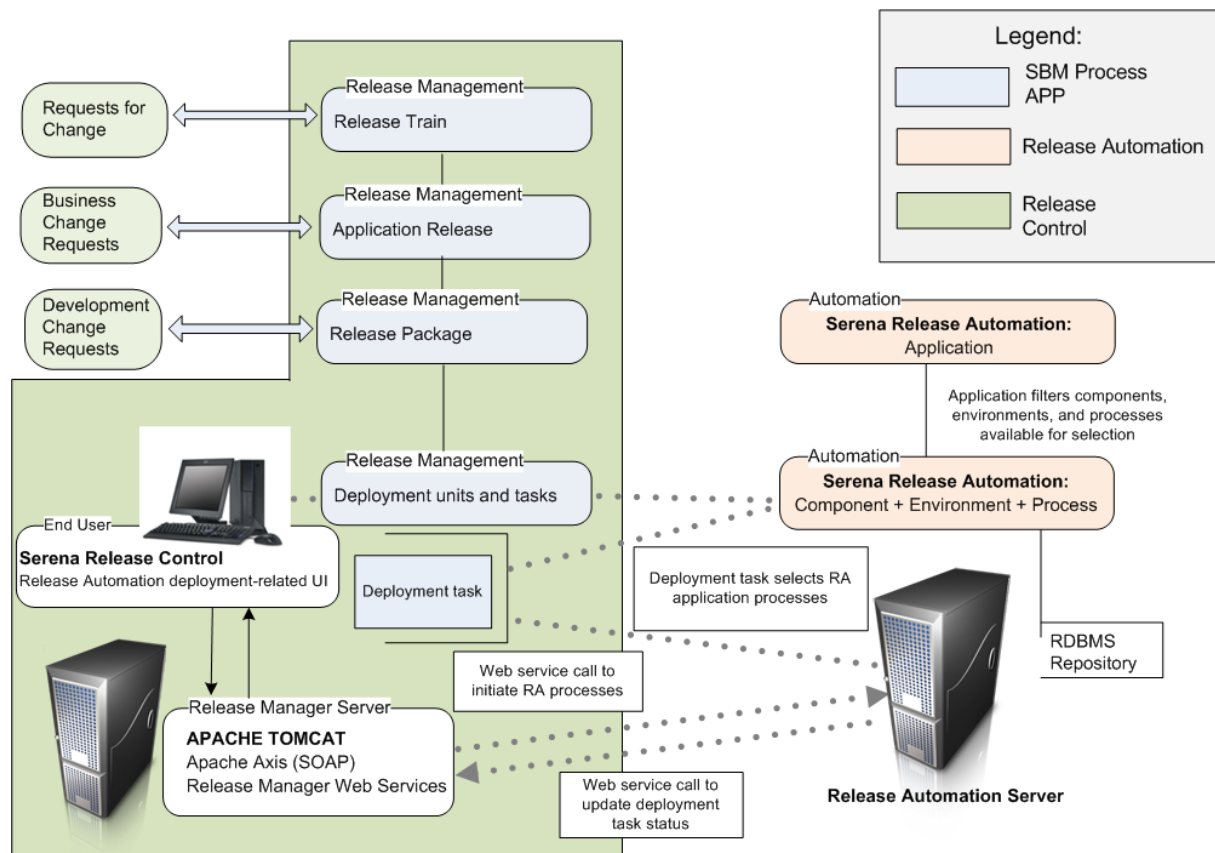
Release Automation Communication Configuration Overview

You must configure Serena Release Automation communication on the Serena Release Automation server and on the Serena Release Manager server to activate the integration.

Two Serena Release Automation options are available, Serena Release Automation and Serena Release Automation, powered by Nolio.

Release Automation Architecture

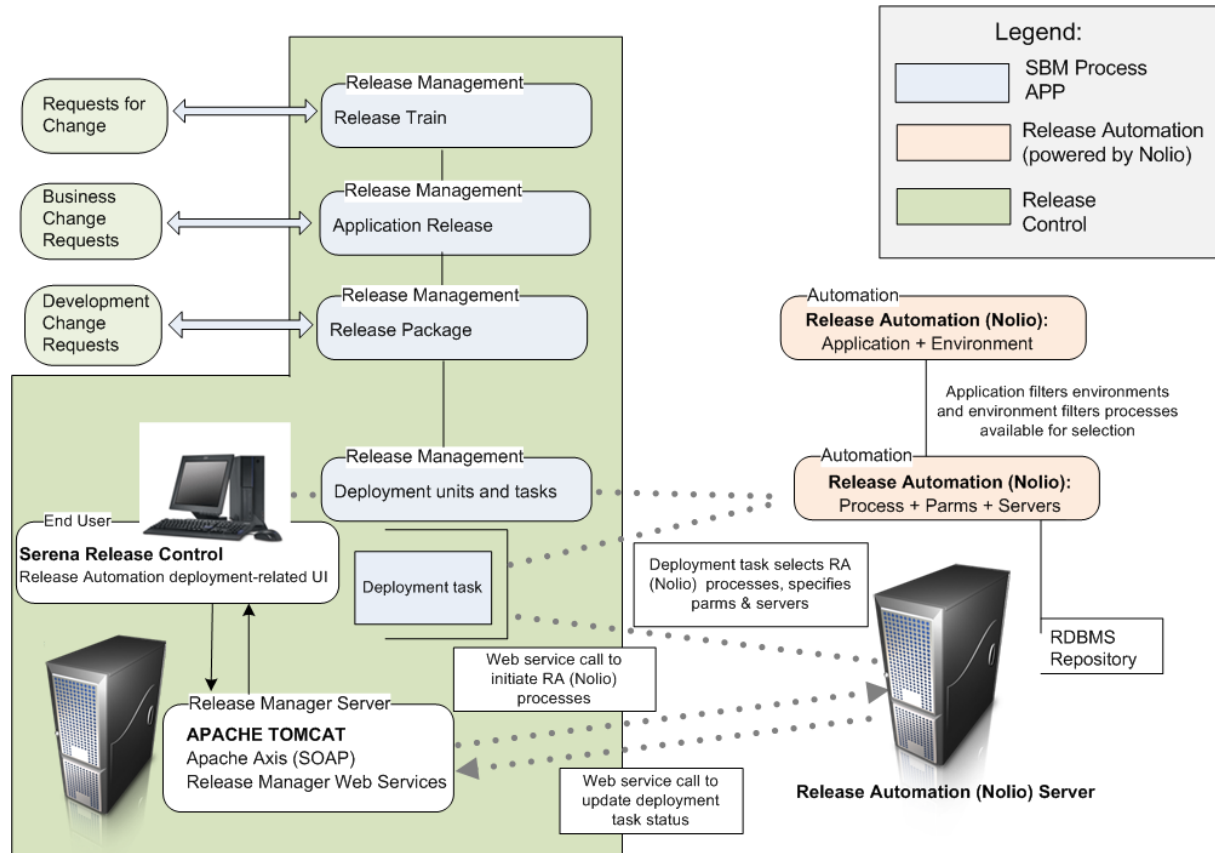
The architecture that supports the Serena Release Automation integration is shown in the following figure:



This figure shows how Serena Release Automation communicates with Serena Release Manager through automation deployment tasks and Web services calls. The automation deployment tasks workflows and forms are configured in the Deployment Tasks process app. Additional connection and client information are specified in the ALM Configurator and the client configuration files.

Release Automation (Nolio) Architecture

The architecture that supports the Serena Release Automation, powered by Nolio, integration is shown in the following figure:



This figure shows how Serena Release Automation communicates with Serena Release Manager through automation deployment tasks and Web services calls. The automation deployment tasks workflows and forms are configured in the Deployment Tasks process app. Additional connection and client information are specified in the ALM Configurator and the client configuration files.

Configuring Communication with Release Automation

If you are using Serena Release Automation, follow the documentation in this section to configure your Release Automation communication with Serena Release Manager.

Configuring Communication on the Release Automation Server

So that Serena Release Manager can get the information from Serena Release Automation, you must configure the communication on the Serena Release Automation server. This information is accessed through the Serena Release Automation Web services, and the Serena Release Automation server and agents must be installed and configured properly so that Serena Release Manager can access them across the network.

For details on installing and configuring Serena Release Automation, see the *Serena Release Automation User's Guide*.

Configuring Release Automation Communication in Release Manager

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

- Update the ALF sign-on credentials through the Serena ALM Configurator **ALF** page. Serena Release Automation uses these to send information to Serena Release Manager.
- Specify the Serena Release Automation client-specific information in the Serena Release Manager common Tomcat Web server classes folder `ra-client.properties` file.

Specifying Client-Specific Information for Release Automation

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

To specify the Serena Release Automation client-specific information:

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

```
..\Program Files\Serena\common\tomcat\6.0\webapps\alm\WEB-INF\classes
```

2. Open the `ra-client.properties` file.
3. Set the properties for the client as follows:

Parameter	Value
<code>ra.server.postDeployUrl</code>	URL to the server where the Serena ALM Web services are running. These are used to initiate the processes. The URL is specified in the form of: <code>http://almhost:8080/alm/servlet/ranotification/</code> You must specify the actual server name for the host; <code>localhost</code> is an invalid value.
<code>ra.server.postDeployMessage</code>	Expressions for the process status message, in the form of: <code>\${p:request.id}:\${p:finalStatus}</code>
<code>ra.server.childStatusPutUrl</code>	URL to the server where the Serena ALM Web services are running. These are used to post notification of the status of the processes. The URL is specified in the form of: <code>http://almhost:8080/alm/servlet/ranotification/</code> You must specify the actual server name for the host; <code>localhost</code> is an invalid value.
<code>ra.server.childStatusMessage</code>	Child status messages, in the form of: <code>requestId:\${p:parentRequest.id}</code> <code>component:\${p:component.name}</code> <code>resource:\${p:resource.name} status:\${p:status}</code> <code>result:\${p:result}</code>

Parameter	Value
ra.request.state.success	List of Serena Release Automation request states that indicate success, delimited by commas.
ra.request.states.failure	List of Serena Release Automation request states that indicate failure, delimited by commas.

Example

This example points to the Release Automation Web services and sets the values needed to retrieve successful and failed status notification.

ra-client.properties

```
ra.server.postDeployUrl=http://sbmhost:8080/alm/servlet/ranotification/
ra.server.postDeployMessage=${p:request.id}:${p:finalStatus}
ra.server.childStatusPutUrl=http://sbmhost:8080/alm/servlet/ranotification/
ra.server.childStatusMessage=requestId:${p:parentRequest.id}
    component:${p:component.name} resource:${p:resource.name} status:${p:status}
    result:${p:result}
ra.server.execution.status=EXECUTING,CLOSED
ra.server.execution.result=NONE,CANCELED,SUCCESSFUL,FAULTED
ra.request.state.success=success
ra.request.states.failure=failed
```

Configuring Objects in Serena Release Automation

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

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

- [Configuring Serena Release Automation Users \[page 112\]](#)
- [Configuring Serena Release Automation Processes and Servers \[page 113\]](#)

Configuring Serena Release Automation Users

Typically, Serena Release Manager users do not need to log into the Serena Release Automation client. The interaction most Serena Release Manager 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. This user must have privileges for the following in Serena Release Automation:
 - Accessing applications
 - Accessing environments
 - Accessing processes
 - Accessing components

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

What Can You Change?

- You can change user information in Serena Release Automation as needed for the users' roles in Serena Release Automation.
- Use caution when changing the administrative user ID that is used to sign on from Serena Release Manager as this user ID is used to execute the underlying integrative functions in Serena Release Automation.

What is the Impact?

- When you create an automation deployment task in Serena Release Manager, the administrative user credentials specified in the configuration files are used to sign on to Serena Release Automation.
- If the administrative user used to sign on to Serena Release Automation does not have proper privileges, the automation deployment tasks will fail.

How Do You Change It?

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

Documentation References

- Complete documentation on configuring users in Serena Release Automation is in the *Serena Release Automation User's Guide*.

Configuring Serena Release Automation Processes and Components

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

- Applications
- Environments
- Processes
- Components

Serena Release Manager uses the objects that you configure as part of the ongoing use of Serena Release Automation. Serena Release Manager filters the rest of the objects based on your selection of application.

Serena Release Automation automation deployment task association with Release Packages is used for the following purposes in Serena Release Manager:

- To automate the initiation of and complete the installation and configuration of components on designated environments after the deployment units are deployed by Dimensions CM.

What Can You Change?

- Serena Release Automation administrators can change objects in Serena Release Automation according to the Serena Release Automation documentation.

What is the Impact?

- When you create an automation deployment task in Serena Release Manager, you must select from the Serena Release Automation applications, environments, processes, and components 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.

How Do You Change It?

Serena Release Automation administrators should configure objects according to the Serena Release Automation documentation.

Documentation References

- Complete documentation on configuring objects in Serena Release Automation is in the *Serena Release Automation User's Guide*.

Configuring Communication with Release Automation (Nolio)

If you are using Serena Release Automation, powered by Nolio, follow the documentation in this section to configure your Release Automation (Nolio) communication with Serena Release Manager.

Configuring Communication on the Release Automation (Nolio) Server

So that Serena Release Manager can get the information from Serena Release Automation, powered by Nolio, you must configure the communication on the Release Automation (Nolio) server.

For Release Automation (Nolio), configure the files as follows:

- Specify the Release Automation (Nolio) 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 Release Automation (Nolio) environment notifications for each application to tell Serena Release Automation the events about which to notify Serena Release Manager.

Specifying the Release Automation (Nolio) Server to Notify

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

To specify the Release Automation (Nolio) server:

1. On the Release Automation (Nolio) server, navigate to the Release Automation (Nolio) installation directory. For example:

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

2. Open the Release Automation (Nolio) `rest.integration.properties` file.

If the file does not exist, create it.

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

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

For example:

```
target.url=http://almhost:8080/alm/servlet/NolioNotification
```

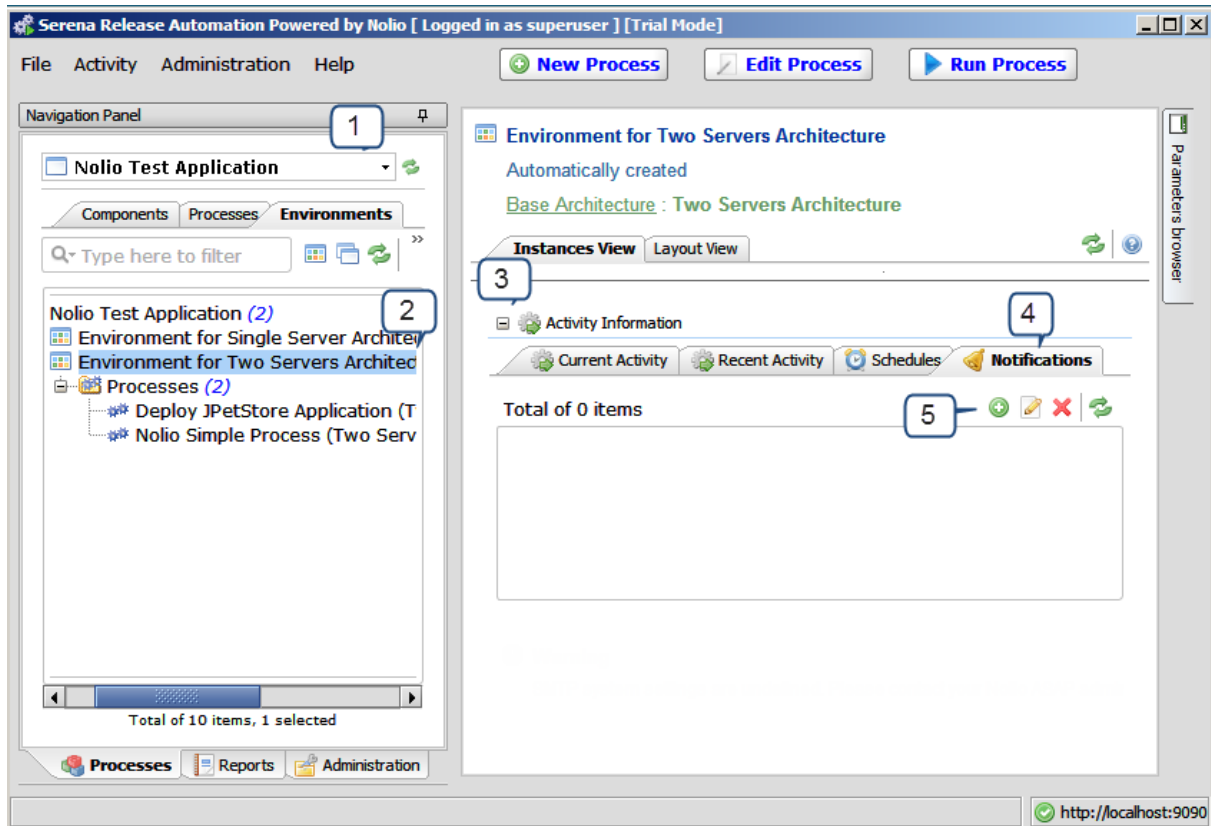
4. Restart the Nolio Server and Nolio Agent services.

IMPORTANT! Make sure to use the exact case given in [Step 3 \[page 114\]](#) for the NolioNotification servlet.

Telling Release Automation (Nolio) Which Event Notifications to Send

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

Release Automation (Nolio) is shown in the following figure.



To configure Release Automation (Nolio) environment notification:

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

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

New Notification

New Environment Notification
Set the required notification settings for this environment

Name

Notify about :
Execution Changes

States to notify about :

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

Notify these users :
superuser

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

Save Cancel

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

7. Click **Save**.

Configuring Release Automation (Nolio) Communication in Release Manager

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

- Update the ALF sign-on credentials through the Serena ALM Configurator **ALF** page. Release Automation (Nolio) uses these to send information to Serena Release Manager.
- Specify the Release Automation (Nolio) client-specific information in the Serena Release Manager common Tomcat Web server classes folder `nolio-client.properties` file.

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

Specifying Client-Specific Information for Release Automation

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

To specify the Serena Release Automation client-specific information:

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

```
..\Program Files\Serena\common\tomcat\6.0\webapps\alm\WEB-INF\classes
```

- Open the `nolio-client.properties` file.
- Set the properties for the client as follows:

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

Example

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

`nolio-client.properties`

```
nolio.job.runprocess.timeout=0
nolio.job.runprocess.wait=false
nolio.job.state.success = FLOW_FINISHED
nolio.job.states.failure = BLOCKED, CREATION_FAILED, FILES_DISTRIBUTION_FAILED,
FILES_PROPAGATION_FAILED, FLOW_FAILED_PAUSED, PRE_FAILED, FLOW_STOPPED
```

Specifying Release Automation (Nolio) Queries

You may specify SQL queries for the information you want returned from Release Automation (Nolio).

To specify the Release Automation (Nolio) connection information:

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

```
..\Program Files\Serena\common\tomcat\6.0\webapps\alm\WEB-INF\classes
```

- Open the `nolio-client-queries.properties` file.

3. Set the properties for the queries as follows:

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

Example

This example shows the default SQL queries for Release Automation (Nolio).

nolio-client-queries.properties

```
query.get.applications = select app_name from applications where id!=1
query.get.environments = select name from environments where id!=1 and
    applicationId = ?
query.get.processes = select process_name from process_container pc,
    process_in_env pe where pc.id = pe.process_id and pe.environment_id = ? order
    by process_name
query.get.servers = select s.server_name, st.server_type_name from servers s,
    server_types st, server_type_instance sti where s.id = sti.mapped_server and
    st.id = sti.server_type and sti.environment_id = ?
query.find.application = select id from applications where app_name = ?
query.find.environment = select id from environments where name = ? and
    applicationId = ?
query.get.job.status = select step_title,step_state from step_events where id in
    (select MAX(id) from step_events where job_id=?)
message.application.not.found = Could not find specified application: {0}
message.environment.not.found = Could not find specified environment: {0}
```

Configuring Objects in Release Automation (Nolio)

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

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

- [Configuring Release Automation \(Nolio\) Users \[page 119\]](#)
- [Configuring Release Automation \(Nolio\) Processes and Servers \[page 119\]](#)

Configuring Release Automation (Nolio) Users

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

There are two main types of Release Automation (Nolio) 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, powered by Nolio, through Web services and remote database access. This user must have privileges for the following in Release Automation (Nolio):
 - Accessing applications
 - Accessing environments
 - Accessing processes
 - Accessing servers
- Any users that have roles in both Release Automation (Nolio) 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 Release Automation (Nolio) as needed for the users' roles in Release Automation (Nolio).
- Use caution when changing the administrative user that is used to sign on from Serena Release Manager and execute the underlying integrative functions in Release Automation (Nolio).

What is the Impact?

- When you create an automation deployment task in Serena Release Manager, the super user credentials specified in the configuration files are used to sign on to Release Automation (Nolio).
- If the super user used to sign on to Release Automation (Nolio) does not have proper privileges, the automation deployment tasks will fail.

How Do You Change It?

Release Automation (Nolio) administrators should configure users in Release Automation (Nolio) according to the Release Automation (Nolio) documentation.

Documentation References

- Complete documentation on configuring users in Release Automation (Nolio) is in the *Serena Release Automation, Powered by Nolio, Installation and Administration Guide*.

Configuring Release Automation (Nolio) Processes and Servers

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

- Applications
- Environments
- Processes
- Servers

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

Release Automation (Nolio) 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 Release Automation (Nolio) to complete the installation and configuration of files on designated servers after the deployment units are deployed by Dimensions CM.

What Can You Change?

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

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

What is the Impact?

- When you create an automation deployment task in Serena Release Manager, you must select from the Release Automation (Nolio) applications, environments, processes, and servers that you have predefined in Release Automation (Nolio).
- 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 Release Automation (Nolio) for that process and server.

How Do You Change It?

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

Documentation References

- Complete documentation on configuring objects in Release Automation (Nolio) is in the *Serena Release Automation, Powered by Nolio, Installation and Administration Guide*.

Provider Configuration

Serena ALM providers are products that integrate or interface with Serena Release Manager and Serena Development Manager to provide access to information that you want to include as part of your orchestrated ALM solution. This section tells you how to configure existing provider connections. Provider connections enable the integration between products internal and external to the Serena Release Manager suite.

[Release Manager Provider Overview \[page 121\]](#)

[Development Manager Provider Overview \[page 131\]](#)

Provider Configuration Overview

Serena ALM providers are products that integrate or interface with Serena Release Manager and Serena Development Manager to provide access to information that you want to include as part of your orchestrated ALM solution.

Serena Release Manager includes connections to the following providers:

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

Serena Development Manager provides you with a choice of Configuration Management tools for managing dev tasks and dev packages. These are:

- Serena Dimensions CM
- Serena ChangeMan ZMF

You can use one or the other, or both of these.

NOTE This section tells how to configure connections to providers for which underlying integration layers have been implemented using the Serena ALM provider mechanism. For information on how to implement a new provider using this mechanism, see [Adding Provider Connections \[page 171\]](#).

Telling Release Manager Which Providers to Use

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

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

To specify the providers:

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

```
..\Program Files\Serena\common\tomcat\6.0\webapps\alm\WEB-INF\classes
```

2. Open the providers.properties file.

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 \[page 129\]](#).

NOTE

- Only one provider can be specified for RFCs and BCRs; more than one can be specified for DCRs and DUs.
- Dimensions CM is NOT implemented as a provider for RFCs and BCRs in the default implementation of Serena Release Manager; it is only implemented as a provider for DCRs and DUs.

IMPORTANT! If you do not want to use one or more of the provider types, simply leave the provider key value blank. Do NOT comment it out or delete it, or Serena Release Manager will no longer function properly.

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, dm_qlarius
# vault provider keys
vault.providers.keys=dm_qlarius, rlmzmf_packages, dvmzmf_packages
# rfc provider keys
rfc.providers.keys=itsm
# bcr provider keys
bcr.providers.keys=bcr
```

In the preceding example,

`requests.providers.keys = sbm_issues`

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

`vault.providers.keys = dm_qlarius, rlmzmf_packages, dvmzmf_packages`

tells Serena Release Manager to use the `dm_qlarius.properties` file, the `rlmzmf_packages.properties` file, and the `dvmzmf_packages.properties` file for release vault providers of deployment units

`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

Configuring Access to Requests for Change

Requests for Change (RFCs) represent operational changes that may affect multiple applications or implement system infrastructure changes within an enterprise. These are associated with release trains and are typically associated with tickets from service management systems such as Serena Service Manager.

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

Designate RFC provider information as follows:

1. [Designating the Details for Each RFC Provider \[page 123\]](#)
2. [Telling Release Manager Which Providers to Use \[page 121\]](#)

Designating the Details for Each RFC Provider

You should give the connection details for each RFC provider in a 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:

```
..\Program Files\Serena\common\tomcat\6.0\webapps\alm\WEB-INF\classes
```

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

```
itsm.properties
```

3. Specify the details in the properties file, leaving entries blank that you do not want to specify. Do NOT delete or mark entries as comments, as that may cause the provider communication to fail.

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

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

Designating RFC Provider Details for *Serena Service Manager (SSM)*

Example

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

```
itsm.properties
```

```
# rfc provider definitions
itsm.provider.name=Itsm
itsm.provider.description=ITSM Request Provider for Change system
itsm.table.tableName=TSM_CHANGEREQUEST
itsm.transition.update=CHANGE_MANAGEMENT.UPDATE1
itsm.transition.assignedRlm=CHANGE_MANAGEMENT.ASSIGNED_VIA_RLM
itsm.transition.implementedRlm=CHANGE_MANAGEMENT.IMPLEMENTED_VIA_RLM
itsm.transition.assignedRlm.type=Execute
itsm.transition.implementedRlm.type=Close
# rfc item fields
itsm.table.field.issueId=ISSUEID
itsm.table.field.state=STATE
itsm.table.field.relatedReleaseTrainId=LINKED_RELEASE
itsm.table.field.related=LINKED_TO_RELEASE
# rfc item states
itsm.defaultState=Approved Changes, Approved
```

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

- [Customizing the SSM Integration \[page 181\]](#)
- In the *Serena Service Manager User's Guide*, "Serena Release Manager Integration".

Configuring Access to Business Change Requests

Business Change Requests (BCRs) represent customer or business unit change requests that affect specific application releases. They are associated with application releases and are typically associated with tickets from help desk or incident management systems such as Serena Service Manager and other SBM solutions.

The default BCR provider connection provides an integration between a BCR in Serena Release Manager and a related issue in the SBM Issue Defect Management (IDM) solution.

Designate BCR provider information as follows:

1. [Designating the Details for Each BCR Provider \[page 124\]](#)
2. [Telling Release Manager Which Providers to Use \[page 121\]](#)

Designating the Details for Each BCR Provider

You should give the connection details for each BCR provider in a 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:

```
..\Program Files\Serena\common\tomcat\6.0\webapps\alm\WEB-INF\classes
```
2. Open one of your provider properties files. For example:

```
bcr.properties
```
3. Specify the details in the properties file, leaving entries blank that you do not want to specify. Do NOT delete or mark entries as comments, as that may cause the provider communication to fail.

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

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

Designating BCR Provider Details for *Serena Business Manager (SBM)*

Example

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

bcr.properties

```
# bcr provider definitions
bcr.provider.name=Business Change Request system
bcr.provider.description=Business Change Request system
bcr.table.tableName=TSM_CHANGEREQUEST
bcr.table.field.issueId=ISSUEID
bcr.table.field.state=STATE
bcr.defaultState=Approved Changes, Approved
```

Configuring Access to Development Change Requests

Development Change Request (DCRs) represent delivered changes from the development process. They are associated with release packages and are typically associated with change requests from systems used to manage development processes, such as Serena Service Manager, the SBM Incident Management solution, Serena Development Manager, and Serena Dimensions CM.

The default DCR provider connections are:

- For SBM:
 - An integration between a DCR in Serena Release Manager and a related incident in the SBM Incident Management solution.
 - An integration between a DCR in Serena Release Manager and a related issue in the SBM Issue Defect Management (IDM) solution.
- For Dimensions CM:
 - An integration between a DCR in Serena Release Manager and a related request in Dimensions CM.

Designate DCR provider information as follows:

1. [Designating the Details for Each DCR Provider \[page 125\]](#)
2. [Telling Release Manager Which Providers to Use \[page 121\]](#)

NOTE You must configure Dimensions CM communication with Serena Release Manager and configure the relevant SBM process apps to enable them to provide the DCRs.

Designating the Details for Each DCR Provider

You should give the connection details for each DCR provider in a 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:

`..\Program Files\Serena\common\tomcat\6.0\webapps\alm\WEB-INF\classes`
2. Open one of your provider properties files. For example:
 - `dm_qlarius.properties`
 - `sbm_issues.properties`
 - `sbm_incidents.properties`
3. Specify the details in the properties file, leaving entries blank that you do not want to specify. Do NOT delete or mark entries as comments, as that may cause the provider communication to fail.

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

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

Examples

- [Designating DCR Provider Details for Dimensions CM Requests \[page 126\]](#)
- [Designating DCR Provider Details for SBM Issues \[page 127\]](#)
- [Designating DCR Provider Details for SBM Incidents \[page 127\]](#)

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 = DCR_QLARIUS
requests.provider.description = Dimensions Requests Provider for QLARIUS product
# vault provider definitions
vault.provider.name = DIM_QLARIUS
vault.provider.description = Dimensions Vault Provider for QLARIUS product
# filter requests by statuses
FILTER_REQUEST_BY_STATUSES = IN QA,IN PROGRESS,UNDER WORK,IN TEST
# filter vault baselines by statuses
FILTER_VAULT_BASELINES_BY_STATUSES = CREATED
#
VAULT_TARGET_PRODUCT = VAULT
#
VAULT_TARGET_PROJECT = VAULT:V1
#
VALT_DML_STAGE = LIVE
```

The text following the keys, `requests.provider.name`, may be used in related UI report search filters. In this example, `DCR_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.

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

Designating DCR Provider Details for *SBM Issues*

Example

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

`sbm_issues.properties`

```
# requests provider definitions
requests.provider.name = DCR_Issues
requests.provider.description = SBM Requests Provider based on Issues solution
REQUESTS_TABLE_DBNAME=UBG_ISSUES
REQUESTS_FIELD_STATUS=STATE
REQUESTS_FIELD_LINK=URL
REQUESTS_FIELD_OWNER=OWNER
REQUESTS_FIELD_PROJECTNAME=PROJECTID
REQUESTS_QUERY_WHERE_CLAUSE =
REQUESTS_ORDER_BY_CLAUSE =
# possible values
# SUBMIT_PROJECTS
# REPORT_PROJECTS
REQUESTS_PROJECTS_TYPE= SUBMIT_PROJECTS
# valid only for REQUESTS_PROJECTS_TYPE = REPORT_PROJECTS
PROJECTS_REPORT_NAME=
PROJECTS_FIELD_TITLE=
PROJECTS_FIELD_STATUS=
PROJECTS_FIELD_OWNER=
PROJECTS_FIELD_TYPE=
# end of properties specific for REQUESTS_PROJECTS_TYPE = REPORT_PROJECTS
```

The text following the key `requests.provider.name` may be used in related UI report search filters. 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 key `requests.provider.name` may be used in related UI report search filters. In this example, `Incidents` simply describes the SBM primary table from which the incidents, or requests, are being retrieved. The `REQUESTS_TABLE_DBNAME` key specifies the primary table from which to retrieve the incidents.

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

Configuring Access to Deployment Units

Deployment Units (DUs) represent a set of deployable components. DUs are associated with release packages, and are typically associated with deployable components held in systems used to manage development processes, such as Serena Dimensions CM, Serena Development Manager, and Serena ChangeMan ZMF.

The default DU provider connections are:

- For Dimensions CM:
 - An integration between a DU in Serena Release Manager and a related baseline with build outputs in Dimensions CM.
- For ChangeMan ZMF:
 - An integration between a DU in Serena Release Manager and a related change package in ChangeMan ZMF.

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

1. [Designating the Details for Each DU Provider \[page 129\]](#)
2. [Telling Release Manager Which Providers to Use \[page 121\]](#)

NOTE You must configure Dimensions CM or ChangeMan ZMF communication with Serena Release Manager to enable them to provide the deployment units.

NOTE If you have Serena Development Manager (DVM) installed as part of the Serena ALM suite, your implementation of Dimensions CM is shared between Serena Release Manager and Serena Development Manager. Information that applies to Dimensions CM deployment unit provider configuration also applies to Serena Development Manager. For information on installing and configuring Serena Development Manager, see *Serena Development Manager Installation and Configuration*.

Designating the Details for Each DU Provider

You should give the connection details for each DU provider in a 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:

```
..\Program Files\Serena\common\tomcat\6.0\webapps\alm\WEB-INF\classes
```

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

```
dm_qlarius.properties
```

```
rlmzmf_packages.properties
```

3. Specify the details in the properties file, leaving entries blank that you do not want to specify. Do NOT delete or mark entries as comments, as that may cause the provider communication to fail.

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

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

Configuring Objects in Serena Business Manager

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

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

- Projects
- Items, such as issues or incidents

What Can You Change?

You can add or change the above information in the SBM process apps you are using, but not in Serena Release Manager. The only things that change in the SBM process apps from Serena

Release Manager should be a result of automations built into the release train workflow in relation to associated RFCs, application releases in relation to associated BCRs, or release packages in relation to DCRs.

What is the Impact?

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

How Do You Change It?

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

Documentation References

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

Configuring Objects in Serena Service Manager

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

- Projects
- Items, such as change requests

What Can You Change?

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

What is the Impact?

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

How Do You Change It?

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

Documentation References

- Complete documentation on submitting and actioning SSM change requests is in the *Serena Service Manager Concepts Guide* in "Change Management".
- Complete documentation on configuring SBM projects is in the *Serena Business Manager SBM 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".

Development Manager Provider Configuration Overview

Serena Development Manager provides you with a choice of Configuration Management tools for managing dev tasks and dev packages. These are:

- Serena Dimensions CM
- Serena ChangeMan ZMF

You can use one or the other, or both of these.

NOTE This section tells how to configure connections to providers for which underlying integration layers have been implemented using the Serena ALM provider mechanism. For information on how to implement a new provider using this mechanism, see [Adding Provider Connections \[page 171\]](#) .

Telling Development Manager Which Providers to Use

After you have defined the login information for the providers in the appropriate properties files as described above, you need to tell Serena ALM which providers you want to use by specifying their properties file names in the provider properties file.

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

To specify the providers:

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

```
C:\Program Files\Serena\common\tomcat\6.0\webapps\alm\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 name of the properties file (without the extension) corresponding to the provider you are using. For Development Manager these will normally be one or both of:

- dm_qlarius.properties
- zmf_packages.properties

IMPORTANT! If you do not want to use one or more of the provider types, simply leave the provider key value blank. Do NOT comment it out or delete it, or Serena Release Manager will no longer function properly.

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, dm_qlarius

# vault provider keys
vault.providers.keys=dm_qlarius, rlmzmf_packages, dvmzmf_packages

# rfc provider keys
rfc.providers.keys=itsm

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

For example, in the preceding example,

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

tells Serena Release Manager to use the `sbm_issues.properties` file and the `dm_qlarius.properties` file for Development Change Requests.

Development Manager Dev Task Provider Details

You should specify the details for each Dev Task provider in a separate properties file for each instance. This keeps the details separate from the selection, and helps with maintenance and security. By default

To designate details for each Dev Task provider:

1. Navigate to the Serena common Tomcat Web server classes folder. For example:
`C:\Program Files\Serena\common\tomcat\6.0\webapps\alm\WEB-INF\classes`
2. Open one of your provider properties files. For example:
 - `dm_qlarius.properties`
 - `dvmzmf_packages.properties`
3. Specify the details in the properties file, leaving entries blank that you do not want to specify. Do NOT delete or mark entries as comments, as that may cause the provider communication to fail.

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

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

CM for Development Manager Request 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

This example sets the values needed to retrieve development change requests and vault 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

# vault provider definitions
vault.provider.name = DIM_QLARIUS
vault.provider.description = Dimensions Vault Provider for QLARIUS product

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

# filter vault baselines by statuses
FILTER_VAULT_BASELINES_BY_STATUSES = CREATED

#
VAULT_TARGET_PRODUCT = VAULT
#
VALT_DML_STAGE = LIVE
```

The text following the keys, `requests.provider.name` and `vault.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 `dm_client_connections.properties` file. See [Dimensions CM Client Connection Information \[page 56\]](#).

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

ZMF for Development Manager Package Selection Criteria

If you plan to use ChangeMan ZMF to process change packages as your development tasks, you must specify the selection criteria for the change packages. Set the ChangeMan ZMF provider details for change packages, using the example `dvmzmf_packages.properties` file or a custom ZMF provider properties file. The default settings in the `dvmzmf_packages.properties` file are the settings you should normally use.

Set the properties for the ZMF filters as follows:

Parameter	Value
<code>vault.dvm.zmf.provider.name</code>	Vault provider name. Default value = <code>DVMZMF_PACKAGES</code>
<code>vault.dvm.zmf.provider.description</code>	Vault provider description. Default value = ChangeMan Vault Baseline Provider for ZMF packages
<code>FILTER_ZMF_PACKAGES_IN_FROZEN_STATUS</code>	Setting this to Y will list all packages that are in FRZ and DEV status. Setting this value to N will list only packages that are in DEV status. Default value = N

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_DVM_DT_ITEMID_TO_ZMF_PACKAGE_WORKREQNO below.</p> <p>NOTE Be aware of the issue It is Possible to Relate a ZMF Change Package to Multiple Dev Tasks (DEF224772) [page 209].</p>
FILTER_PLANNED_PERMANENT_ZMF_PACKAGES	<p>Include change packages with package type planned permanent. (Values Y or N) This should be set to Y.</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 = N</p>
FILTER_UNPLANNED_PERMANENT_ZMF_PACKAGES	<p>Include change packages with package type unplanned permanent. (Values Y or N)</p> <p>Default value = N</p>
FILTER_UNPLANNED_TEMPORARY_ZMF_PACKAGES	<p>Include change packages with package type unplanned temporary. (Values Y or N)</p> <p>Default value = N</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 = N</p>

<p>FILTER_ZMF_PACKAGES_BY_PROMOTION_LEVEL</p>	<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> <p>10 – INT Integration Test</p> <p>20 – UAT User Acceptance Test</p> <p>30 – PAT Production Acceptance Test</p> <p>Default value = blank</p>
<p>FILTER_ZMF_PACKAGES_BY_AUDIT_LEVEL</p>	<p>Include audit return code.</p> <p>The audit return code must be <i>less than or equal to</i> the audit level filter.</p> <p>Default value = blank</p>
<p>RELATE_DVM_DT_ITEMID_TO_ZMF_PACKAGE_WORKREQNO</p>	<p>Choose whether to fill in the ZMF change package work request number with the development change request value from Serena Development Manager. (Values Y or N)</p> <p>Default value = Y</p> <p>Note: Whatever the setting is for this parameter, you will need to make sure that the value of:</p> <p>RELATE_RLM_DU_ITEMID_TO_ZMF_PACKAGE_WORKREQNO</p> <p>in the rlmzmf_packages.properties file</p> <p>is set to the opposite value if you are using both providers, otherwise they will overwrite each-other's values in this field.</p> <p>For example, if</p> <p>RELATE_DVM_DT_ITEMID_TO_ZMF_PACKAGE_WORKREQNO = Y</p> <p>then set</p> <p>RELATE_RLM_DU_ITEMID_TO_ZMF_PACKAGE_WORKREQNO = N</p>
<p>RELATE_DVM_DT_INSTALL_DATE_TO_ZMF_PACKAGE_INSTALL_DATE</p>	<p>Choose whether to fill in the ZMF change package installation date with the development task date from Serena Development Manager. (Values Y or N)</p> <p>Default value = Y</p>

Example

This example sets the values needed to retrieve change packages related to dev tasks from a ChangeMan ZMF sample system.

dvmzmf_packages.properties

```
# Vault provider name and description for DVM.
vault.dvm.zmf.provider.name = DVMZMF_PACKAGES
vault.dvm.zmf.provider.description = ChangeMan Vault Baseline Provider for ZMF
packages

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

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

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

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

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

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

FILTER_ZMF_PACKAGES_BY_AUDIT_LEVEL =

# Fill in the ZMF change package work request number with the Development Manager
  deployment unit's SBM item ID.
RELATE_DVM_DT_ITEMID_TO_ZMF_PACKAGE_WORKREQNO = Y

# Fill in the ZMF change package installation date with the development task
  install date from Development Manager.
RELATE_DVM_DT_INSTALL_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-client-connection.properties` file. See [ZMF Client Connection Information \[page 58\]](#).

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 Development Manager.

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

Vault Request Configuration

This section tells you how to configure vault requests and vault templates. Vault requests and vault templates enable you to bring component sets from external systems into the Serena Release Manager suite. See the following sections for details.

[Vault Request Purpose \[page 138\]](#)

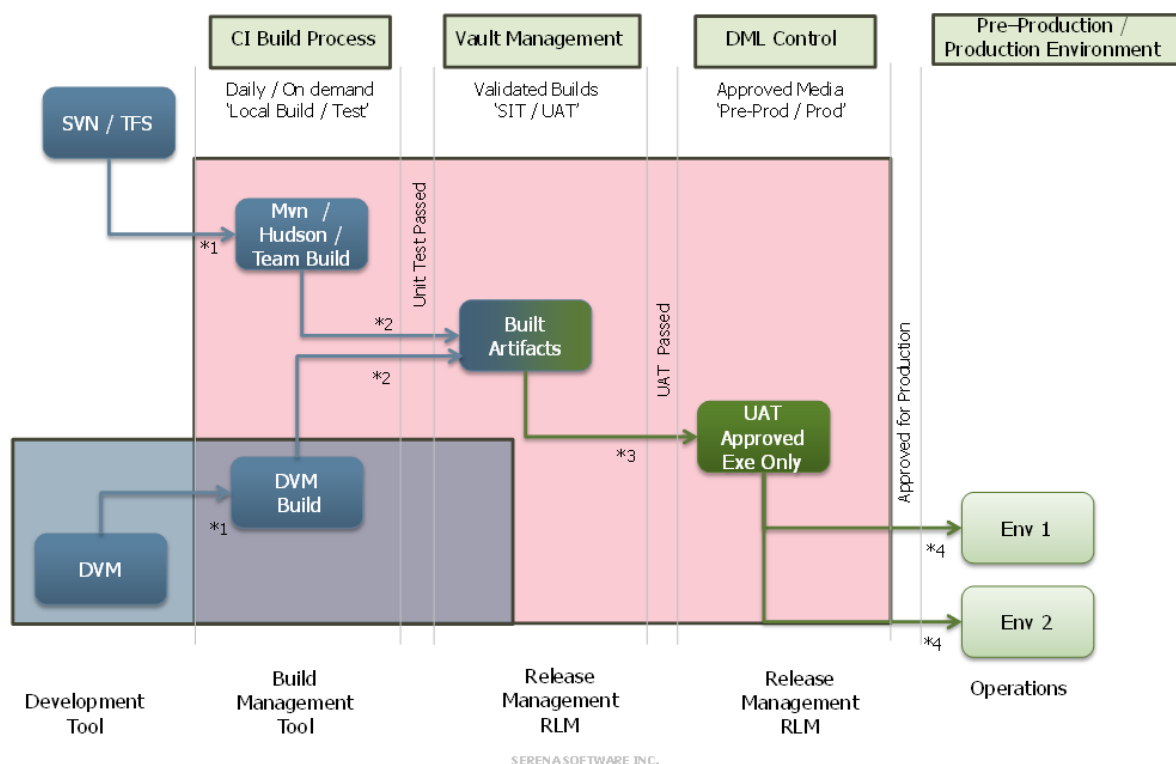
[Vault Request Architecture \[page 139\]](#)

[Vault Template Purpose \[page 139\]](#)

Vault Request Purpose

Serena Release Manager users, such as Release Managers or Release Engineers, use the vault request to add component sets to the Serena Release Manager release vault from external sources, such as other development tools, build management tools, and from existing baselines within the same instance of Dimensions CM or Serena Development Manager that they are using with Serena Release Manager.

The process of adding component sets, or built artifacts, into Serena Release Manager from external sources is shown in the following figure.



This figure shows how configuration items (CI) that have been built into sets of build artifacts are introduced into Serena Release Manager (RLM) from development tools and build managements tools, in this example Serena Development Manager (DVM) and Hudson / Team Build respectively.

After the built artifacts are tested through the Serena Release Manager release management stages and pass to the final stage, such as **Approved Exe, Production, or Live**, they are flagged as definitive media. Built artifacts can also brought directly in as definitive media, such as when you want to bring pre-tested vendor software into your Definitive Media Library (DML). Once the

definitive media is in the vault, it is considered part of your DML and is ready for IT operations to distribute or release as part of the IT operations processes in your organization.

Vault Request Architecture

The vault request implementation in Serena Release Manager is done through integration with the Dimensions CM system that provides the release vault functionality.

NOTE If you have Serena Development Manager (DVM) installed as part of the Serena ALM suite, your implementation of Dimensions CM is shared between Serena Release Manager and Serena Development Manager. Information that applies to Dimensions CM deployment unit provider configuration also applies to Serena Development Manager. For information on installing and configuring Serena Development Manager, see *Serena Development Manager Installation and Configuration*.

The Vault in a Vault Request

The "vault" in this context is a Dimensions CM product that must be pre-defined and specified your release vault provider configuration file. It is a special use of the Dimensions CM release vault that stores those built artifacts that are considered release or distribution-ready. These may need to be sent through the final testing process or may be ready to store as definitive media.

In the default implementation of Serena Release Manager, the example product ID defined in the `dm_qlarius.properties` file is `VAULT`, but you can name the product ID whatever makes sense for your implementation of the Vault Request functionality.

A source baseline either already exists in Dimensions CM or must be created temporarily in Dimensions CM when items are introduced from a work area. Once a source baseline exists, this baseline is copied to the target baseline based on the criteria specified in the vault request.

Definitive Media Library (DML)

The Definitive Media Library (DML) in Serena Release Manager is a set of release vault baselines that are flagged as definitive media. Any baselines flagged as definitive media are considered ready for IT operations to release or distribute according to your organization's processes.

Vault requests are used to introduce production-ready component sets into Serena Release Manager either directly as definitive media or to be tested as part of your organization's defined testing stages and then flagged as definitive media.

In the default implementation of Serena Release Manager, baselines tested through the release vault in Serena Release Manager are automatically marked as definitive media upon deployment to the Production, or Live, stage.

Vault Template Purpose

Serena Release Manager users, such as Release Managers or Release Engineers, set up the vault templates in Serena Release Manager and Serena Development Manager users, such as Development Managers, use the vault template to add component sets, or built artifacts, to the Serena Release Manager release vault directly from Serena Development Manager Development Packages.

The creation of a vault template in Serena Release Manager is very similar to that of the vault request, except that it doesn't have the source information filled in. It has only the target and options filled in, and the source is filled in from the Serena Development Manager side.

The process of adding component sets, or built artifacts, into Serena Release Manager from Serena Development Manager using vault templates is documented in *Serena Development Manager Getting Started Guide*.

See the Vault Template process app and its related workflow and form to see the Serena Release Manager side of the Vault Template implementation.

Release Manager Upgrade Scenarios

For an upgrade from Release Manager version 4.0 to version 4.0.1, see [Serena Release Manager Typical Upgrade \[page 140\]](#).

For an upgrade from Release Manager versions earlier than 4.0 to version 4.0.1, see [Serena Release Manager Manual Upgrade \[page 140\]](#).

Serena Release Manager Typical Upgrade

This section leads you through a typical upgrade of Serena Release Manager. These instructions should be used only when upgrading from Release Manager version 4.0 to version 4.0.1.

For an upgrade from Release Manager versions earlier than 4.0 to version 4.0.1, see [Serena Release Manager Manual Upgrade \[page 140\]](#).

To perform a typical upgrade of Release Manager:

1. Document custom changes.

If you have a customized version of Release Manager, you should carefully document the changes you made in each process app before beginning your upgrade so that you can reapply the changes as needed after the upgrade.

2. Back up your data and product files.

It is a good practice to snapshot your system or back up existing files and data before beginning the upgrade.

At minimum, you should back up the data as follows:

- a. Copy the files from the Serena Common Web services alm\WEB-INF\classes folder to a temporary folder.

For example:

```
..\Program Files\Serena\common\tomcat\6.0\webapps\alm\WEB-INF\classes
```

- b. Backup your SBM database.

- c. For an easier restoration if necessary, back up the product directories:

- i. Copy the Serena Common Web services webapps folder to a temporary folder.

For example:

```
..\Program Files\Serena\common\tomcat\6.0\webapps
```

- ii. Copy the Serena\Solutions folder to a temporary folder.

For example:

```
..\Program Files\Serena\Solutions
```

3. Follow the [Installation and Configuration Checklist \[page 18\]](#), taking note of the section [Which Process Apps Should I Promote and Deploy? \[page 29\]](#)

Serena Release Manager Manual Upgrade

This section leads you through a manual upgrade of Serena Release Manager. These instructions should be used only when upgrading using the Serena ALM installer are not appropriate, such as when you are upgrading from a version of Release Manager prior to version 4.0. to version 4.0.1.

For a typical upgrade from Release Manager version 4.0 to version 4.0.1, see [Serena Release Manager Typical Upgrade \[page 140\]](#).

- 1: Document Custom Changes [\[page 142\]](#)
- 2: Backup [\[page 142\]](#)
- 3: Copy the Upgrade Package [\[page 143\]](#)
- 4: Configure Web Services [\[page 143\]](#)
- 5: Install the new Serena Common Tools [\[page 144\]](#)
- 6: Copy the Solution File [\[page 144\]](#)
- 7: Install the User Interface Files and Report Templates [\[page 145\]](#)
- 8: Import the Solution [\[page 145\]](#)
- 9: Promote the Snapshots [\[page 145\]](#)
- 10: Configure the Process Apps [\[page 145\]](#)
- 11: Set Privileges [\[page 145\]](#)
- 12: Verify Serena ALM is activated [\[page 145\]](#)
- 13: Refresh the Web browser data [\[page 146\]](#)
- 14: Update the Registry [\[page 146\]](#)
- 15: Configure Release Vault [\[page 146\]](#)
- 16: Configure Release Automation [\[page 146\]](#)
- 17: Configure Connections and Providers [\[page 146\]](#)

Upgrading Serena Release Manager Manually

This section gives the manual instructions for upgrading Serena Release Manager in non-typical situations. For typical upgrades, see [Serena Release Manager Typical Upgrade \[page 140\]](#).

If you are not sure what upgrade path you should follow, please check with Serena Support.

IMPORTANT! If your configuration co-locates Serena ALM with Dimensions CM 12.2.1.1 or older, Dimensions CM must be upgraded before installing Serena ALM 4.0.1. The Serena ALM 4.0.1 installation includes a new version of the Serena Common Tomcat Web server that is not backwards compatible with earlier versions of Dimensions CM.

If you upgrade SBM to a version that is not supported by your version of Serena ALM, Serena ALM will no longer function properly. Please see the supported platforms for your version of Serena ALM as instructed in [Software Compatibility Requirements \[page 15\]](#).

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

Step	Actions
1: Document Custom Changes	If you have a customized version of Release Manager, you should carefully document the changes you made in each process app before beginning your upgrade so that you can reapply the changes as needed after the upgrade.
2: Backup	<p>Back up data and product files. (Optional)</p> <p>It is a good practice to snapshot your system or back up existing files and data before beginning the upgrade.</p> <ol style="list-style-type: none"> 1. At minimum, you should back up the data as follows: <ol style="list-style-type: none"> a. Copy the files from the Serena Common Web services alm\WEB-INF\classes folder to a temporary folder. <p>For example:</p> <pre>..\Program Files\Serena\common\tomcat\6.0\ webapps\alm\WEB-INF\classes</pre> b. Backup your SBM database. <ol style="list-style-type: none"> 2. For an easier restoration if necessary, back up the product directories: <ol style="list-style-type: none"> a. Copy the Serena Common Web services webapps folder to a temporary folder. <p>For example:</p> <pre>..\Program Files\Serena\common\tomcat\6.0\ webapps</pre> b. Copy the Serena\Solutions folder to a temporary folder. <p>For example:</p> <pre>..\Program Files\Serena\Solutions</pre>

3: Copy the Upgrade Package	<p>Prepare for and extract the upgrade package as follows:</p> <ol style="list-style-type: none"> 1. Delete the Serena\Solutions folder. For example: C:\Program Files\Serena\Solutions 2. Download the upgrade package, for example ALM4.0.1_Win32_and_Win64_Upgrade.zip, and copy it to a temporary folder. 3. Extract the upgrade package to the Serena\Solutions folder. 4. The following files should now appear under the Solutions folder: <ul style="list-style-type: none"> • com.serena.alm.sbm.shell-4.0.1.zip • solution file: for example, ALM_Solution_Pack-4.0.1.0.186.sln • war files <ul style="list-style-type: none"> ◦ alm.war ◦ almzmf.war ◦ almzmfalf.war ◦ almzmfws.war <p>NOTE You can ignore any other files in the Solutions folder for now.</p>
4: Install the new Serena Common Tools	<p>Install the new Serena Common Tomcat common tools as follows:</p> <ol style="list-style-type: none"> 1. Download the full installation package, for example ALM4.0.1_b0186_win32.zip or ALM4.0.1_b0186_win64.zip, and extract the files. 2. Download the full installer and unpack it. 3. Backup your existing common tools. For example: c:\program files\serena\common\tomcat\6.0 Tip: View the properties of the Serena Common Tools service for the exact path. 4. Uninstall the existing Serena Common Tools. There is no upgrade for the common tools. 5. Install the new Serena Common Tools. 6. Verify that the new Serena Common Tomcat service starts in Windows Services.

<p>5: Configure Web Services</p>	<p>Configure the Web services files in the Serena Common Tomcat Web server.</p> <ol style="list-style-type: none"> 1. Stop the Serena Common Tomcat service in Windows. For example, select Start Administrative Tools Services, select the service, and click Stop. 2. Navigate to the Serena Common Tomcat webapps folder. For example: <code>..\Program Files\Serena\common\tomcat\6.0\webapps</code> 3. Delete the following folders under the webapps folder: <code>alm</code> <code>almzmf</code> <code>almzmfalf</code> <code>almzmfws</code> 4. Delete the following war files under the webapps folder, if present: <code>alm.war</code> <code>almzmf.war</code> <code>almzmfalf.war</code> <code>almzmfws.war</code> 5. From the ALM folder where you extracted them in a preceding step, copy all of the following war files to the Serena Common Tomcat webapps folder: <code>alm</code> <code>almzmf</code> <code>almzmfalf</code> <code>almzmfws</code> 6. Start the Serena Common Tomcat service. This war file contents are automatically extracted to new directories in that location. You should now see the following directories under webapps: <code>alm</code> <code>almzmf</code> <code>almzmfalf</code> <code>almzmfws</code>
<p>6: Copy the Solution File</p>	<p>Copy the solution file that contains the Serena ALM process apps and all related orchestrations, reports, and tables.</p> <ol style="list-style-type: none"> 1. From the Solutions folder, copy the solution pack .sln file, such as <code>alm.solution.pack.4.0.1.0.186.sln</code>, to the SBM WEB-INF\solutions folder. For example: <code>C:\Program Files\Serena\SBM\Common\jboss405\server\default\deploy\mashupmgr.war\WEB-INF\solutions</code>

7: Install the User Interface Files and Report Templates	<p>Install the user interface files and report templates.</p> <ol style="list-style-type: none"> 1. Extract the <code>com.serena.alm.sbm.shell-4.0.1.zip</code> file directly to the SBM Application Engine folder. For example: <code>C:\Program Files\Serena\SBM\Application Engine\</code> <p>The files in the zip file should extract to the appropriate directory structure. For any conflicts, select the option to replace with the newer versions.</p> <p>Verify the extraction by looking at the dates of the files in the <code>template\shell\alm</code> folder, for example <code>C:\Program Files\Serena\SBM\Application Engine\template\shell\alm</code>. The files should have the date close to that of the upgrade package you used.</p>
8: Import the Solution	<p>Import the Serena ALM solution.</p> <ol style="list-style-type: none"> 1. Log into the SBM Application Repository as an SBM administrative user. 2. Navigate to the Solutions tab and import the solution. For example: <code>ALM_Solution_Pack-4.0.1.0.186</code> <p>For detailed instructions see Importing the Serena ALM Solution [page 26].</p>
9: Promote the Snapshots	<p>Promote the snapshots.</p> <ol style="list-style-type: none"> 1. Navigate to the Process App Snapshots tab. 2. In SBM Application Repository, promote the snapshots. When promoting the snapshots, make sure to select the endpoints as needed, and make sure the endpoints are authenticated with Security Token. <p>IMPORTANT! Make sure to check for any warnings after you have promoted the snapshots. For information on analyzing warnings or errors, see Snapshot Promotion Errors [page 188].</p> <p>For detailed instructions on promoting the snapshots, see Which Process Apps Should I Promote and Deploy? [page 29] and Promoting the Release Manager Snapshots [page 31].</p>
10: Configure the Process Apps	<p>Configure the Release Manager Process Apps as required for each solution. After the process apps are configured, you must deploy them from within SBM Composer. This validates the target endpoints prior to deployment to ensure that your environment is correctly configured.</p> <p>For detailed instructions see Configuring the Release Manager Process Applications [page 37].</p>
11: Set Privileges and roles	<p>Set privileges for the administrative user to the Release Manager objects, such as projects, reports, and tables and ensure roles are enabled for all Serena ALM projects. See Configuring the Admin User for Release Manager [page 48].</p>
12: Verify Serena ALM is activated	<p>Verify that Serena ALM is activated by entering the URL in your Web browser. For example:</p> <p><code>http://<hostname>/tmtrack/tmtrack.dll?shell=alm</code></p> <p>where <code>hostname</code> is your Serena ALM host server name.</p>

13: Refresh the Web browser data	Clear and refresh your Web browser cache to ensure that out-of-date UI shell elements are no longer saved in your Web browser. See Clearing the Web Browser Cache [page 193]
14: Update the Registry	<p>Update the registry to ensure the proper version of Serena ALM is registered for future reference. (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. <p>View the registry key for the version of Release Manager that you are upgrading from:</p> <ul style="list-style-type: none"> • The registry key for Release Manager version 4.0 is: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{F4852B57-324F-406F-87FF-18C9Fa5B0DD9} • The registry key for Release Manager version 3.5 is: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{37AA8300-7D82-42E7-A139-F7B49D0CF3DC} • The registry key for Release Manager versions prior to 3.3 is: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{1F0B02EF-2F1D-48EF-B397-DF5488FC7D27} b. Modify DisplayVersion by changing the Value data entry to 4.0.1. <p>NOTE</p> <p>This registry location has all the information that you will see in Windows Add/Remove programs and Programs and Features.</p>
15: Configure Release Vault	Configure Release Vault as needed. For detailed instructions see Dimensions CM for Release Manager Configuration [page 78] and/or ChangeMan ZMF for Serena ALM Configuration [page 93] .
16: Configure Release Automation	Configure the Release Automation objects as needed. For detailed instructions see Release Automation Configuration [page 109] .
17: Configure Connections and Providers	<p>Configure the connections using the ALM Configurator and specify details in provider properties files as needed. For detailed instructions see Provider Configuration [page 121].</p> <p>CAUTION! If you choose to restore values from configuration files you backed up, you should copy only your implementation-specific values into specific parameters to ensure that you don't introduce errors. These files change significantly from release to release, so restoring by copying and replacing entire files is not recommended.</p>

Development Manager Installation and Upgrade Scenarios

There are a number of scenarios for how you might want to install and configure Development Manager 4.0.1 depending on whether you are upgrading and whether you are also using Requirements Manager.

See the following sections according to whether you are upgrading or performing a clean install:

- [Upgrading an existing installation of Development Manager to 4.0.1 \[page 147\]](#)
- [Performing a new installation of Development Manager 4.0.1 \[page 147\]](#)

See the following sections according to how you will be using Development Manager:

- [Installing Development Manager and Requirements Manager at the Same Time \[page 148\]](#)
- [Installing Development Manager Without Installing Requirements Manager \[page 148\]](#)
- [Installing Requirements Manager without installing Development Manager \[page 148\]](#)
- [Configuring Development Manager when you Have Previously Installed a Stand-alone Version of Requirements Manager. \[page 148\]](#)
- [Configuring Requirements Manager when you Have Previously Installed a Stand-alone Version of Development Manager 4.0.1. \[page 148\]](#)
- [Installing Development Manager Without Products and Components \[page 153\]](#)

These sections give a high-level view of which steps to carry out for each of these scenarios.

Upgrading an existing installation of Development Manager to 4.0.1

You need to carry out the configuration steps described in the sections below, taking note of the section [Which Process Apps Should I Promote and Deploy? \[page 29\]](#):

- [Importing and Promoting Serena ALM \[page 26\]](#)

You will need to be aware of the issues [Create Dev CR Button not Appearing After Upgrade \[page 203\]](#), and [Error Promoting Dev Change Requests Snapshot \(DEF217760\) \[page 204\]](#).

- [Configuring the Development Manager Process Applications \[page 39\]](#)

Performing a new installation of Development Manager 4.0.1

After carrying out the installation, carry out the steps described in [Installation and Configuration Checklist \[page 18\]](#), taking note of the section [Which Process Apps Should I Promote and Deploy? \[page 29\]](#). Then, depending on whether or not you are using Requirements Manager, see the corresponding section below:

- [Installing Development Manager and Requirements Manager at the Same Time \[page 148\]](#)

or

- [Installing Development Manager Without Installing Requirements Manager \[page 148\]](#)

Installing Development Manager and Requirements Manager at the Same Time

After performing the steps above for either [Performing a new installation of Development Manager 4.0.1 \[page 147\]](#) or [Upgrading an existing installation of Development Manager to 4.0.1 \[page 147\]](#), you will also need to consult the *Serena Requirements Manager Installation and Configuration Guide* for details on configuring Requirements Manager.

Installing Development Manager Without Installing Requirements Manager

After performing the steps above for either [Performing a new installation of Development Manager 4.0.1 \[page 147\]](#) or [Upgrading an existing installation of Development Manager to 4.0.1 \[page 147\]](#), you will then need to carry out the steps described in [Configuring a Stand-Alone Development Manager Installation \[page 149\]](#).

Installing Requirements Manager without installing Development Manager

For details, see the *Serena Requirements Manager Installation and Configuration Guide*.

Configuring Development Manager when you Have Previously Installed a Stand-alone Version of Requirements Manager.

You will need to promote and deploy the additional process apps for Development Manager:

- Dev Control Orchestrations
- Dev Tasks
- Dev Change Requests
- Dev Packages
- Quality Center Connector (if you are using Quality Center)
- Global Process App (eval)
- RLM_AUX

See [Importing and Promoting Serena ALM \[page 26\]](#) for details.

You will then need to perform any necessary configuration steps described in:

- [Creating and Assigning the Development Manager Users \[page 75\]](#)
- [Configuring the Development Manager Process Applications \[page 39\]](#)
- [Configuring ZMF Communication on the Mainframe \[page 93\]](#)

Then you will need to perform the steps described in [Adapting an Existing Stand-alone Requirements Manager Installation for use with Development Manager \[page 151\]](#).

Configuring Requirements Manager when you Have Previously Installed a Stand-alone Version of Development Manager 4.0.1.

For details of configuring Requirements Manager, see the *Serena Requirements Manager Installation and Configuration Guide*.

Configuring a Stand-Alone Development Manager Installation

If you are configuring an installation of Development Manager that does not include Requirements Manager, you need to make some changes to the process apps to exclude the Requirements Manager components from the SBM user workspace. These changes consist of:

- Hiding a number of Requirements Manager-related forms by enabling the rule Hide Requirements Manager data. These forms are:
 - Dev Change Requests:
 - Change Request Base State Form
 - Change Requests Submit Form
 - Change Requests Update Form
 - ALM Projects:
 - Project Submit Form
 - Project Update Form
- Removing the Requirements Manager-related columns from certain report definitions. These reports are:
 - Dev Change Requests:
 - CRs associated to this Project
 - CRs associated to this Dev Package
 - Dev CR.View.All
 - Dev CR.View.Closed
 - Dev CR.View.In Development
 - Dev CR.View.In Test
 - Dev CR.View.Mine
 - Dev CR.View.Waiting to Start
- Removing the Requirements Manager-related columns from the above reports in the SBM workspace.
- Removing the Viewers Role from the Requirements Manager Project.

Hiding the Requirements Manager-related Forms

1. Open SBM Composer.
2. Open a process app, for example Dev Change Requests.
3. Select Visual Design.
4. Under Forms, select Change Request Base State Form.
5. Select Actions in the Property Editor.
6. Click the Enabled check box for Hide RQM Data.

7. Repeat [Step 4 \[page 149\]](#) through [Step 6 \[page 149\]](#) for:
 - Change Requests Submit Form
 - Change Requests Update Form
8. Repeat [Step 2 \[page 149\]](#) through [Step 6 \[page 149\]](#) for:
 - ALM Projects:
 - Project Submit Form
 - Project Update Form

Deploy the Process Apps after completing these steps.

Removing Requirements Manager-related Columns from Report Definitions

1. Open SBM Composer.
2. In SBM Composer, open the Dev Change Requests process app.
3. Select **Report Definitions**.
4. *Select CRs associated to this Project.*
5. Right click the column *Associated Requirements*, and select **Delete**.
6. Repeat the above steps for:
 - *CRs Associated to this Dev Package.*
 - *View.All*
 - *View.Closed*
 - *View.In Development*
 - *View.In Test*
 - *View.Mine*
 - *View.Waiting to Start*

Deploy the Process Apps after completing these steps.

Removing Requirements Manager-related Columns from Reports

1. Open the SBM User Workspace by entering:
`http://<hostname>/tmtrack/tmtrack.dll?`
In a Web browser
2. Log in as Admin.
3. Select the Dev Change Requests Project.
4. Select **Reports**.
5. Click **Find Reports**.
6. Click the **Search** button.

7. Click the Edit button for *CR(s) associated to this Dev Package*.
8. Remove the column *Associated Requirements*.
9. Repeat these steps for the reports:
 - *CR(s) associated to this Project*.
 - *Dev CR.View.All*
 - *Dev CR.View.Closed*
 - *Dev CR.View.In Development*
 - *Dev CR.View.In Test*
 - *Dev CR.View.Mine*
 - *Dev CR.View.Waiting to Start*

Removing the Viewers Role from the Requirements Manager Project

1. Log into SBM Administrator by entering the following URL in a browser
`http://<hostname>/tmtrack/tmtrack.dll?shell=alm`
2. Click the Administrator button in the toolbar
3. Select **Groups**.
4. Select *Everyone* and click **Details**.
5. Select Roles.
6. Under Projects, select RQM.
7. Select the *Viewers* role and uncheck **Enabled**.
8. Click **Save**.

Adapting an Existing Stand-alone Requirements Manager Installation for use with Development Manager

If you have configured an installation of Requirements Manager that did not include Development Manager, you will have previously carried out the steps described in the *Serena Requirements Manager Installation and Configuration Guide* to hide certain Development Manager features from the SBM user workspace. You will therefore need to reverse those changes to include the Requirements Manager components in an integration between Development Manager and Requirements Manager. You will also need to update the endpoint for the `rtmService` in the Dev Control Orchestrations process app.

You will need to make the following changes to the process apps:

- Dev Control Orchestrations process app:
- Update the endpoint for the `rtmService` for the Dimensions RM server details.
- ALM Projects process app:

Disable the rule *Hide DVM data* for the following forms:

- Project Base State Form With Child Transitions

- Project Base State Form Without Child Transitions
- Project Submit Form
- Project Update Form
- Requirement Approval Management process app:
- Approval Process: Enable the *Distribute* transition.

Setting the Correct Endpoint for rtmService

After you have completed Installing and configuring Requirements Manager you need to edit the endpoint for rtmService in the RM to RQM Connector process app and redeploy it.

1. Open SBM Application Repository
2. Select **Process Apps** on the left.
3. Select the RM to RQM Connector process app and click the **Deploy** button
4. Click in the Destination field.
5. Select the environment and click Next.
6. On the Mappings page, click the **Choose Destination Endpoint** button to choose the correct endpoint (not rtmService_2).

Unhiding the Development Manager-related Forms

1. Open SBM Composer.
2. Open the ALM Projects process app.
3. Select Visual Design.
4. Under Forms, select *Project Base State Form With Child Transitions*.
5. Select **Actions** in the Property Editor.
6. Uncheck the Enabled check box for *Hide DVM Data*.
7. Repeat this for:
 - Project Base State Form Without Child Transitions
 - Project Submit Form
 - Project Update Form
8. Deploy the process app after completing these steps.

Enabling the Distribute transition for the Approval Process Application Workflow

1. Open SBM Composer.
2. Open the *Requirement Approval Management* process app.
3. Select the *Approval Process* application workflow.
4. Select the *Distribute* transition.
5. Set **Status** to Enabled.

-
6. Deploy the process app after completing these steps.

Adapting Development Manager to Hide Products and Components

If you are not using products and components, you may wish to carry out the steps described in [Hiding the Product and Component Related Fields \[page 153\]](#) and [Removing the Viewer's Role from the Component Lifecycle Project \[page 153\]](#) if you do not want them to appear in the Development Manager process apps.

Hiding Form Fields for Products and Components

If you do are not using products and components for your development work you may wish to hide the forms related to those fields in the process apps so that they do not appear in the Development Manager process apps. To do this, carry out the following steps:

1. Open SBM Composer.
2. Open a process app, for example ALM Projects.
3. Select Visual Design.
4. Under Forms, select a form that has the action *Hide OPD Data*.
5. Select Actions in the Property Editor.
6. Click the **Enabled** check box for Hide OPD Data.
7. Repeat [Step 4 \[page 153\]](#) through [Step >6 \[page 153\]](#) on each form where the *Hide OPD Data* action is available to be enabled in the process apps:
 - ALM Projects
 - Dev Change Requests
 - Dev Tasks
8. Redeploy the Process Apps after completing these steps.

Removing the Viewers Role from the Component Lifecycle Project

1. Log into SBM Administrator by entering the following URL in a browser
`http://<hostname>/tmtrack/tmtrack.dll?shell=alm`
2. Click the Administrator button in the toolbar
3. Select **Groups**.
4. Select *Everyone* and click **Details**.
5. Select Roles.
6. Under Projects, select Component Lifecycle Project.
7. Select the *Viewers* role and uncheck **Enabled**.
8. Click **Save**.

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 Manager, 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 Release Manager 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 section includes the following topics.

[Customizing Release Manager Workflows \[page 154\]](#)

[Modifying Release Types and Stages \[page 155\]](#)

[Adding Provider Connections \[page 171\]](#)

[Configuring Release Manager to Use a Different Port \[page 176\]](#)

[Activating Environment Association to Release Packages \[page 180\]](#)

[Customizing the SSM Integration \[page 181\]](#)

[Setting Maximum Associations for Release Manager Objects \[page 184\]](#)

[Customizing the User Interface \[page 185\]](#)

Customizing Release Manager 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 Manager processes.

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

What Can You Change?

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

IMPORTANT! The Release Package workflow contains logic that is integrated with the Release Type and Stage implementation. Information on changing the Release Package workflow is included in context in [Modifying Release Types and Stages \[page 155\]](#).

What is the Impact?

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

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

How Do You Change It?

- Change the workflow and related objects in SBM Composer according to the SBM documentation.
- Change the auxiliary table entries in SBM System Administrator according to the SBM documentation.

Documentation References

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

Modifying Release Types and Stages

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

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

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

What Can You Change?

You can change the following release type and stage information:

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

- Delete release types
- Delete stages

What is the Impact?

- Release types and stages are interrelated, and if you change one aspect, you must change all related information in both release train and release package primary and auxiliary tables, forms, workflows, and UI JavaScript.
- You must be careful when changing the Release Package system field definitions because they are used by Serena Release Manager to deploy release packages. These include the following:
 - **Package type:** Controls the release package staging process sequence.
 - **Deploy state:** Controls the re-deployment process.
 - **Next Deploy Transition:** The update transition name to use when a release package is successfully deployed. The default release package workflow value is set to **Deployed**.
 - **Failed Deploy Transition:** The update transition name to use when the release package deployment process fails. The default release package workflow value is set to **Fail Deployment**.
- You must be careful when changing the Release Package process app not to adversely affect the implementation of the Deployment Task execution. Deployment Task field dependencies are as follows:
 - Deployment Task Status Single Selection control field:

The task status single selection field, TASK_STATUS, controls when a deployment task can be executed for deployment, when to execute the task deployment process, and when to execute the fail deployment process.

- Default settings:

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

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

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

How Do You Change It?

You add or change most stage information in SBM Composer.

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

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

Summary of Adding a Stage

Adding a Stage for the Release Train Process App

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

2. Add the field controls in the related forms:
 - createReleaseTrain
 - viewReleaseTrain
3. Edit **Attributes visibility** in the Visual Design JavaScripts to specify when to show or hide the new date fields.
4. Include the new Start and End Date fields in the report All Release Train.

Adding the Stage for the Release Package Process App

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

Example of Adding a Stage

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

Adding a Stage for the Release Train Process App

To add a stage for the Release Train process app:

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

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

Release Train (Primary)

Release Train (Primary Table)
[No description]

Field name	Type	Database field name	Section	Dependen
Type : Binary/Triinary : 1 item				
Active/Inactive	Binary/Triinary	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

Property Editor

PAT end date Date/Time Field

General

Name: PAT end date

Database field name: PAT_ENDDATE Type: Date/Time

Description:

Privilege section: Standard

Production Acceptance Testing

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

createReleaseTrain

Official release name: Code name: Release type: Release manager:

Official release n... Code name Release type Release mana...

Description:

Description

Release train schedule

Integration test start date: Integration test end date:

Integration test start date Integration test end date

UAT start date: UAT end date:

UAT start date UAT end date

PAT start date: PAT end date:

PAT start date PAT end date

Production deployment start date: Production deployment end date:

Production deployment... Production deployment...

Add the new PAT Start and End Control

Property Editor

createReleaseTrain Transition Form

General Name: createReleaseTrain

JavaScripts Description:

Options: ☒ Remove transition buttons matching custom transition controls

☒ Validate required fields before form submit

Size guide: ☒ Show Width: 800 Height: 600

Property Editor Message List

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

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

```

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

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

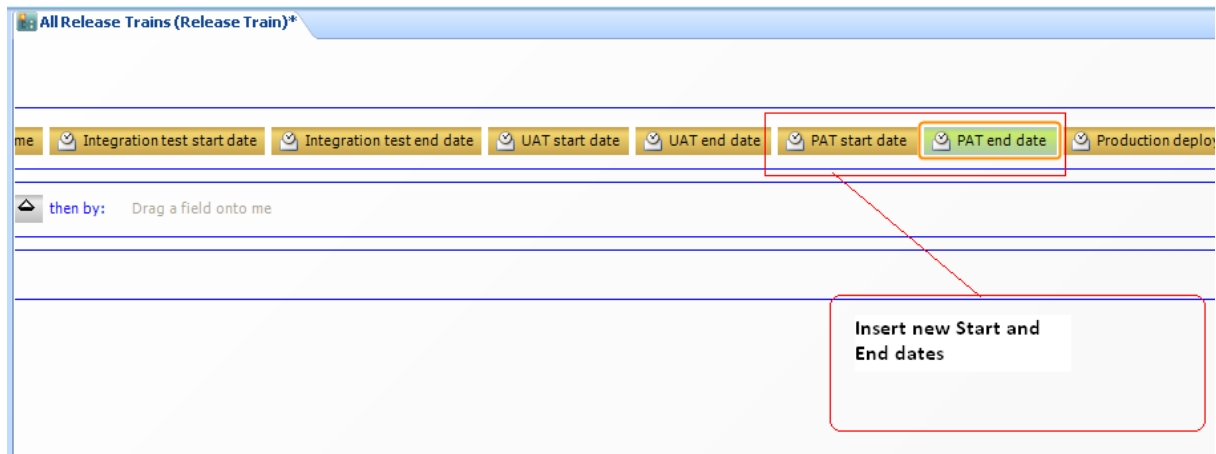
    whatToHide(type, fields);
    hideNeeded(fields);
}
);
AddChangeCallback("ISSUETYPE",
function() {
    var type = GetFieldValue("ISSUETYPE");
    var fields = {};

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

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

```

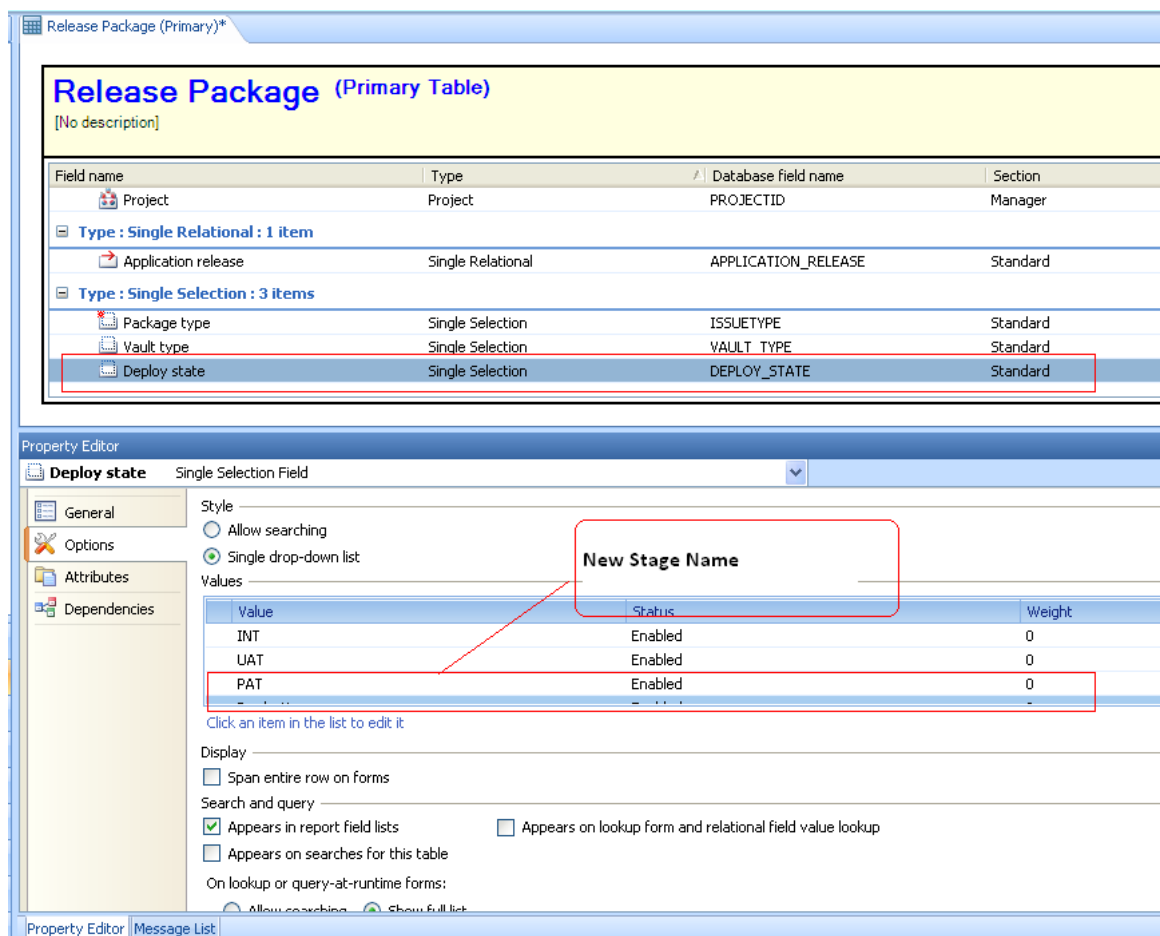
4. Include the new Start and End Date fields in the report **All Release Trains** as shown in the following figure.



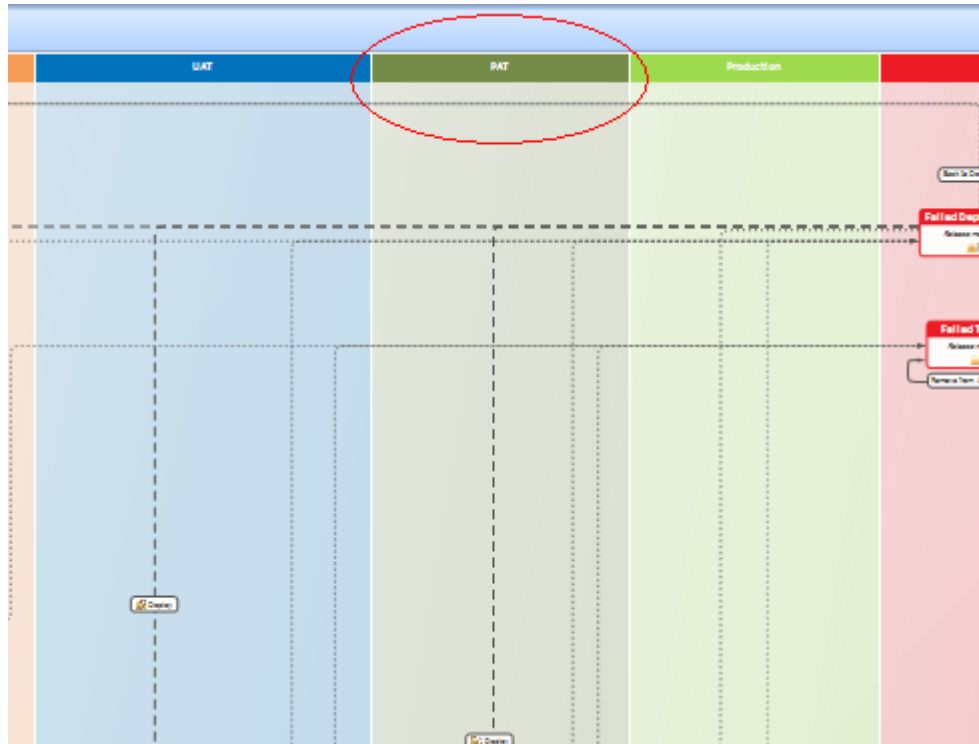
Adding the Stage for the Release Package Process App

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



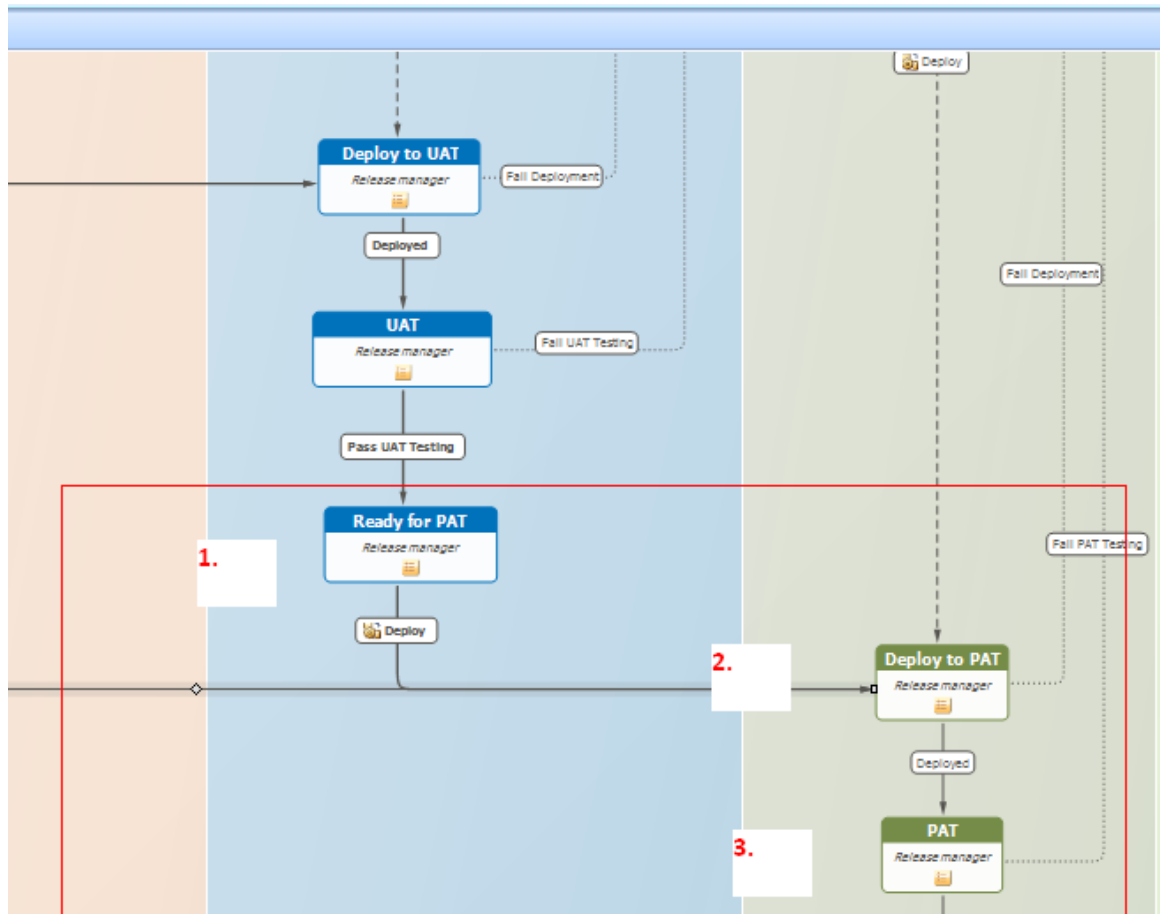
2. Create a new swim lane for the new stage process and arrange it according to the Release Package staging process sequence as shown in the following figure.



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
		Restrict by Role	default values
	To transition: Fail Deployment	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

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

- 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

6. Add a new re-deploy transition action.

Set the rule as follows:

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

The rule settings are shown in the following figure.

Action Wizard

Which transition do you want to invoke?

Step 1: Select transition

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

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

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

Cancel < Back Next > Finish

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

Release Train (Primary)

Release Train (Primary Table)
[No description]

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

On-Premise
Standard Fields

Stage Name: /LAT

Code: e27c19

Start Date DB Prefix: PAT_STARTDATE

End Date DB Prefix: PAT_ENDDATE

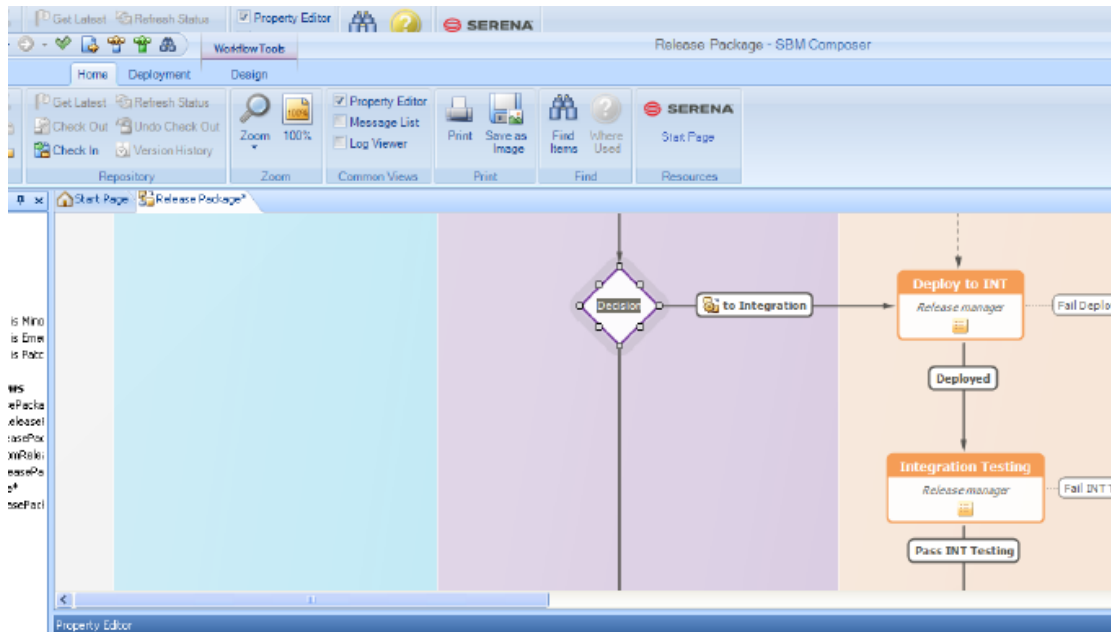
Package deploy state: Deploy to PAT

Connected to: on-Q4-TEST4 (On-Premise)

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

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

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



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

Start Page Release Package (Primary)

Release Package (Primary Table)

[No description]

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

Property Editor

Package type Single Selection Field

General

Options

Attributes

Dependencies

Allow searching

Single drop-down list

Values

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

Click an item in the list to edit it

Display

Span entire row on forms

Default weight

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

The screenshot shows the 'Release Package type is Patch' rule configuration in the Release Manager. The top bar includes a 'Start Page' button and the title 'Release Package type is Patch'. The main area displays the rule expression 'Package type in Patch' with a prompt to 'Drag an operator or field onto the expression above'. Below this is a 'Rule summary' box showing 'Package type in Patch'. At the bottom, the 'Property Editor' is open, showing the rule's name and description fields.

Start Page Release Package type is Patch

Rule: Release Package type is Patch

Package type in Patch

Drag an operator or field onto the expression above

Rule summary

Package type in Patch

Property Editor

Release Package type is Patch Rule

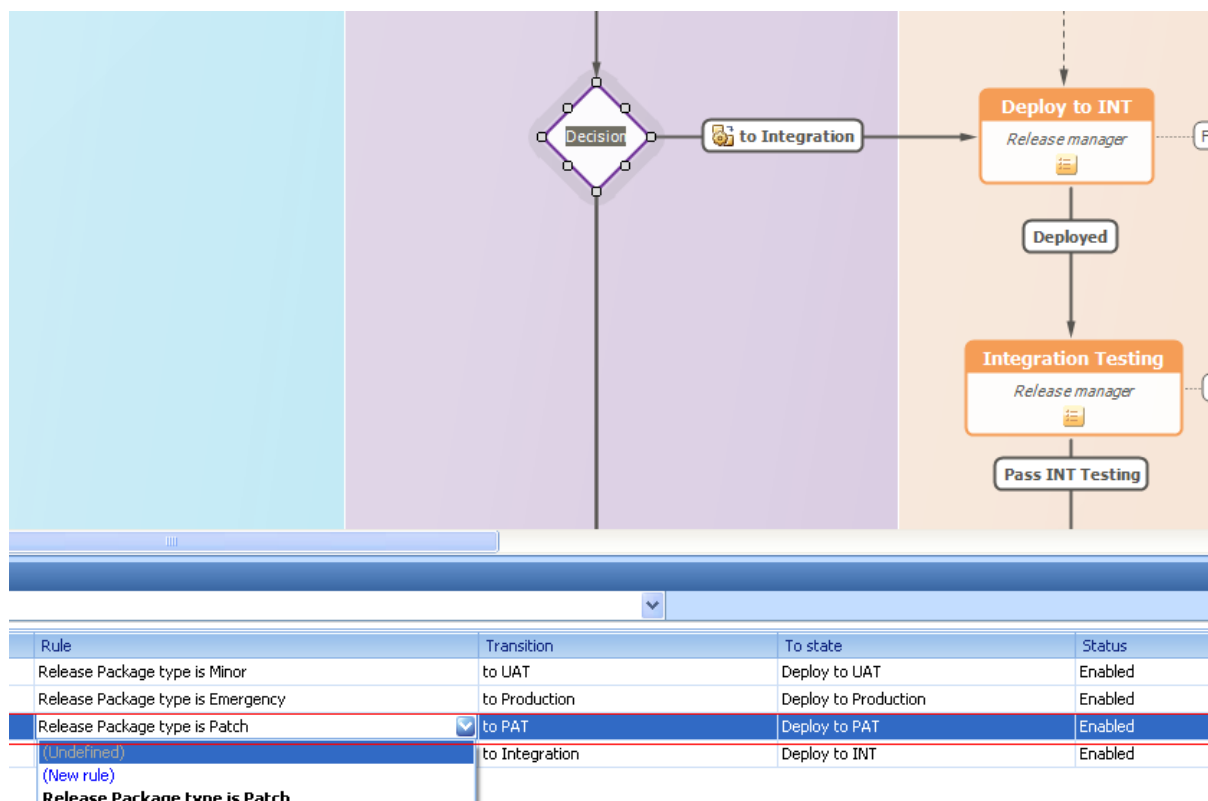
General Name: Release Package type is Patch

Description:

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

To and From	Transition	Option	Selection
From decision box: Deploy To state: Deploy to PAT	to PAT	Options	Quick transition and Hide button on form
		Form	None
		Field Privileges	default values
		Field Overrides	Failed Deploy Transition <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 want is already implemented for Serena Release Manager, please check the most current Serena Release Manager documentation and the online knowledgebase on the Serena Customer Support website.

To implement the providers, see the following sections:

- [Creating a Class for Your Provider \[page 172\]](#)
- [Creating Properties Files for Your Providers \[page 172\]](#)
- [Building and Packaging \[page 175\]](#)
- [Telling Serena Release Manager to Use This Provider \[page 175\]](#)

NOTE For documentation on configuring existing provider connections, see [Provider Configuration \[page 121\]](#).

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.alm.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.alm.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 \[page 172\]](#).
- Define all instance-specific values for parameters in a properties file. See [Telling Serena Release Manager to Use This Provider \[page 173\]](#).

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.alm.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.alm.provider.fs.FSRequestsProvider">
<property name="providerName" value ="${requests.provider.name}"/>
<property name="providerDescription" value ="${requests.provider.description}"/>
<property name="requestsFile" value ="${provider.fs.requests.file}"/>
</bean>
<bean id="deployUnitsProvider"
    class="com.serena.alm.provider.fs.FSDeployUnitsProvider">
<property name="providerName" value ="${vault.provider.name}"/>
<property name="providerDescription" value ="${vault.provider.description}"/>
<property name="depunitsFile" value ="${provider.fs.depunits.file}"/>
<property name="stagesFile" value ="${provider.fs.stages.file}"/>
<property name="depareaFile" value ="${provider.fs.deparea.file}"/>
</bean>
</beans>
```

Telling Serena Release Manager to Use This Provider

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

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

Examples

fs_example.properties

```
# requests provider definitions
requests.provider.name = filesystem
requests.provider.description = Simple file-system Request Provider
# vault provider definitions
vault.provider.name = filesystem
vault.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_dir>|<area_stage_id>|<area_status>|<depunit_proj_name>
AR0001|Dev area|c:\work\|SIT|Open|QLARIUS:RLM_TEST
AR0002|Dev area|c:\work2\|SIT|Open|QLARIUS:RLM_TEST2
```

Building and Packaging

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

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

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

```
com.serena.alm.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:

```
..\Program Files\Serena\common\Tomcat\6.0\webapps\alm
```

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 alm folder. For example:

```
..\Program Files\Serena\common\Tomcat\6.0\webapps\alm
```

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:

```
..\Program Files\Serena\common\tomcat\6.0
```

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

```
..\Program Files\Serena\common\tomcat\6.0\webapps\alm\WEB-INF\classes
```

6. Open providers.properties in your text editor.

7. Add providers:

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

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

- b. Add your new deployment units provider to `vault.providers.keys`. For example:

```
vault.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 `vault.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
# vault provider keys
vault.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 Manager installer automatically installs and configures the Serena Common Web server to run on the default port of 8080. If this port is already in use by another application on your server, or if you already have an instance of the Serena Common Web server running on a different port on this server, you will need to decide how you want to proceed.

- Serena Release Manager is configured to use port 8080 by default, but changing the Web server to use port 8080 may impact other Serena products if they rely on a previously configured port. Make sure all Serena products using the same Serena Common Web server are configured to use the same port number, or install the products on separate servers so that they can use different Serena Common Web servers with different port numbers.
- Use the default installation and configuration procedures on the Serena Release Manager server in the following scenarios:
 - There is a non-Serena Tomcat, IIS, or other Web server installed on this server on a port other than 8080, and port 8080 is free.
 - There is a Serena Common Tomcat Web server already installed on this server on port 8080.
 - There is not a Web server on this server.
- If you want to use a different port for Serena Release Manager, you must change the port number as shown in [Checklist for Changing the Port Number \[page 177\]](#).

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

Checklist for Changing the Port Number

Procedure	Reference
Install and configure the Serena Common Tomcat Web server and Serena Release Manager as usual. If you have already done this, you do not need to do this again.	See Installing Serena Release Manager and Release Automation Configuration [page 109].
Rerun the Serena Release Manager installer at the command line with parameters to change the port the Serena Common Tomcat Web server runs on.	See Changing the Port on Which the Common Web Server Runs [page 177].
Configure the endpoints for the snapshots to point to the non-default port, and then promote the snapshots and deploy the process apps.	See Promoting the Release Manager Snapshots [page 31].
Configure all RESTgrid widgets in the process app forms to point to the non-default port and then redeploy the process apps.	See Configuring a Non-Default Web Server Port in the Process Apps [page 178].
Change the port to which the Web Services WSDLs point and re-import the WSDLs.	See Changing the Web Services to Point to a Different Port [page 180].
When you are finished, publish and redeploy the process apps. As you deploy, verify that the endpoints of the process application destinations are pointing to the port number you specified during the installation rather than the default port number of 8080. If they are not, update them as needed before deploying.	See Publishing and Deploying the Process Apps [page 38].

Changing the Port on Which the Common Web Server Runs

The default port on which this is configured to run is 8080. If you want to use a different port, you must first install the Common Web server and Serena Release Manager, 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 Manager installation file executable, `ALM.exe`. For example:

```
cd Users\<username>\Downloads\ALMBuild\Win64
```

3. Depending on your operating system, enter the following command at the prompt:

```
"ALM 32-bit.exe" /V"/L*v "%TEMP%\<logfilename>.log" TC_PORT=<port#>"
```

or

```
"ALM 64-bit.exe" /V"/L*v "%TEMP%\<logfilename>.log" TC_PORT=<port#>"
```

For example, here is the command for a 64-bit installation to log the install and change the Tomcat Web server port to 1234:

```
"ALM 64-bit.exe" /V"/L*v "%TEMP%\test_install.log" TC_PORT=1234"
```

The Serena Release Manager installer appears.

4. Deselect all features except for **Provider** and click **Next**.

The installer reconfigures the Serena Common Tomcat port to which Serena ALM is connected.

5. Continue with the next checklist item in [Checklist for Changing the Port Number \[page 177\]](#).

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 8080, you must change the port numbers in the Serena Release Manager process apps that have RESTgrid widgets. After you change the process apps, you need to redeploy them.

Changing the Port Number in the Forms

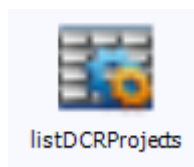
You must find the RESTgrid widgets in the forms, and update each occurrence of the default port number, 8080, to the port number for your Web server installation.

To change and redeploy the process apps:

1. Open the process app you want to change in SBM Composer. For example, open Release Package from the Application Repository.
2. Display the **Visual Design** view.
3. Under **Forms**, select one of the forms. For example, select createRelPackage. For a list of forms that you should update, see [Forms with RESTgrid Widgets \[page 179\]](#).

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 8080 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/alm/services/ReleaseRequestService/getRequestProjects
```

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

- d. Click **Update outputs**.

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

- e. Click **OK**.

7. Save and check in your changes.
8. Continue for each RESTgrid widget control in each process app. See [Forms with RESTgrid Widgets \[page 179\]](#) for the list of default controls to change.
9. Continue with the next checklist item in [Checklist for Changing the Port Number \[page 177\]](#).

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	listDimCMPProjects
	addZMF_Projects	listZMFProjects
	createRelPackage, createRPfromAR	listDCRProjects, listDimCMPProjects, listZMFProjects
Application Release	addBusinessChangeRequest	RESTGridWidget
Release Train	addRFC	listRfcRequests

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

Changing the Web Services to Point to a Different Port

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

To change and re-import the Web services WSDLs:

1. From SBM Composer, open one of the Serena Release Manager process apps.
2. In the navigation pane, click **Extensions**.
3. Under **Web Services**, select one of the following Serena Release Manager Web services, RLMUtilServices or ReleaseRequestService.
4. In the **Property Editor** beside the **WSDL** field, click **Reimport**.
5. Change the port number to the one you are using.
6. Click **OK** to re-import the updated Web service.
7. Repeat for each of the Serena Release Manager Web services in each of the process apps except RLM_AUX and Environments, which do not use endpoints.
8. Continue with the next checklist item in [Checklist for Changing the Port Number \[page 177\]](#).

Activating Environment Association to Release Packages

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

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

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

1. In SBM Composer, open the Release Package blueprint.
2. On the **Environments** tab form, add the button controls for **Add Environments** and **Remove Environments**.

3. In Serena Release Manager in the native SBM interface, create a report with the reference name that is referenced in the orchestration workflow.
4. Deploy the updated process app and test your changes.

Customizing the SSM Integration

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

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

What Can You Change?

You can change the SSM integration as follows:

- Create the reports that SSM and Serena Release Manager use for the integration.
- Change the events within the release train workflow.
- Change the integration in SSM.
- Change the instance of SSM to which Serena Release Manager connects.
- Configure the Release Train non-Gantt calendar to display the SSM Requests along with the release train Production dates.

What is the Impact?

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

How Do You Change It?

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

Documentation References

- Documentation on using the default SSM-Serena Release Manager integration from SSM is in *Serena Service Manager Concepts Guide* in "Serena Release Manager".
- Documentation on using the default SSM-Serena Release Manager integration from Serena Release Manager is in *Serena Release Manager Getting Started Guide* in "Associating RFCs with Release Trains".

- Documentation on configuring workflow events in SBM is in the *Serena Business Manager SBM Composer Guide* in "Applications".
- Documentation on managing reports in SBM is in the *Serena Business Manager User's Guide* in "Working with Reports".

Creating the Reports Used for the SSM Integration

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

To create the Release Manager reports used by SSM:

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

Example

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

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

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

Title:	Suitable Release Trains
Reference Name:	Suitable_Trains
Privilege Category:	Guest
Report Project:	Base Project
Columns to Display:	Official release name
	Production deployment start date
	Production deployment end date

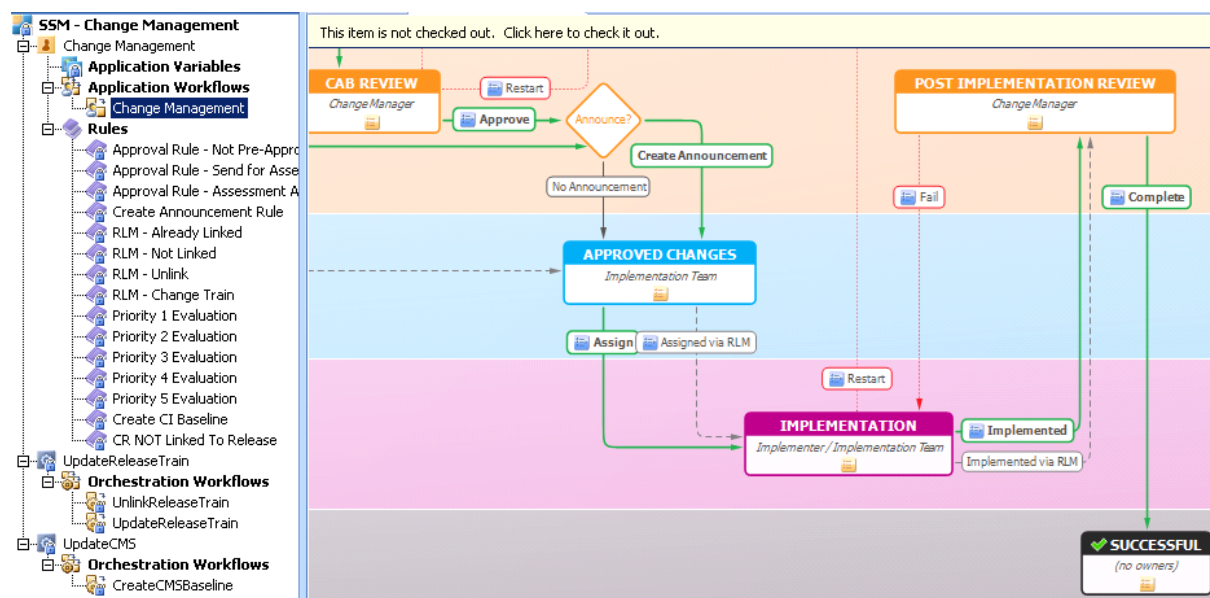
Include items from Sub-projects	(Select)
Use Basic Conditions:	State in Planning
	Production deployment start date = (Query At Runtime)
	Production deployment end date = (Query At Runtime)

Changing the Integration in SSM

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

Integration Points in the SSM Workflow

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



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

Configuring the SSM Feed for the Release Train Calendar

You can configure the non-Gantt Release Train calendar to display the SSM request dates along with the associated release train Production dates.

You must create the associated report in SSM provide the calendar feed. The report is referenced in the Serena Release Manager calendar user interface file, `../template/shell/alm/config/calendar.jsvar`.

To create the SSM report for the calendar:

1. In SSM, create a report as follows:

Title:	SSM.Approved.Changes
Reference Name:	SSM.Approved.Changes
Privilege Category:	Guest
Report Project:	Changes
Columns to Display:	Change ID
	Title
	State
	Implementation Start Date
	Implementation End Date
Search Filter	
Include items from Sub-projects	(Select)
Use Basic Conditions:	State in APPROVED CHANGES
	and
	Linked to Release = Yes
Additional Options:	
Optional HTML Template:	jsonlist.htm

Changing the Instance of SSM that Release Manager Uses

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

Setting Maximum Associations for Release Manager Objects

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

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

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

To change the maximum values for the associations:

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


```
..\Program Files\Serena\common\tomcat\6.0\webapps\alm\WEB-INF\classes
```

2. Open the `rlm-sbm.properties` file.
3. Update the value of one or more of the following entries as needed:

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

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

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

Customizing the User Interface

The Serena Release Manager user interface is populated using underlying SBM foundational objects and a UI shell layer written in HTML and JavaScript. You can usually change the information that appears on lists quite quickly by changing the underlying reports.

To change the UI shell itself requires a more in-depth knowledge of the product architecture. If you want to change elements there, please contact Serena Services for advice.

If you want to send input feeds of set a date-based items from an integrating product to the Release Train calendar, see the implementation of the SSM request calendar feeds and use that as an example.

Troubleshooting Release Manager

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

[Troubleshooting Overview \[page 186\]](#)

[Information from the User Interface \[page 186\]](#)

[Information from the ALM Configurator \[page 187\]](#)

[Information from Log Files \[page 187\]](#)

[Symptoms and Solutions \[page 188\]](#)

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 user interface.
- Information displayed in the Serena ALM Configurator.
- Information stored in installer log files.
- Information stored in the suite execution log file, `alm.log`.
- Troubleshooting information for integrating products, such as SBM, Dimensions CM, ChangeMan ZMF, and Serena Release Automation.

Information from the User Interface

Serena Release Manager 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 Manager 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 \[page 188\]](#) for more assistance in these situations.

Activity Log

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

Activity Page

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

History

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

Information from the ALM Configurator

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

From the ALM Configurator, click the following links in the navigation pane to display the corresponding pages.

- **CONFIG:** Verify and change connection information as needed.
- **SHOW LOG:** Show and analyze the product log file.
- **SERVICES:** List the Web services and verify that they are running.

Information from Log Files

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

Execution Log File

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

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

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

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

```
..\Program Files\Serena\common\tomcat\6.0
```

To change the location of the log file:

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

```
..\Program Files\Serena\common\tomcat\6.0\webapps\alm\WEB-INF\classes
```

2. Open the `log4j.properties` file.
3. Change the following line to specify the location for the file.

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

Installer Log Files

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

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

```
>cd %TEMP%
```

The installer log files are as follows:

```
Install_SUITE_comp32.log or Install_SUITE_comp64.log
```

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

summary.rtf

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

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

Symptoms and Solutions

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

- [Installer Errors \[page 188\]](#)
- [Snapshot Promotion Errors \[page 188\]](#)
- [Cannot log into Serena Release Manager \[page 190\]](#)
- [Cannot Log into Serena ALM Configurator \[page 190\]](#)
- [User Interface Display Issues \[page 192\]](#)
- [Create Release Fails with a Check Uniqueness Error \[page 194\]](#)
- [Unable to update landing page layout \[page 194\]](#)
- [Matches Not Found for Selections \[page 194\]](#)
- [No Change Package Data Displayed in ZMF Deployment Tasks \[page 195\]](#)
- [Release Package Deployment Fails \[page 195\]](#)
- [Slow Response Time \[page 196\]](#)

Installer Errors

If the installer fails, here are some possible solutions.

Common Tools files are missing from the install

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

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

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

If you do not have Serena Common Web services installed, you must make sure the Common folder for the Serena Common Web services installer is in the same folder as the ALM.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 SBM environment endpoints are not mapped properly

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

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

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

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

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

Some common oversights are:

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

The promotion succeeds but gives warnings

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

Warnings you can ignore

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

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

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

```
WARN -- Import: Invalid Data '19c59fdf-f13d-4b1a-a07c-0ffc1c14a37' 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.

Cannot log into Serena Release Manager

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

Verifying the Web Services Connection

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

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

`http://localhost: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. After you have verified that the Serena Common Tools Web server is running, browse to the `alm` subdirectory of your Tomcat installation. For example:

`http://localhost:8080/alm`

3. Log into the Serena ALM Configurator.
4. In the navigation pane, click **List Services**.

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

- **RLMUtilService**
- **ReleaseRequestService**
- **DeploymentAutomationService**
- **DeployUnitService**

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

Cannot Log into Serena ALM Configurator

If you cannot log into the Serena ALM Configurator, check the following.

Error Obtaining Security Token (DEF224185)

If when attempting to log on to the Serena ALM Configurator, you receive the error:

ALF SSO Gatekeeper error has occurred: Error obtaining security token.

Detail

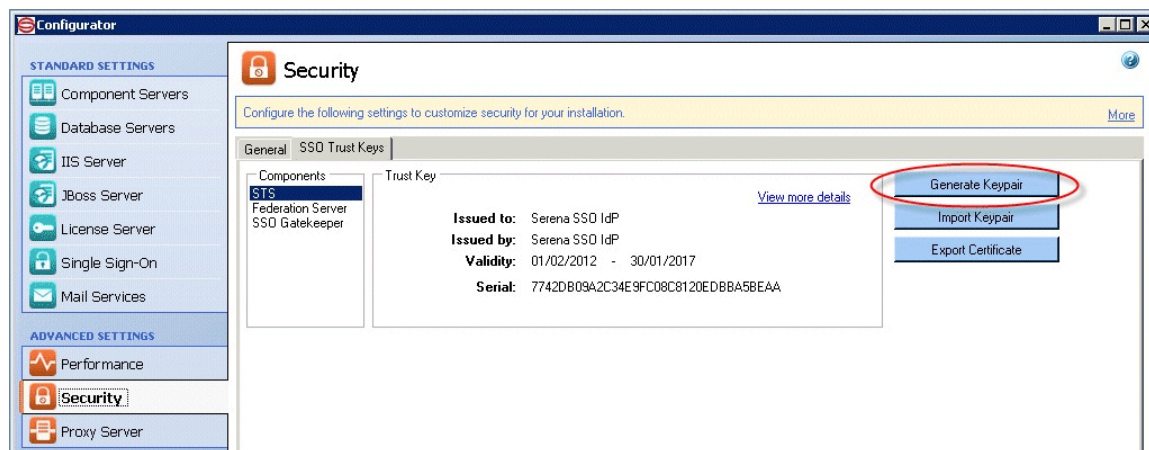
Validation of WS-Federation token failed with code 40:Token issuer not allowed.

You will need to update your SSO STS certificates by carrying out the following steps:

On the server where SBM is installed:

If you are using SBM 10.1 to 10.1.1.3:

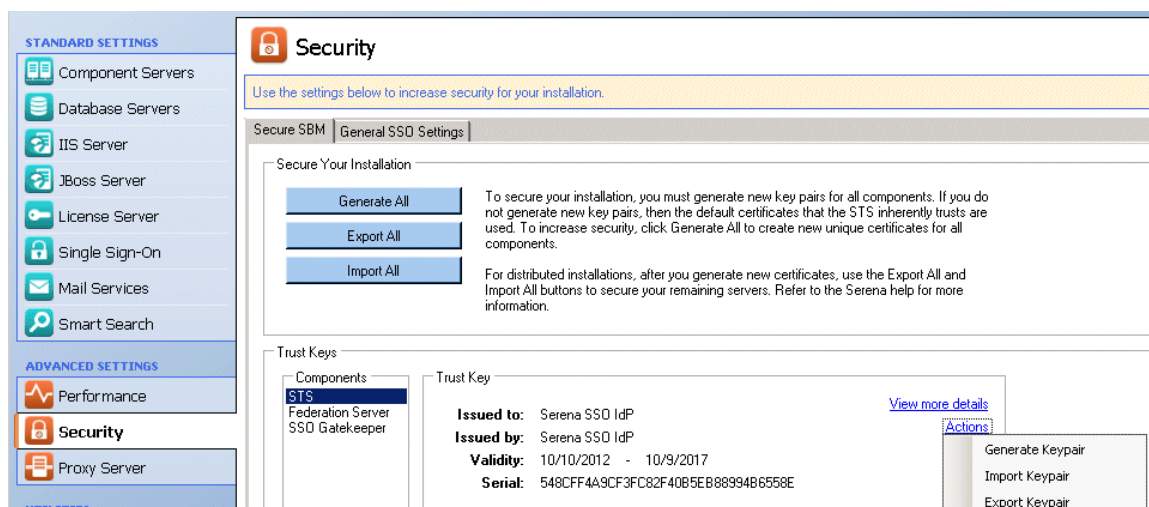
1. Start SBM Configurator and navigate to Security | SSO Trust Keys | STS.
2. Use the **Generate Keypair** button to regenerate the STS certificate.



3. Use the **Export Certificate** button to export the newly generated certificate to a .pem file (for example to C:\Temp\sts.pem).
4. Click **Apply** to apply the changes.

If you are using SBM 10.1.1.4:

1. Start SBM Configurator and navigate to Security | Secure SBM | Trust Keys | STS.
2. Under **Actions**, select **Generate Keypair** to regenerate the STS certificate.



3. Under **Actions**, select **Export Certificate** to export the newly generated certificate. Save this as a .pem file (for example to C:\Temp\sts.pem).
4. Click **Apply** to apply the changes.

On each machine where the Serena Common Tomcat is installed:

1. Stop Tomcat.
2. Copy the sts.pem file to that machine (for example to C:\Temp\sts.pem).
3. Under the folder where the common Tomcat is installed, for example:
 - ..\Program Files\Serena\Common Tools\tomcat\6.0\...
 - ..\Program Files\Common\tomcat\6.0\...
 - ..\Program Files\Serena\Dimensions 12.2\Common Tools\tomcat\6.0
4. Navigate to

`\alfssogatekeeper\conf`

and make a backup of the `truststore.jks` file.

5. Under the Common Tools or Common folder, Navigate to

`\jre\6.0\bin.`

6. Delete the existing STS certificate, for example using the following command:

```
keytool -delete -keystore "C:\Program Files\Serena\Common
Tools\tomcat\6.0\alfssogatekeeper\conf\truststore.jks
-alias sts
```

Enter the default password *changeit* when prompted.

7. Import the certificate from the `C:\Temp\sts.pem` file into `truststore.jks`, for example by using the following command:

```
keytool -import -keystore "C:\Program Files\Serena\Common
Tools\tomcat\6.0\alfssogatekeeper\conf\truststore.jks
-file "C:\temp\sts.pem" -alias sts
```

After entering the password, when prompted with *Trust this certificate? [no]:*, reply **Y**.

8. Restart Tomcat.

If this is the Dimensions CM Server:

Also do the following:

1. Stop the Dimensions Listener service.

2. Create a backup of

`C:\Program Files\Serena\Dimensions 12.2\CM\dfs\sts.pem`

3. Copy the file `C:\Temp\sts.pem` to

`C:\Program Files\Serena\Dimensions 12.2\CM\dfs\sts.pem`

4. Restart the Dimensions Listener service.

User Interface Display Issues

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

You can't view something you created or added

If you add something and it doesn't appear in the place it should in the UI, or information you expect to see in a view does not appear, you may not have proper privileges, roles, or ownership set for Serena Release Manager in SBM.

Examples:

- If you added release trains and they do not appear in the inbox in the **Manage All Items** section, your privileges probably aren't set properly.
- If you created a deployment process template and do not see the option to add deployment tasks, the owning role may not be enabled for the RLM Aux project.
- If you created an item as one user and log in as another, you may not see the item you created as the other user, depending on privilege settings and ownership.

Some of the UI elements are missing

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

The UI shows outdated elements

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

Clearing the Web Browser Cache

Clear and refresh your Web browser cache to ensure that out-of-date UI shell elements are no longer saved in your Web browser.

1. To clear the cache, choose the option in your Web browser to delete history and select cache from the options given.
2. To refresh the template cache, enter the following URL in your browser:

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

where hostname is your Serena ALM host server name.

Release trains are not appearing on the Calendar page

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

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

Deployment tasks for all release packages are shown in the report for each release package.

If the deployment tasks for all release packages are shown for each release package, the report may have the incorrect filter as described in this known issue, DEF231760.

To fix this issue, do the following:

1. In the native SBM user workspace (enter the Serena ALM URL without shell=alm), select the **Deployment Task** tab.
2. In the navigation pane, expand the **Reports** section.
3. Click **Find Reports**.
4. Search for the **Assigned Tasks** report.
5. Click **Modify Listing** to edit the report.
6. In the **Search filter** section, add following filters:

Release Package in (Query At Runtime)

and

Deployment Type in Instance

Create Release Fails with a Check Uniqueness Error

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

Error occurred during web service invocation:

SOAP Fault Code: env:Client

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

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

The Web services are using the wrong port number

The check uniqueness error may indicate that the Serena Release Manager Web services are defined for a different port number than the one on which the Serena Common Tomcat services are currently running. If you are using a non-default port number (not 8080), this message may indicate that you have not changed the port number in all of the Web services WSDLs.

An incorrect version of the Web services is present

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

The SBM connection properties are not set properly

Make sure that the `sbm-client-connection.properties` file has the correct URL for the Serena Release Manager to SBM connection, and that it has a valid and fully authorized username and password. This is now a required configuration to enable you to create objects in Serena Release Manager. If this connection is not configured properly, you will receive a checkuniqueness error.

Unable to update landing page layout

If a user cannot change the landing page, check their privileges for the `ALM_USER_SETTINGS` table in the SBM Application Administrator. To change the landing page layout in Serena ALM Settings, users must have the privileges Submit, Update, Delete, and View for the `ALM_USER_SETTINGS` table.

Matches Not Found for Selections

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

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

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

User credentials must match in both products

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

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

DimClientException ... Error: Not an authorized user

check to make sure that the same administrative password is set up with the same password in both SBM and Dimensions CM.

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

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

SSO must be selected for each RESTgrid widget

The security token may not be included in the URL the RESTgrid widget uses to pass to the SSO server. The **Use SSO authentication** check box is set by default for each RESTgrid widget to enable the security token to be included in the URL. See [Configuring SSO in RESTgrid Widgets \[page 37\]](#) for more details.

Provider file entries cannot be commented out or deleted

You cannot comment out or delete provider file entries. If you do, a null pointer is passed to the `DeploymentUnitSkeleton.getRLMService`.

If you have commented or deleted values in the provider configuration files, you must restore the entries, leaving values blank for any entries that you do not want to use. See [Provider Configuration \[page 121\]](#).

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

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

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

No Change Package Data Displayed in ZMF Deployment Tasks

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

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

Release Package Deployment Fails

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

A deployment vault task fails when you deploy a release package

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

If you are sure the task works in native Dimensions CM, check the **Activity Page** and **Activity Log** tab for information. If those do not give enough information for you to identify the problem, check the details in the `alm.log` file.

Verify that the connection requirements are met

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

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

If the deployment task fails with the error: Project was not found

If your deployment of a Dimensions CM or DVM deployment task fails with an error similar to the following, make sure that you specified the VAULT parameters in all upper case in the provider file, such as `dm_qlarius.properties`.

```
"Deployment of unit VAULT:V_QA_BASE1 to area(s) V_PRE-PROD_AREA has
failed : Project was not found"
```

```
"Promotion of deployment unit VAULT:V_QA_BASE1 to stage PRE-PROD has
failed : Project was not found."
```

This also causes a message similar to the following to be generated in the log file:

```
10-23@04:29:03 DEBUG [DimClientException.java:28] [max] :
DimClientException happened : Project was not found.

10-23@04:29:03 ERROR [RLMUtilServiceSkeleton.java:519] [max] : Error:
Deployment of unit VAULT:V_QA_BASE1 to area(s) V_PRE-PROD_AREA has
failed : Project was not found.
```

Check for an SbmDeployUnitHelper error message

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

Error

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

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

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

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

An automation deployment task fails when you deploy a release package

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

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

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

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.

Troubleshooting Development Manager

This section describes common problems that you may encounter when installing, configuring, or using Serena Development Manager

Installation Issues

- [Specifying Configurator SSO Support in ALM Installer \(DEF221902\) \[page 199\]](#)
- [QC Connector Does not Install when Using Remote Desktop \(DEF213757\) \[page 199\]](#)
- [Error Obtaining Security Token \(DEF224185\) \[page 200\]](#)

Upgrading Issues

- [Error Messages During Upgrade Installation and Common Tools not Installed \(DEF230170\) \[page 202\]](#)
- [Reports not Updated on Upgrade \(DEF221885\) \[page 203\]](#)
- [Create Dev CR Button not Appearing After Upgrade \[page 203\]](#)
- [Error Promoting Dev Change Requests Snapshot \(DEF217760\) \[page 204\]](#)
- [Promoting Snapshots Fails after Upgrade to DVM 4.0 \(DEF216946\) \[page 204\]](#)
- [Installation Interrupted Error on Upgrade when an Already Installed Component is deselected \(DEF212478\) \[page 205\]](#)

Development Control Issues

- [Cannot Unblock a Story That was Blocked in one Backlog and Then Moved to Another backlog \(DEF231681\) \[page 206\]](#)
- [Showing or Hiding Burndown Chart on Backlog Page Causes Firefox to Crash \(DEF231957\) \[page 206\]](#)
- [Add Task Button not Displayed on Taskboard \(DEF231797\) \[page 206\]](#)
- [Select Sprint drop down on Tasks Tab is Blank if There is only One Sprint \(DEF231799\) \[page 206\]](#)
- [Load Error for Affected Components When Editing a Project \(DEF231545\) \[page 206\]](#)
- [Automatic Transition of Dev Change Request not Occurring when Child Tasks Complete \(DEF226280\) \[page 206\]](#)
- [Dev Task not Updating after Associated Request is Actioned \[page 207\]](#)
- [Unexpected Error Appears when Editing an ALM Project with a Child Dev CR \(DEF226496\) \[page 208\]](#)
- [Error Occurs when Selecting an Existing ZMF Change Package with Correct Install Date \(DEF226545\) \[page 208\]](#)
- ["Please Contact Your Administrator" Message Appears on Forms \(DEF216549\) \[page 208\]](#)
- [It is Possible to Relate a ZMF Change Package to Multiple Dev Tasks \(DEF224772\) \[page 209\]](#)
- [Baseline Link in Dev Package not Functioning \(DEF226085\) \[page 209\]](#)

- Admin User Does not have Group Privileges for SBM Sample Data for SBM 10.1 [page 210]
- Actioning a Task in Dimensions CM to Complete Does not Transition the Task in SBM to Complete. [page 210]

Configuration Issues

- Incorrect Destination Endpoint Set when Promoting Dev Change Requests Snapshot (DEF231834) [page 210]
- Error When Deploying Process Apps [page 210]
- Missing Endpoints (DEF221077, DEF225893) [page 211]
- Parent CR Dev Complete rule invalid (DEF226370) [page 213]
- Warnings When Promoting Process Apps [page 214]
- Error Promoting Global Process Apps when Upgrading an Existing Development Manager Installation (DEF213024) [page 215]
- Groups and Users Unassigned from Projects when Upgrading an Existing Development Manager Installation (DEF211637) [page 215]
- User Groups Missing for Sample Database on MSSQL (DEF218177) [page 215]
- New Endpoints Created After Deploying Process Apps (DEF214586) [page 215]
- Put Files into Database not Functioning Correctly (DEF221461) [page 218]
- There is an Access Denied failure when Attempting to Create an Item [page 218]
- The REST Grids are not Returning Values [page 218]
- TASK requests are not Being created in CM When Creating Dev Tasks [page 218]
- A Baseline Cannot be Created from a Dev Package [page 218]
- The Dev Control Orchestrations Process Application Doesn't Handle Events from the Dev Tasks Process Application [page 219]
- The Dev Control Orchestrations Process Application Doesn't Handle Events from the ALM Projects Process Application [page 220]
- The Dev Control Orchestrations Process Application Doesn't Handle Events from the Dev Packages Process Application [page 222]

Installation Issues

Specifying Configurator SSO Support in ALM Installer (DEF221902)

It is advisable to only select the option **Configurator SSO Support** and configure the SSO details on the first occasion that you run the ALM Suite Installer. When you run the installer on subsequent occasions, you should deselect this option. If you configure these details more than once, SSO may not function correctly.

Also you should not use *localhost* for the server name, but specify the name of the actual server.

QC Connector Does not Install when Using Remote Desktop (DEF213757)

When installing Development Control via Remote Desktop access to a Windows 2003 system, installation may fail if you include the Connector to Quality Center option. To avoid this issue, run Remote Desktop in Console or Administration mode by running one of the following commands:

```
mstsc /admin
```

or

```
mstsc /console
```

Error Obtaining Security Token (DEF224185)

If when attempting to log on to the Serena ALM Configurator, you receive the error:

ALF SSO Gatekeeper error has occurred: Error obtaining security token.

Detail

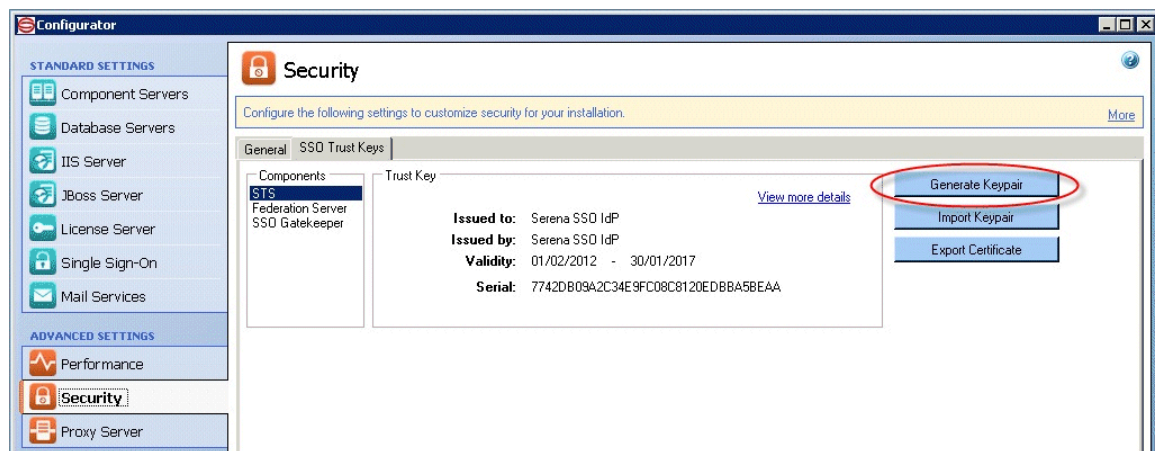
Validation of WS-Federation token failed with code 40:Token issuer not allowed.

You will need to update your SSO STS certificates by carrying out the following steps:

On the server where SBM is installed:

If you are using SBM 10.1 to 10.1.1.3:

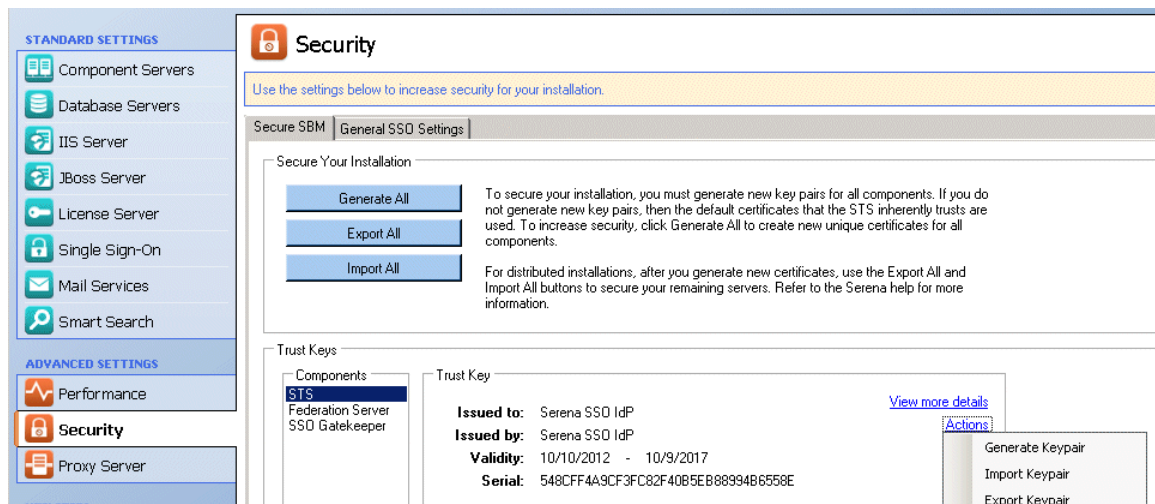
1. Start SBM Configurator and navigate to Security | SSO Trust Keys | STS.
2. Use the **Generate Keypair** button to regenerate the STS certificate.



3. Use the **Export Certificate** button to export the newly generated certificate to a .pem file (for example to C:\Temp\sts.pem).
4. Click **Apply** to apply the changes.

If you are using SBM 10.1.1.4:

1. Start SBM Configurator and navigate to Security | Secure SBM | Trust Keys | STS.
2. Under **Actions**, select **Generate Keypair** to regenerate the STS certificate.



3. Under **Actions**, select **Export Certificate** to export the newly generated certificate. Save this as a .pem file (for example to C:\Temp\sts.pem).
4. Click **Apply** to apply the changes.

On each machine where the Serena Common Tomcat is installed:

1. Stop Tomcat.
2. Copy the sts.pem file to that machine (for example to C:\Temp\sts.pem).
3. Under the folder where the common Tomcat is installed, for example:
 - C:\Program Files\Serena\Common Tools\tomcat\6.0\...
 - C:\Program Files\Common\tomcat\6.0\...
 - C:\Program Files\Serena\Dimensions 12.2\Common Tools\tomcat\6.0
4. Navigate to
 - \alfssogatekeeper\conf
 and make a backup of the truststore.jks file.
5. Under the Common Tools or Common folder, Navigate to
 - \jre\6.0\bin.
6. Delete the existing STS certificate, for example using the following command:


```
keytool -delete -keystore "C:\Program Files\Serena\Common Tools\tomcat\6.0\alfssogatekeeper\conf\truststore.jks -alias sts
```

 Enter the default password *changeit* when prompted.
7. Import the certificate from the C:\Temp\sts.pem file into truststore.jks, for example by using the following command:


```
keytool -import -keystore "C:\Program Files\Serena\Common Tools\tomcat\6.0\alfssogatekeeper\conf\truststore.jks -file "C:\temp\sts.pem" -alias sts
```

 After entering the password, when prompted with *Trust this certificate? [no]:*, reply **Y**.
8. Restart Tomcat.

If this is the Dimensions CM Server:

Also do the following:

1. Stop the Dimensions Listener service.
2. Create a backup of

C:\Program Files\Serena\Dimensions 12.2\CM\dfs\sts.pem

3. Copy the file C:\Temp\sts.pem to

C:\Program Files\Serena\Dimensions 12.2\CM\dfs\sts.pem

4. Restart the Dimensions Listener service.

Upgrading Issues

Error Messages During Upgrade Installation and Common Tools not Installed (DEF230170)

A problem may occur when you are running the ALM Suite installer to upgrade to ALM 4.0.1.

Errors appear during installation, such as:

Failed to access file: C:\Program Files\Serena\Dimensions 12.2\Common Tools\tomcat\6.0\conf\web.xml to configure sso

and

Service Serena Common Tomcat (SerenaTomcat) failed to start

and the new Serena Common Tools are not installed.

To remedy this situation, the following files should be removed from the Serena Common Tools installation location.

Stop the Serena Common Tools service before deleting these files:

C:\Program Files\Common\tomcat\6.0.pre4.0.1\alfssogatekeeper\lib\alfssocommon-3.3-SNAPSHOT.jar

C:\Program Files\Common\tomcat\6.0.pre4.0.1\alfssogatekeeper\lib\alfssocommon-wsmodel-3.3-SNAPSHOT.jar

C:\Program Files\Common\tomcat\6.0.pre4.0.1\alfssogatekeeper\lib\alfssocommon-wstrust-client-3.3-SNAPSHOT.jar

C:\Program Files\Common\tomcat\6.0.pre4.0.1\alfssogatekeeper\lib\alfssogatekeeper-3.3-SNAPSHOT.jar

C:\Program Files\Common\tomcat\6.0.pre4.0.1\alfssogatekeeper\lib\alfssojnibridge-3.3-SNAPSHOT.jar

C:\Program Files\Common\tomcat\6.0.pre4.0.1\lib\alfssogatekeeper-filterloader-3.3-SNAPSHOT.jar

C:\Program Files\Common\tomcat\6.0.pre4.0.1\lib\commons-logging-api.jar

C:\Program Files\Common\tomcat\6.0.pre4.0.1\lib\log4j.jar

C:\Program Files\Common\tomcat\6.0\webapps\SS0TokenGenerator\WEB-INF\alfssogatekeeper\lib\alfssocommon-3.3-SNAPSHOT.jar

C:\Program Files\Common\tomcat\6.0\ webapps\SS0TokenGenerator\WEB-INF\alfssogatekeeper\lib\alfssocommon-wsmodel-3.3-SNAPSHOT.jar

C:\Program Files\Common\tomcat\6.0\ webapps\SS0TokenGenerator\WEB-INF\alfssogatekeeper\lib\alfssocommon-wstrust-client-3.3-SNAPSHOT.jar

C:\Program Files\Common\tomcat\6.0\ webapps\SS0TokenGenerator\WEB-INF\alfssogatekeeper\lib\alfssogatekeeper-3.3-SNAPSHOT.jar

C:\Program Files\Common\tomcat\6.0\ webapps\SS0TokenGenerator\WEB-INF\alfssogatekeeper\lib\alfssojnibridge-3.3-SNAPSHOT.jar

Reports not Updated on Upgrade (DEF221885)

Some reports for Release 4.0 that have been updated from the previous release do not override the previous versions that are referenced in the process apps. These are:

Process Application	Report name	Reference Name
Dev Change Request	CR(s) associated to this Dev Package	CR4DP
Dev Change Request	CR(s) associated to this Project	CR4Proj
Development Packages	Dev Package(s) associated with this Project	DP4Proj
Dev Tasks	Task(s) associated with this CR	Task4CR
Dev Tasks	CM Task(s) associated with this Dev Package	Task4DP

To resolve this issue, perform **one** of the following actions:

- Before promoting the process apps, delete the reports. The new versions of the reports will appear in the list after promotion.
- After promoting the process apps, edit the old versions of the reports to remove or change the reference name (this will remove the link to the old reports in the process apps). Then edit the new reports (that will be displayed in the Web UI in red) and set the correct reference names as shown above.
- After promoting the process apps, create your own reports and specify the reference names shown above to link the reports to the process apps.

ZMF Dev Tasks are shown in Dimensions CM Dev Package (DEF224942)

If you have a project that has both Dimensions CM and ZMF selected as providers, there is a problem whereby ZMF dev tasks are displayed on the Associated Tasks tab of the Dev Package view if you have upgraded from a previous version of Development Manager.

The solution to this problem is to delete the report "Task(s) associated to this dev package" from the Dev Tasks application before promoting the Release 4.0 process apps.

Create Dev CR Button not Appearing After Upgrade

There is a problem where, after upgrading to DVM 4.0 from a previous version, the **Create Dev CR** button does not appear on the Change Requests tab for an ALM project. This can be remedied by using the following steps:

1. Open SBM Composer.
2. Deploy the *Dev Change Requests* process app.

3. Deploy the *ALM Projects* process app.

Error Promoting Dev Change Requests Snapshot (DEF217760)

On promoting the Dev Change Requests snapshot after upgrading to DVM 4.0 from a previous version, the promotion may fail with the following error appearing in the log file:

Import: Incoming project already exists at this level. Name: 'Dev Change Requests Project', Existing UUID: '60b85e4d-b11a-47c8-9f2c-66674945a733'

The solution is to select None for Projects on the Entities page of the Promote wizard.

Promoting Snapshots Fails after Upgrade to DVM 4.0 (DEF216946)

When performing an upgrade to Development Manager 4.0, promoting snapshots may fail with an error in the log file such as:

```
ERROR -- Cannot deploy BPEL definition for process model alf/
9c629f6d-0686-41d5-a41a-1c079ba973fa/ - 500: could not execute query. ORA-00923:
FROM keyword not found where expected.
```

Failed to complete the deployment to server "Default BPEL Server" at 4/23/12 12:18 PM.

The following steps should resolve the problem:

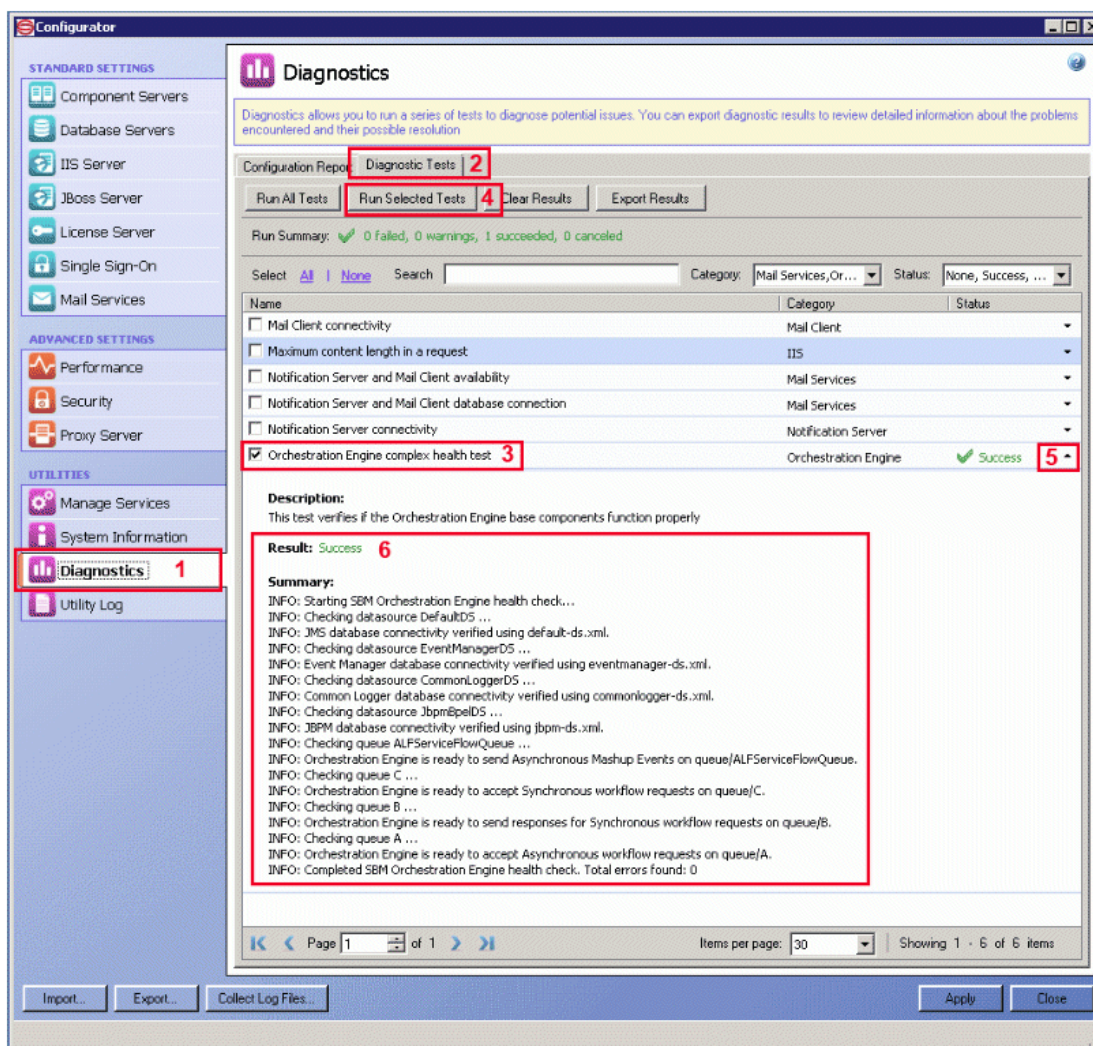
1. Open SBM Configurator.
2. Restart the Serena Common JBOSS service.
3. Do one of the following:
 - If you are using SBM 10.1 or later:

Make sure that the file:

```
C:\Program
Files\Serena\SBM\Common\jboss405\server\default\log\sbm_oe_health.log
```

Doesn't contain any errors.

- If you are using SBM 10.1.1.1:
 - a. Select Diagnostics
 - b. Click the **Diagnostic Tests** button.
 - c. Select **Orchestration Engine complex health test**.
 - d. Click the **Run Selected Tests** button.



4. If there are errors found from either of the above, stop and restart Serena Common JBOSS.
5. Promote the snapshots.

Installation Interrupted Error on Upgrade when an Already Installed Component is deselected (DEF212478)

If upgrading an installation and deselecting an option that was previously installed, the installer fails. You will need to run the installer without deselecting an already installed component.

Upgrading of the QCC Connector

If you have updated your version of SBM since the last time you installed the Serena Connector for Quality Center, you must uninstall the existing version of the QC Connector before running the installer for Serena ALM 4.0.

This is because the QC Connector is installed against a specific version of SBM. If the SBM version entered during the install of the QC Connector as part of ALM 4.0 does not match that entered during the original installation of the QC Connector, the upgrade does not take place.

Development Control Issues

Cannot Unblock a Story That was Blocked in one Backlog and Then Moved to Another backlog (DEF231681)

A problem occurs when a story is blocked in a backlog and then subsequently moved to another backlog. When attempting to unblock the story in the new backlog, an error occurs "Server Error".

Showing or Hiding Burndown Chart on Backlog Page Causes Firefox to Crash (DEF231957)

There is a problem where, under certain conditions, clicking the *Show Burndown* or *Hide Burndown* link on a Release or Sprint backlog page may cause Firefox browser to crash under certain circumstances. The solution is to avoid clicking this link or to use a different browser.

Add Task Button not Displayed on Taskboard (DEF231797)

On the taskboard for a sprint, the add button for adding a task is only shown for the last story in the list. You can, however, add a task by selecting the story, selecting the Tasks tab and clicking the New Task button.

Select Sprint drop down on Tasks Tab is Blank if There is only One Sprint (DEF231799)

On the Tasks tab for the Tasks application there is a drop-down list that enables you to select a sprint that you can display using the *Open Taskboard* link. If you have only created one sprint in the Agile workspace, the dropdown list is blank. Once you have created more than one sprint, the drop down list will correctly display the list of sprints.

Load Error for Affected Components When Editing a Project (DEF231545)

When Editing an ALM project, you may encounter a problem where the Affected Components list contains the error ! Load error! (error). This is due to the fact that the Project Manager editing an ALM Project needs to have User role access to the *Dev Change Requests* and *Dev Tasks* projects. This can be remedied by granting them the User role in those projects via the SBM Administrator.

Automatic Transition of Dev Change Request not Occurring when Child Tasks Complete (DEF226280)

There is a problem when using MSSQL on an SBM Server where an automatic transition to Development Complete is not occurring for dev change requests when the child dev tasks are all complete.

To remedy this, it is necessary to perform an update on the MSSQL SBM sample database:

1. In SBM Configurator, stop the Serena Common JBOSS and IIS services. (you may find it necessary to stop IIS twice.)
2. Ensure that ALLOW_SNAPSHOT_ISOLATION and READ_COMMITTED_SNAPSHOT are disabled (at least one of the values equals 0) by performing the following SQL command:

```
SELECT
    snapshot_isolation_state,
    is_read_committed_snapshot_on
FROM sys.databases
```

```
WHERE name= '<SBM_DATABASE_NAME>'
```

Where <SBM_DATABASE_NAME> is the name of the SBM database.

3. Check whether there are active connections to the SBM database. (If your SBM database is not returned in the list it means that there are no connections.)

```
SELECT
```

```
DB_NAME(dbid) as DBName,
```

```
COUNT(dbid) as NumberOfConnections,
```

```
loginame as LoginName
```

```
FROM sys.sysprocesses
```

```
WHERE dbid > 0 and DB_NAME(dbid) not in ( 'master', 'tempdb')
```

```
GROUP BY dbid, loginame
```

If there are any connections, close them:

```
USE master
```

```
GO
```

```
ALTER DATABASE sample SET OFFLINE WITH ROLLBACK IMMEDIATE
```

```
ALTER DATABASE sample SET ONLINE
```

4. Apply the following sql statement:

```
ALTER DATABASE sample SET ALLOW_SNAPSHOT_ISOLATION ON
```

```
ALTER DATABASE sample SET READ_COMMITTED_SNAPSHOT ON WITH NO_WAIT
```

5. Ensure that ALLOW_SNAPSHOT_ISOLATION and READ_COMMITTED_SNAPSHOT are enabled (both values equal 1):

```
SELECT
```

```
snapshot_isolation_state,
```

```
is_read_committed_snapshot_on
```

```
FROM sys.databases
```

```
WHERE name= '<SBM_DATABASE_NAME>'
```

Where <SBM_DATABASE_NAME> is the name of the SBM database.

6. In SBM Configurator, restart the Serena Common JBOSS and IIS services.

Dev Task not Updating after Associated Request is Actioned

If you encounter a problem where a request has been actioned in Dimensions CM but the corresponding dev task in SBM has not been transitioned, it is likely that the user involved needs to be assigned the role DEVELOPMENT ENGINEER with Capability SECONDARY.

To do this:

1. Open the Dimensions CM Administration Console as a member of the Administration group.
2. Under **Users and Roles**, select **Role assignments | By user**

3. Select the user and select the **Add** button
4. Set the following values:
 - **Assignment Type:** Delegation Candidate.
 - **Project Role:** DEVELOPMENT ENGINEER
 - **Design Part:** The top level design part for the project, for example QLARIUS
 - **Role Capability:** Secondary
5. Click **OK**.

Unexpected Error Appears when Editing an ALM Project with a Child Dev CR (DEF226496)

If you have an ALM Project with a child dev change request, and you click the **Edit** button, a web service error may occur. This can be remedied by creating a child dev task for the associated dev change request and then performing the edit.

Error Occurs when Selecting an Existing ZMF Change Package with Correct Install Date (DEF226545)

There is a problem when creating a dev task having previously selected the **Create a new ChangeMan ZMF package for this task** checkbox, and Set **From Install Date/Time** to the current date. If you subsequently uncheck the **Create a new ChangeMan ZMF package for this task** checkbox, select an existing ZMF change package, and click **OK** you will receive an error.

One or more fields are invalid.

From Install Time must be in the future!

To remedy this, after clicking on **Attach ZMF package**,

1. Remove the contents of the **From Install Date/Time** field in the *Create a new "ChangeMan ZMF package for this task* form.
2. Select the **Use an existing ChangeMan ZMF package for this task** checkbox.

"Please Contact Your Administrator" Message Appears on Forms (DEF216549)

If you experience a problem where the message Please Contact Your Administrator keeps appearing on various forms, it is likely that a report that is being used in an embedded report widget is missing. This will cause the error to appear, and not cleared for a period of time, thus appearing on other pages. The solution is to make sure all reports exist and that the widgets are properly calling them.

To check that the reports are set up correctly:

1. In the user environment view an item (almproj,devcr,devpack,tasks,ram).
 2. View each tab that has an embedded report.
- If you see the report showing data, or even if you just see the columns and there is no data, then that report is setup correctly
 - if the report is not there and it says something like *This report may be deleted, you do not have permissions to see this report*, or any other error message, then the report has not been set up correctly.

To create a report with a reference name:

1. Go to the process application (in the SBM user environment) where the report lives (For example, if the report is *All CR's in a Project* then you want the CR tab.)

2. Click reports in the bottom left
3. Click on browse application reports
4. Find the name of the report that is missing.
5. (For example for *CR's associated to a Project* look for "CRs associated to this project")
6. Click the **Edit** button.
7. Click **Save As**.
8. Copy and paste the reference name from the comment into the **Reference Name** field.
9. Give the report a unique name (for example, change "CRs" to "CR(s)")
10. Save the report.

It is Possible to Relate a ZMF Change Package to Multiple Dev Tasks (DEF224772)

In the `dvmzmf_packages.properties` file, it is possible to change the following variables from the default values of **Y** to the following:

```
FILTER_ZMF_PACKAGES_BY_WORKREQNO = N
```

```
RELATE_DVM_DT_ITEMID_TO_ZMF_PACKAGE_WORKREQNO = N
```

If you have done this, it becomes possible to relate the same ZMF Change Package to more than one Dev Task, as no error message is displayed when this happens. The intended behavior for the out-of-the-box workflows is that there is a one-to-one relationship between a ZMF ChangePackage and a Dev Task in Development Manager.

Baseline Link in Dev Package not Functioning (DEF226085)

When clicking the URL link to a baseline from the Baselines tab of a dev package, an error may occur stating *There were one or more problems with the requested link* and the baseline is not displayed.

This problem is caused by duplicate entries in the `gatekeeper-core-config.xml` that occur when selecting the option to configure SSO support on multiple upgrades/installations.

To remedy the problem:

1. On the Dimensions CM server, Locate the file `gatekeeper-core-config.xml` in the Serena Common Tomcat installation, for example:

```
C:\Program Files\Serena\Dimensions 12.2\Common Tools
\tomcat\6.0\alfssogatekeeper\conf\gatekeeper-core-config.xml
```

In this file there should be entries for three types of parameter: `SecurityTokenService`, `SecurityTokenServiceExternal`, and `FederationServerURL`, for example:

```
<parameter name="SecurityTokenService"
Type="xsd:anyURI">http://sbmservname:8085/TokenService/services/
Trust</parameter>
```

```
...
```

```
...
```

```
<parameter name="SecurityTokenServiceExternal"
Type="xsd:anyURI">http://sbmservname:8085/TokenService/services/
Trust</parameter>
```

```
...
```

...

```
<parameter name="FederationServerURL"
Type="xsd:anyURI">http://sbmservname:8085/ALFSS0Login/login</parameter>
```

2. If there are multiple entries for each of these parameters, delete the duplicate entries, leaving only the currently valid one for each.

Admin User Does not have Group Privileges for SBM Sample Data for SBM 10.1

When using SBM 10.1, the Admin user for the SBM sample database does not have access to any groups in the Web Administrator. The solution to this is to create a user with Remote Administration privileges, and use this user to grant the privileges for the groups for the Admin user.

1. Open SBM System Administrator and select the Users tab.
2. Add a new User as a Managed Administrator.
3. On the Privileges tab, make sure Remote Administration is selected.
4. Save the details and log out.
5. Log in as the user you created above.
6. On the Users tab, select the Admin user and select Edit.
7. Select Administration and select Groups.
8. Select all the groups and click OK.

Actioning a Task in Dimensions CM to Complete Does not Transition the Task in SBM to Complete.

When actioning a task request from Peer Review to Complete in Dimensions CM and delegating the Development Engineer role (for example to user Amy) the corresponding task in SBM is not transitioned to Complete in SBM.

To remedy this, the transition in the TASK lifecycle in Dimensions CM needs to have DEVELOPMENT ENGINEER assigned as a Pending role. For details, see [Assign Development Engineer Role to TASK Lifecycle Transitions \[page 88\]](#).

Configuration Issues

Incorrect Destination Endpoint Set when Promoting Dev Change Requests Snapshot (DEF231834)

During the promotion of the Dev Change Requests snapshot, the destination endpoint for DVMUtilServices is set to DevManagerServices instead of DVMUtilServices. This appears on the Mapping page during promotion. To remedy this, select the endpoint, click Choose Destination Endpoint, and select the endpoint DVMUtilServices.

Error When Deploying Process Apps

If you receive an error in SBM Composer when deploying a process app:

Unexpected exception during DefinitionUtilities.ExportDefinition

It is likely that the process app contains references to another process app that needs to be deployed before it. The process apps need to be deployed in the order specified below:

1. Global Process App (eval)
2. Requirements Approval Management (if you are using Requirements Manager)

3. ALM Projects
4. Dev Change Requests
5. Dev Tasks
6. Dev Package
7. Dev Control Orchestrations
8. Products and Components
9. Quality Center Connector (if you are using HP Quality Center)
10. RM to RQM Connector (if you are using Requirements Manager)
11. Release Train (if you are using Release Manager)
12. Application Release (if you are using Release Manager)
13. Release Package (if you are using Release Manager)
14. Deployment (if you are using Release Manager)
15. Environment (if you are using Release Manager)
16. ReleaseTemplate (if you are using Release Manager)
17. RLM_AUX
18. Vault_Request (if you are using Release Manager)

Missing Endpoints (DEF221077, DEF225893)

There is a problem whereby when you are promoting the Serena ALM snapshots, some of the endpoints are not set up correctly. This means that the default endpoint is not automatically selected when you promote the process apps, and the server name is not set correctly. These endpoints are:

- DVMUtilService,
- RLMUtilService,
- ReleaseRequestService

To remedy this:

When promoting each process app:

On the *Current Mappings* page, if the field under the **Destination** column is not set to an endpoint:

1. Select any Source that does not have a Destination Endpoint and click the **Choose Destination Endpoint** button.
2. Select the endpoint from the list.

If the Destination Endpoint has not yet been defined for the selected Source, click **Create a new endpoint**. You will need to do this for RLMUtilService and ReleaseRequestService.

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>/alm/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 alm/services directory of the common Tomcat Web server.

Example endpoints with the default port number are shown in the following table.

Source	Destination Endpoint
RLMUtilService	http://almhost:8080/alm/services/RLMUtilService
ReleaseRequestService	http://almhost:8080/alm/services/ ReleaseRequestService

- c. In the **Authentication** field, select **Security Token**.
- d. Click **Test Connection** to test the connection.
- e. Click **OK**.
- f. Select the Source Endpoint again and click **Choose Destination Endpoint**. Select the endpoint you just created.

When updating the endpoints:

(For details, see [Updating the Endpoints \[page 39\]](#) in the Development Manager Installation and Configuration Guide)

From SBM Application Administrator:

1. Click **Environments**.
2. Select the environment you used for the promotions.
3. Select the **Endpoints** tab.
4. Check that the server name portion of the URL for each endpoint is set to localhost or the name of the SBM server. If it is not:
 - a. Select the endpoint.
 - b. Click the Edit button
 - c. Change the server name to localhost,

for example, DVMUtilService should be:

`http://localhost:8080/alm/services/DVMUtilService`
5. Click **OK**.

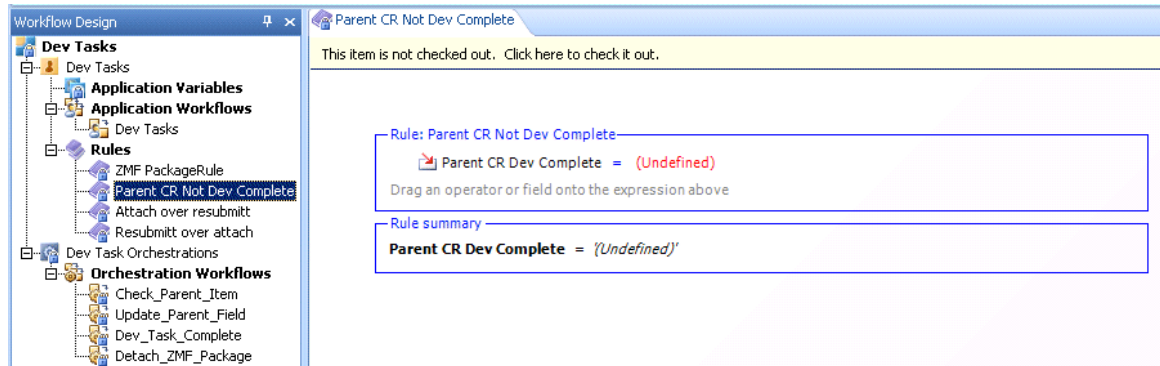
Deploy the relevant process apps after updating the endpoints.

Parent CR Dev Complete rule invalid (DEF226370)

When Deploying the process apps you may receive an error because some rules with sub-relational field dependencies are not set. There are four rules for which this may occur.

To remedy the problem, you will need to set the undefined rules to the correct value.

1. In SBM Composer, open the Dev Tasks process app.
2. Under **Rules**, select *Parent Dev CR not Complete*.

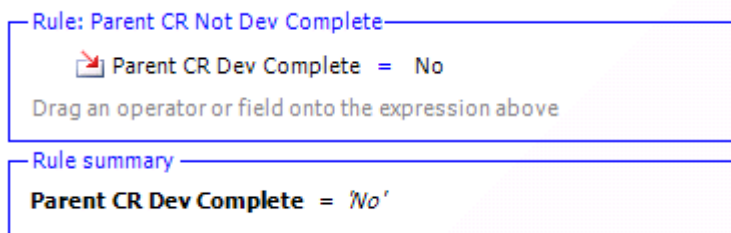


3. If the rule is undefined, change it to *Parent CR Dev Complete = No*.
4. Repeat this for the three rules in the Dev Packages process app as specified below.

The rules that may need to be set are:

- Dev Tasks:
- Parent Dev CR not Complete:

Parent CR Dev Complete = No




- Dev Packages:
- ConfMgmtSys Contains only ChangeMan ZMF:

Configuration Management System(s) contains any ChangeMan ZMF

AND

Configuration Management System(s) does not contain any Dimensions CM

—Rule: ConfMgmtSys contains only ChangeMan ZMF—



- Configuration Management System(s) contains any ChangeMan ZMF
- Configuration Management System(s) does not contain any Dimensions CM

—Rule summary—

Configuration Management System(s) contains any 'ChangeMan ZMF' **And** **Configuration Management System(s)** does not contain any 'Dimensions CM'


- ConfMgmtSys Contains only Dimensions CM

Configuration Management System(s) contains any Dimensions CM

AND

Configuration Management System(s) does not contain any ChangeMan ZMF

—Rule: ConfMgmtSys contains only Dimensions CM—



- Configuration Management System(s) contains any Dimensions CM
- Configuration Management System(s) does not contain any ChangeMan ZMF

—Rule summary—

Configuration Management System(s) contains any 'Dimensions CM' **And** **Configuration Management System(s)** does not contain any 'ChangeMan ZMF'


- ConfMgmtSys Contains only (None)

Configuration Management System(s) contains any (none)

OR

Configuration Management System(s) does not contain all ChangeMan ZMF, Dimensions CM

—Rule: ConfMgmtSys contains only (None)—



- Configuration Management System(s) contains any (None)
- Configuration Management System(s) does not contain all ChangeMan ZMF, Dimensions CM

—Rule summary—

Configuration Management System(s) contains any '(None)' **or** **Configuration Management System(s)** does not contain all 'ChangeMan ZMF, Dimensions CM'

Continue deploying the process apps after making these changes.

Warnings When Promoting Process Apps

When promoting the process apps, you may see "Unable to create field" warning messages in the promote log. For example:

```
WARN -- Unable to create field 'alm_Originating Baseline' in table 'Dev Packages' because relational table 'Related Vault Baselines' is not defined.
```

```
WARN -- Unable to create field 'Vault Template' in table 'Dev Packages' because relational table 'Vault_Request' is not defined.
```

Such messages will not affect the success of the promotions or the functioning of Development Manager, and can be ignored.

Error Promoting Global Process Apps when Upgrading an Existing Development Manager Installation (DEF213024)

If you are upgrading a previous installation of Development Manager, you may receive an error in SBM Application Administrator when promoting the *Global Process App (eval)* process app.

This problem can be overcome by setting the following values on the Global Entities page of the Promote Wizard.

- **Groups:** None
- **Users:** None

Groups and Users Unassigned from Projects when Upgrading an Existing Development Manager Installation (DEF211637)

When upgrading an existing installation of Development Manager, you may find that some groups and users are unassigned from projects after promoting snapshots.

This problem can be overcome by setting the following values on the Global Entities page of the Promote Wizard.

- **Groups:** None
- **Users:** None

User Groups Missing for Sample Database on MSSQL (DEF218177)

If you are using MSSQL, the user groups are not present for the sample data. In this case you will need to assign the necessary roles for the users to enable them to use the applications using SBM Administrator. For details, see [Creating and Assigning the Users \[page 75\]](#).

New Endpoints Created After Deploying Process Apps (DEF214586)

When you have redeployed the process apps, for example after updating the application links in the Event Mappings as described in the post-installation steps, you may find that duplicate endpoints have been created, such as

dmwebservices_2

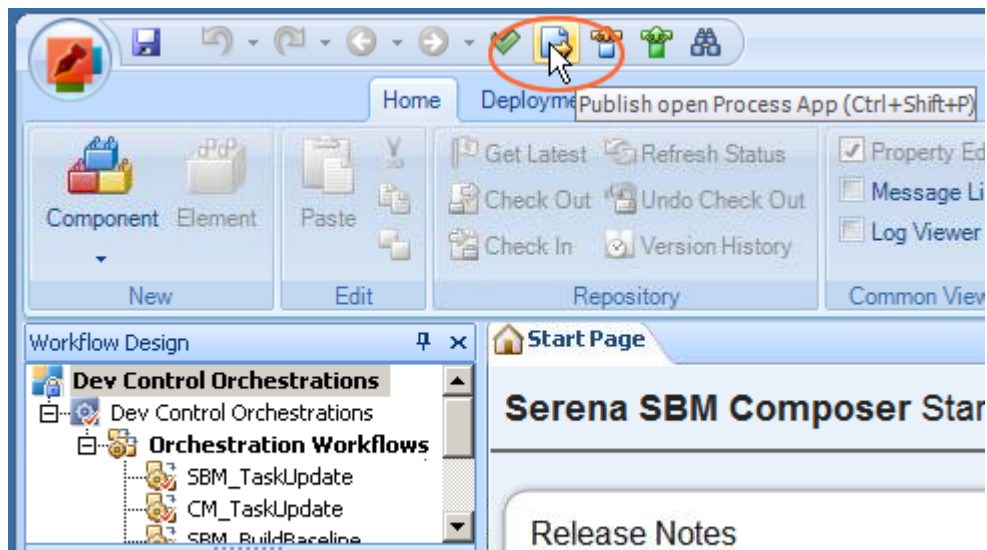
DevManagerServices_2

This results in an error in SBM Composer such as:

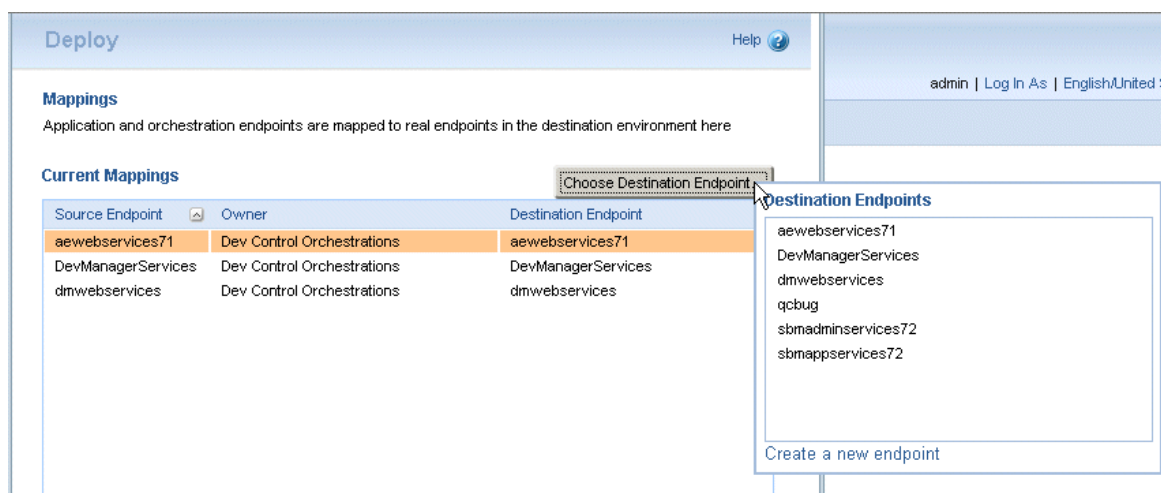
The Orchestration Engine cannot send the Web service request at service step CM_Create_Standard_Baseline to the endpoint ...

The solution is to use SBM Application Administrator to perform the deployments.

1. Open each process app in SBM Composer
2. Click the **Publish** button in the toolbar



3. In the Publish Process App dialog box, select **Allow others to deploy this version of the process app** and click **Publish**.
4. Open SBM Application Administrator by entering <http://localhost:8085/mashupmgr/> in a browser.
5. Select **Process Apps** on the left.
6. Select a process app and click the **Deploy** button
7. Click in the Destination field.
8. Select the environment and click Next.
9. Check that the list of endpoints is correct.



10. Click Done and complete the deployment.
11. Repeat [Step 6 \[page 216\]](#) through [Step 10 \[page 216\]](#) for the other process apps.

The User Interface is not Being Displayed (ENH213101)

The User Interface for Development Manager should have a blue and grey color scheme when you log in using the URL:

<http://<hostname>/tmtrack/tmtrack.dll?shell=alm>

If the correct User Interface is not being displayed you will need to perform the following steps:

1. Open the zip file \Solutions\Development Control\com.serena.dvm.shell.zip, in the Serena installation folder, for example:

```
C:\Program Files\Serena\Solutions\Development Control
\com.serena.dvm.sbm.shell.zip
```

2. Extract the bin and template folders to the Application Engine folder, for example:

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

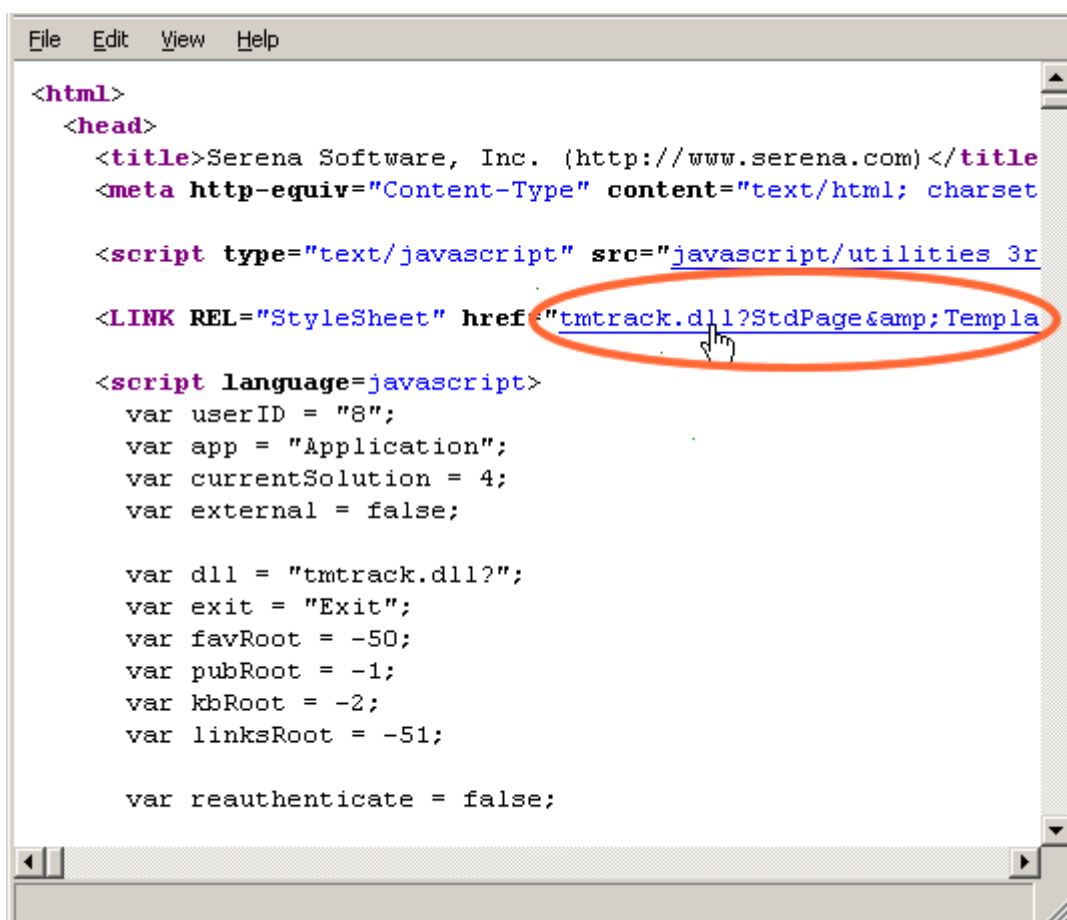
Click **Yes to all** to overwrite the files

3. Perform the steps described in Installing the UI Files
4. Open the Development Manager application by entering:

```
http://<hostname>/tmtrack/tmtrack.dll?shell=alm
```

In a Web browser

5. If the stylesheet is not refreshing you may need to refresh them manually:
 - a. Right click on the browser window, and select **View Page Source**



- b. Click on the link for the stylesheet.
This will open the .css file from the cache.
- c. In the stylesheet window, select **View | Reload** or press F5.
- d. Close and reopen Development Manager.

Put Files into Database not Functioning Correctly (DEF221461)

You may experience a problem whereby having installed the ALM Suites and performed the action File | Put Files into Database using SBM System Administrator, and then made changes to the shell files and performed Put Files into Database again, that old versions of the files are still present in SBM. This means that you will need to replace the files again and repeat the Put Files into Database action.

This problem can be avoided by performing File | Get Files From Database in SBM System Administrator before running the ALM Suite installer.

There is an Access Denied failure when Attempting to Create an Item

If a process is attempting to create an item, and there is an error, for example "[ERROR] Access Denied," in the log, and the item is not created, then it is likely that you do not have access privileges for the connector user. Carry out the following steps:

1. Open the SBM System Administrator.
2. Select users.
3. Select the connector user (for example DMSYS) and click Edit.
4. Select the Administration tab.
5. Select **Global Administration** and click OK.

The REST Grids are not Returning Values

Check whether:

1. The Development Manager web services is running.
2. You have configured the Development Manager web services correctly.
3. If SSO is enabled, ensure that the user you are using in SBM also exists in Dimensions CM.
4. Make sure the SBM services JBOSS and IIS have been restarted, using SBM configurator for example.

TASK requests are not Being created in CM When Creating Dev Tasks

1. Deploy the Dev Control Orchestrations process app.
2. Restart the server.

If this does not work, check whether you are using the connector account (for example dmsys). You should not be using the connector account to create items in SBM, it is reserved for connecting only.

A Baseline Cannot be Created from a Dev Package

If when trying to create a baseline from a dev package, no baseline gets created in Dimensions CM, and the following appears in the Windows Event Viewer:

Error occurred in file:

'.\WSInvoker.cpp', line 1287.

Error mapping data with field 'Baseline Build Configuration'

you can resolve the event viewer error by unlocking the event definition.

Do the following:

1. In SBM Composer open the Dev Packages application.
2. Show Extensions.
3. Expand *Orchestration Links*.
4. Click on **Event without Reply**.
5. Check the item out.

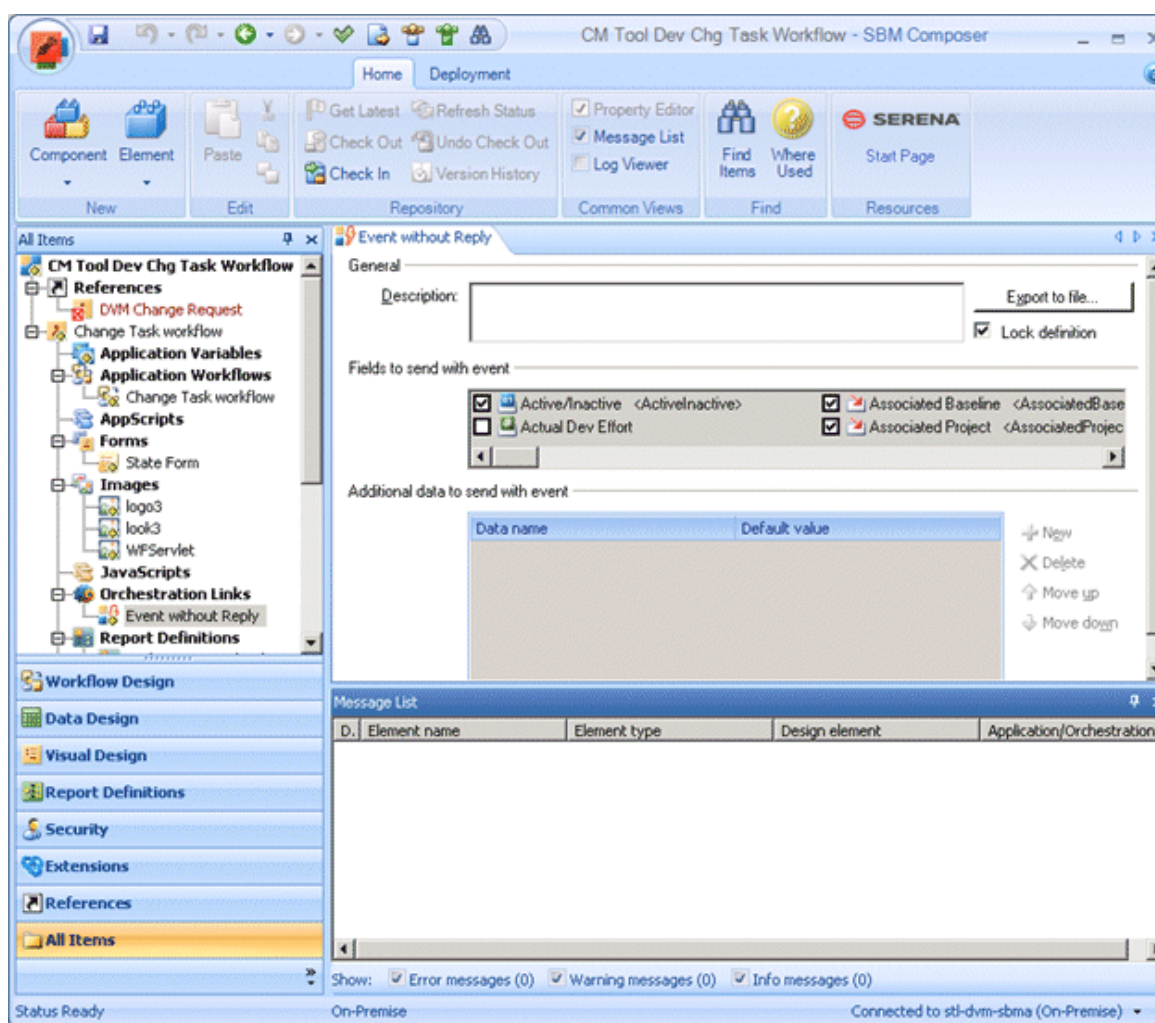
Ensure that **Lock definition** (on the right-hand side) is **not** checked.

6. Check in the item and re-deploy.

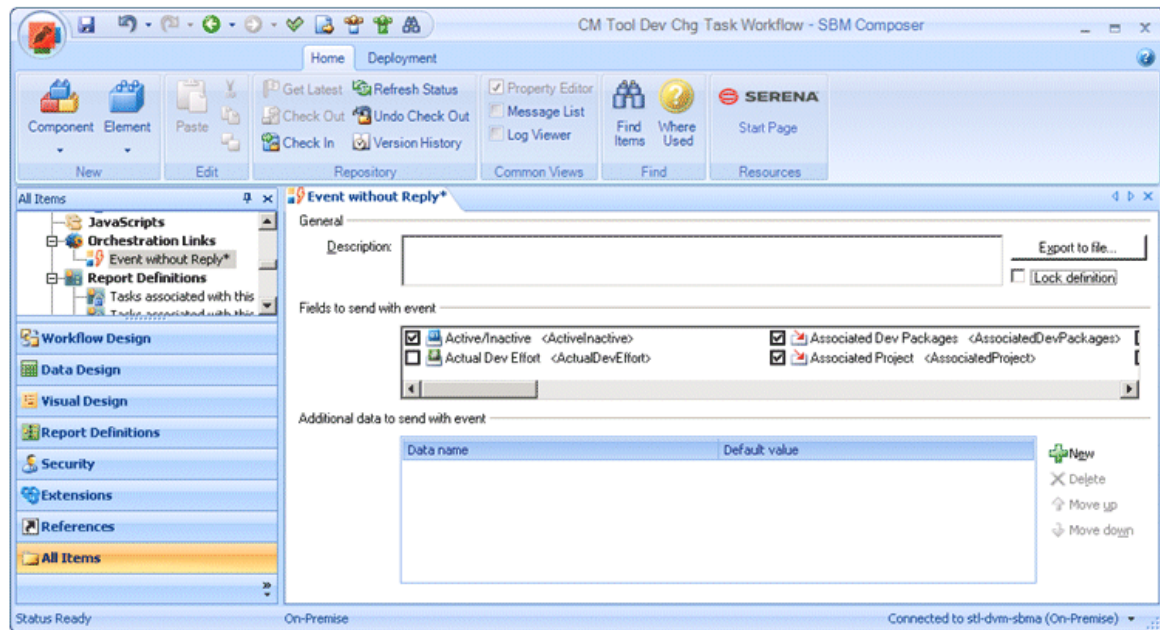
The Dev Control Orchestrations Process Application Doesn't Handle Events from the Dev Tasks Process Application

You need to recreate ChangeTaskworkflowEventDefinition:

1. In SBM Composer, Open the Dev Tasks process application.
2. Under Orchestration Links, select **Event without Reply**.



3. Uncheck **Lock definition**.
4. Click **Export to file...** , type the file name and export the Orchestration link to a file.



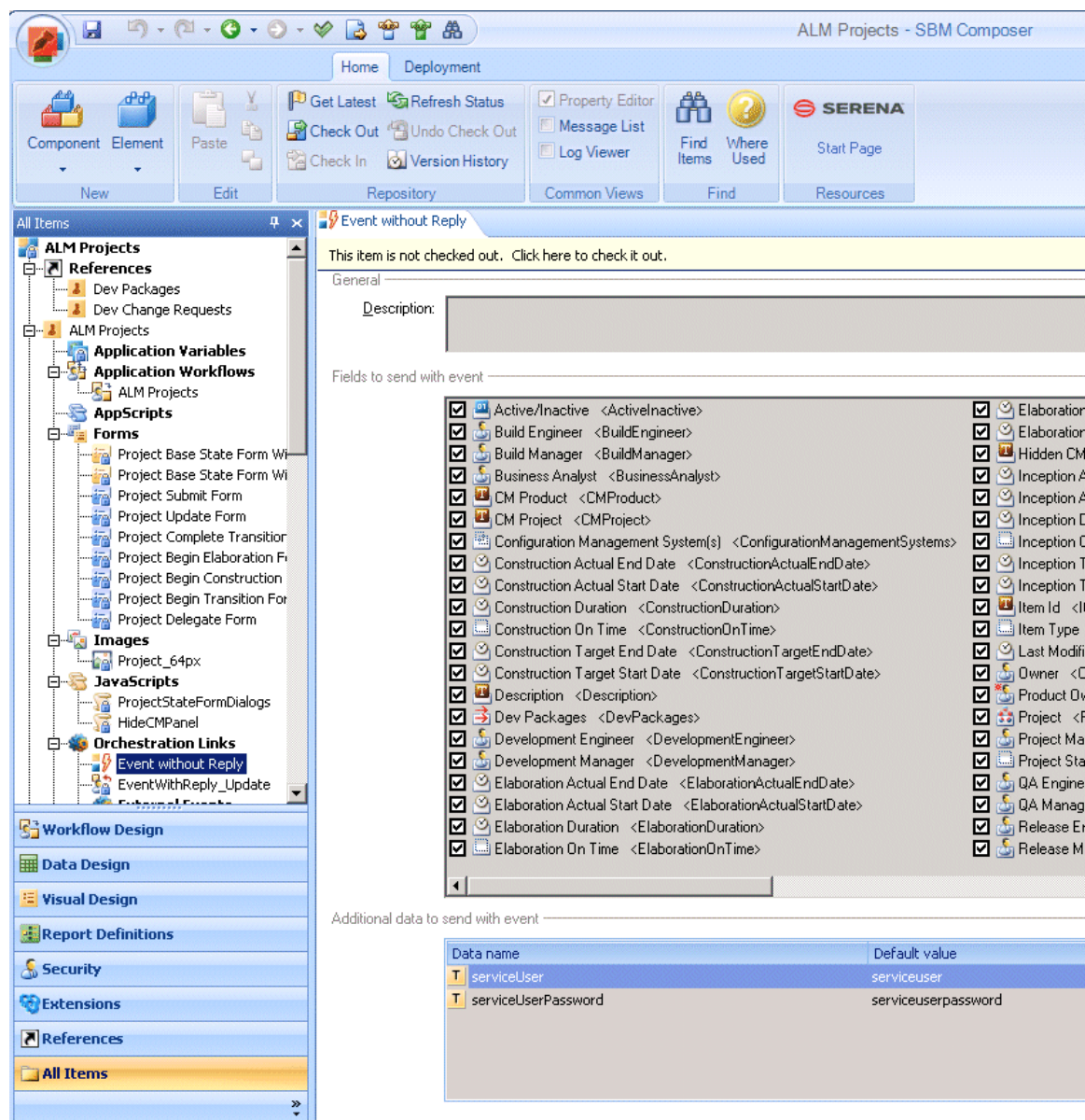
5. Click Yes to automatically lock the definition
6. Check in the item and re-deploy.

The Dev Control Orchestrations Process Application Doesn't Handle Events from the ALM Projects Process Application

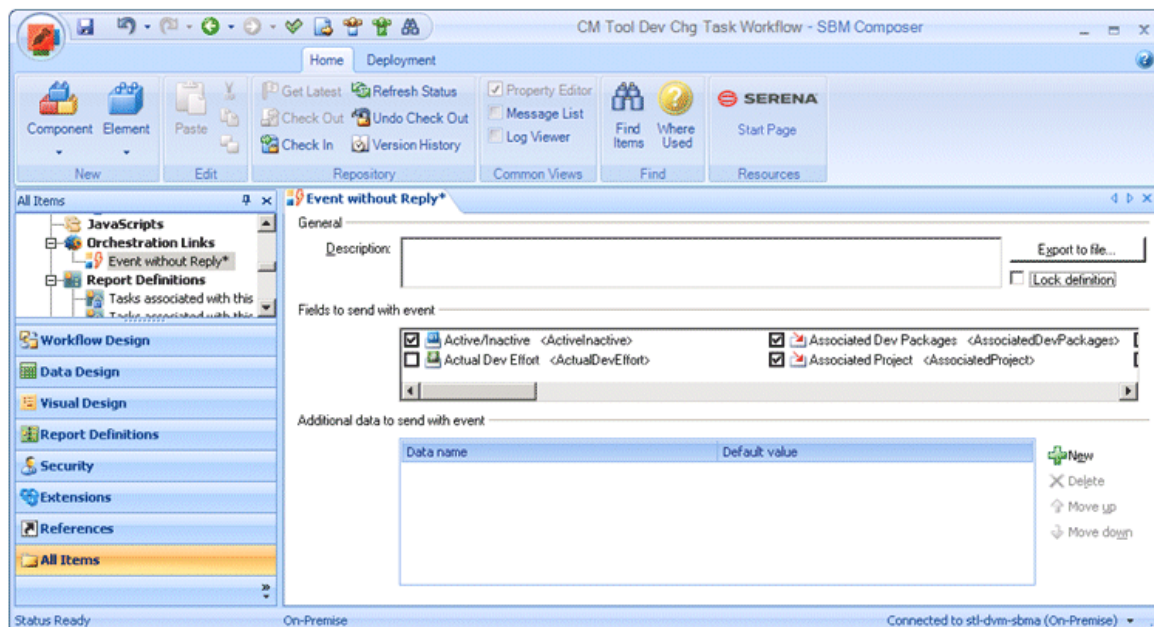
In this case, you need to recreate

DevelopmentControlChangeRequestWorkflowEventDefinition.

1. In SBM Composer, open the ALM Projects process application.
2. Select **Orchestration Links**. Links can be exported from here.
3. Select the **Event without Reply** orchestration link.



4. Uncheck **Lock definition**.
5. Click **Export to file...** , type the file name and export the Orchestration link to a file.

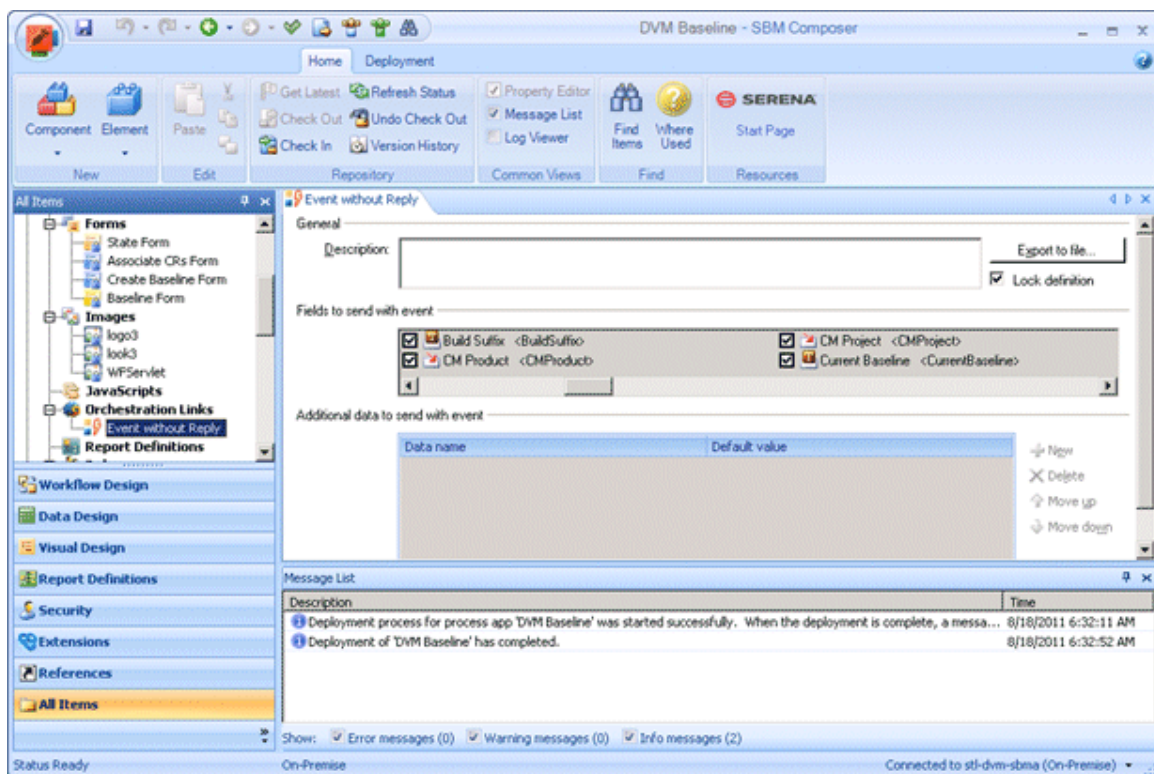


6. Check in the item and re-deploy.

The Dev Control Orchestrations Process Application Doesn't Handle Events from the Dev Packages Process Application

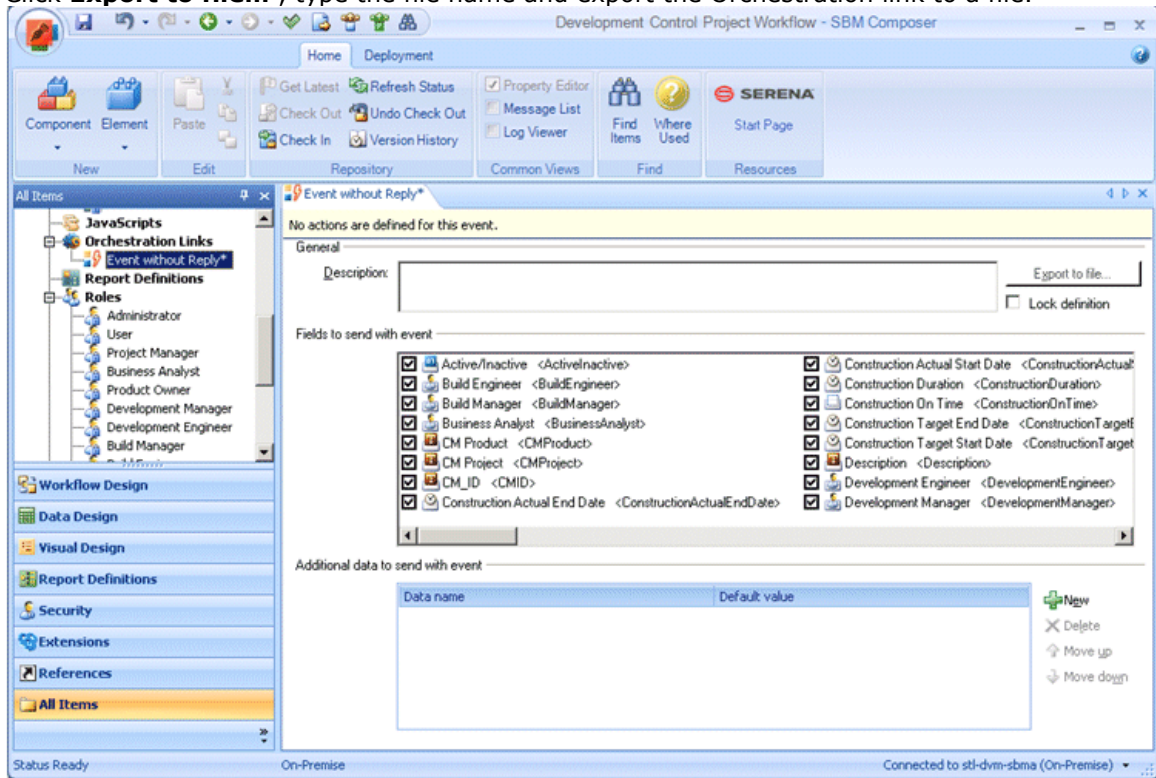
You need to recreate DVMBaselineEventDefinition.

1. In SBM Composer, open the ALM Projects process application.
2. Select **Orchestration Links**. Links can be exported from here.
3. Select the **Event without reply** orchestration link.



4. Uncheck **Lock definition**.

5. Click **Export to file...** , type the file name and export the Orchestration link to a file.



6. Check in the item and re-deploy.

Serena ALM Workflow Reference

Each major aspect of Release Manager, Development Control, and Requirements Control has a lifecycle associated with it. The default release process lifecycles for each Serena ALM solution are based on typical ALM process workflows.

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

[Release Manager Workflows \[page 224\]](#)

[Development Manager Workflows \[page 238\]](#)

Release Manager Workflows

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 \[page 224\]](#)

[Release Train Workflow \[page 226\]](#)

[Application Release Workflow \[page 227\]](#)

[Release Package Workflow \[page 228\]](#)

[Deployment Task Workflows \[page 231\]](#)

[Deployment Process Template Workflow \[page 235\]](#)

[Environment Workflow \[page 236\]](#)

[Vault Request Workflow \[page 237\]](#)

[Vault Template Workflow \[page 238\]](#)

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.

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

Workflow Restrictions

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

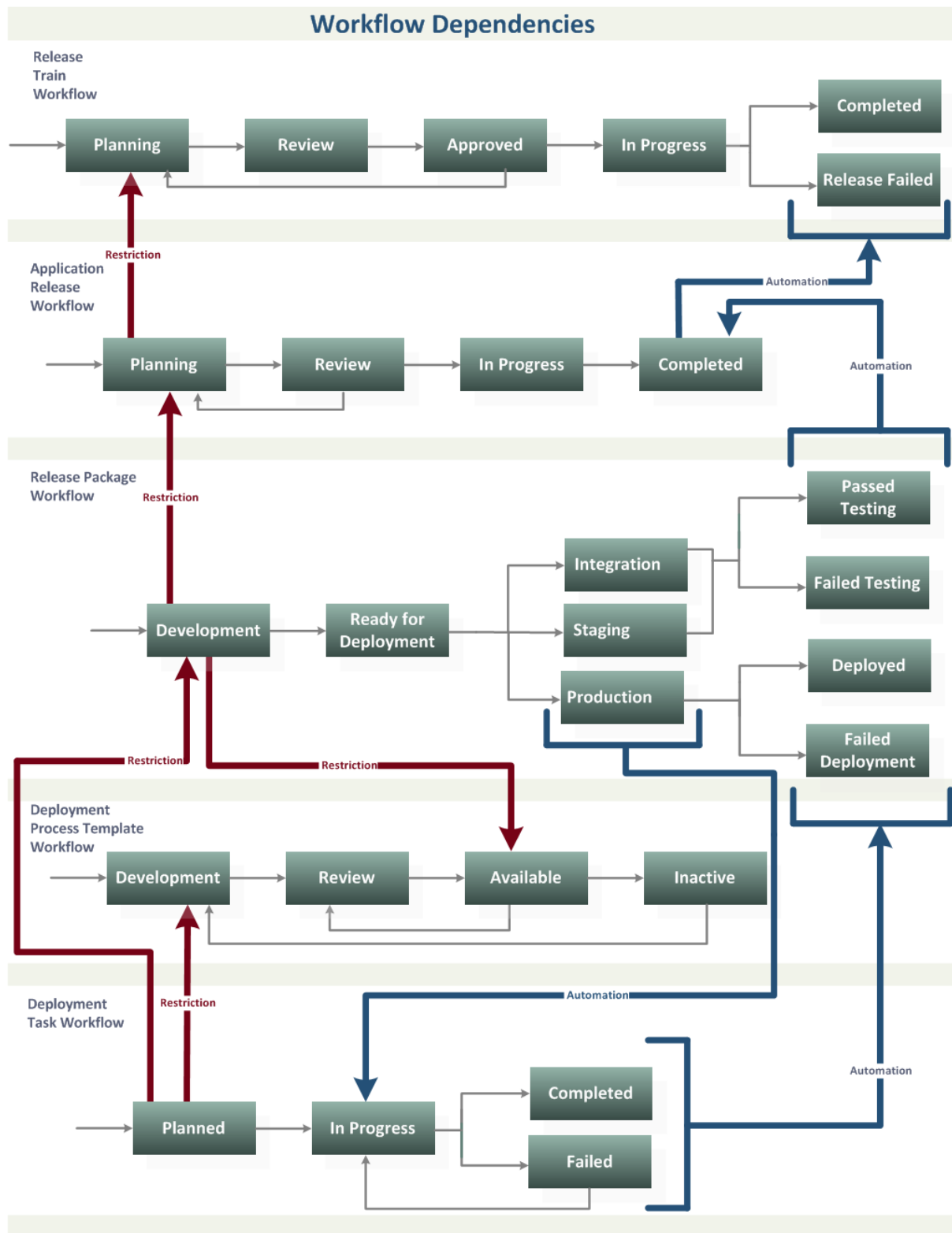
Workflows for Deployment Process Templates are standalone, although deployment process templates populate the deployment tasks for release packages and in that respect have a link to

the workflows for both Deployment Tasks and Release Packages. Workflows for environments are completely standalone by default.

Workflow Automations

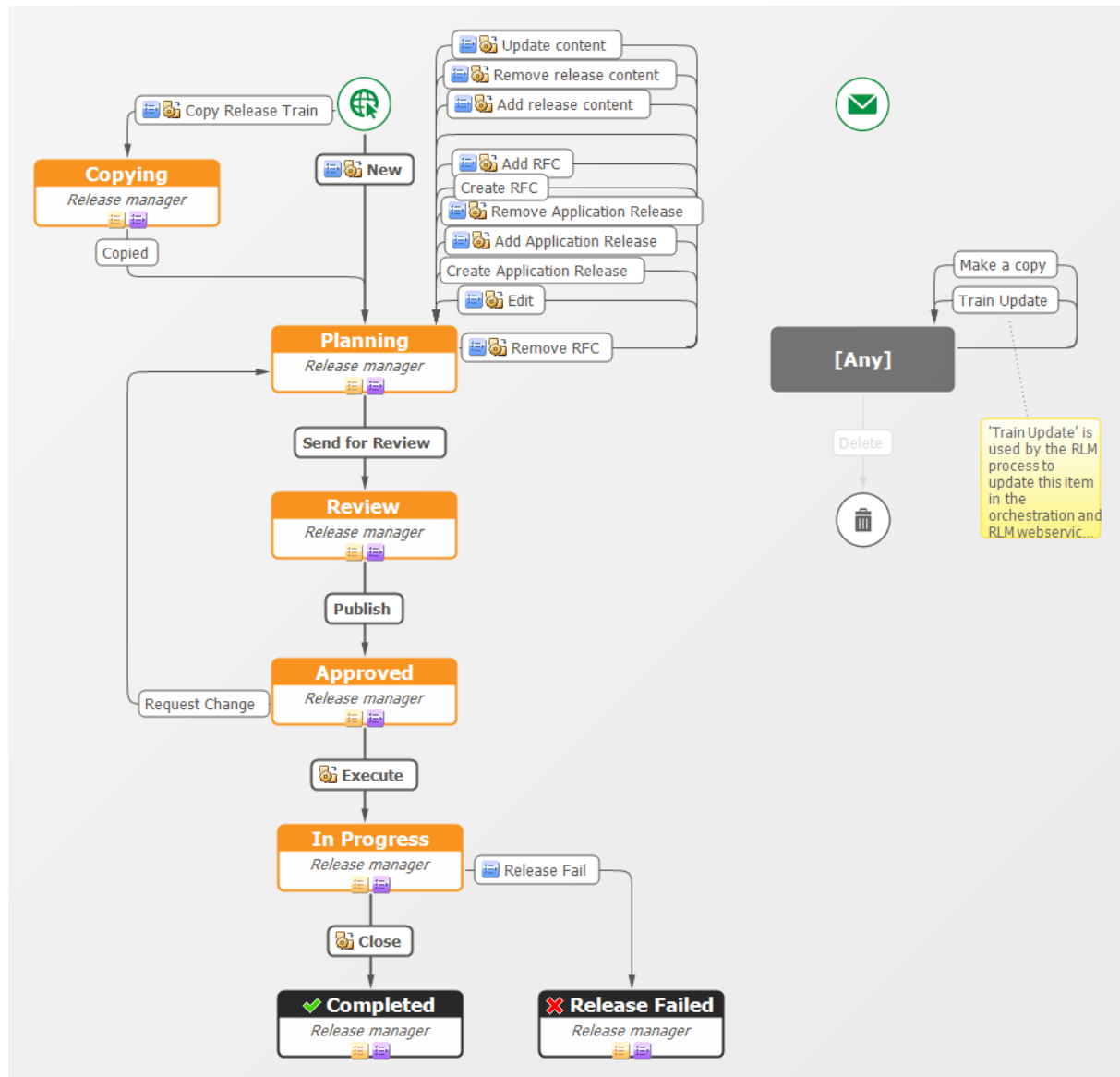
Many of the workflows are moved to completion states automatically when associated objects are complete. The automation is usually initiated at the lowest level of association and moves up the hierarchy as each level of release activities are completed. For example, after all deployment tasks for a release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

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



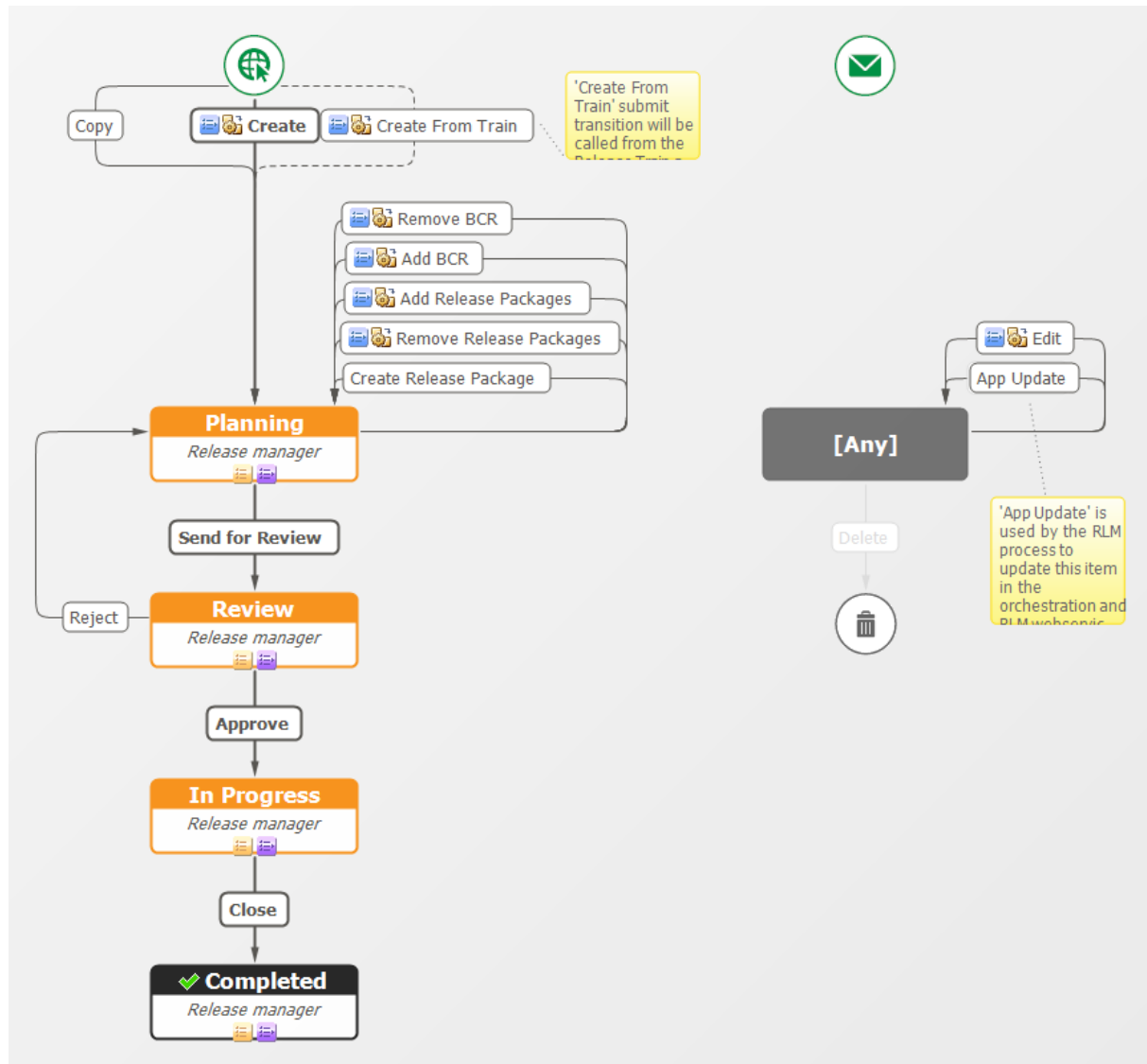
Release Train Workflow

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.

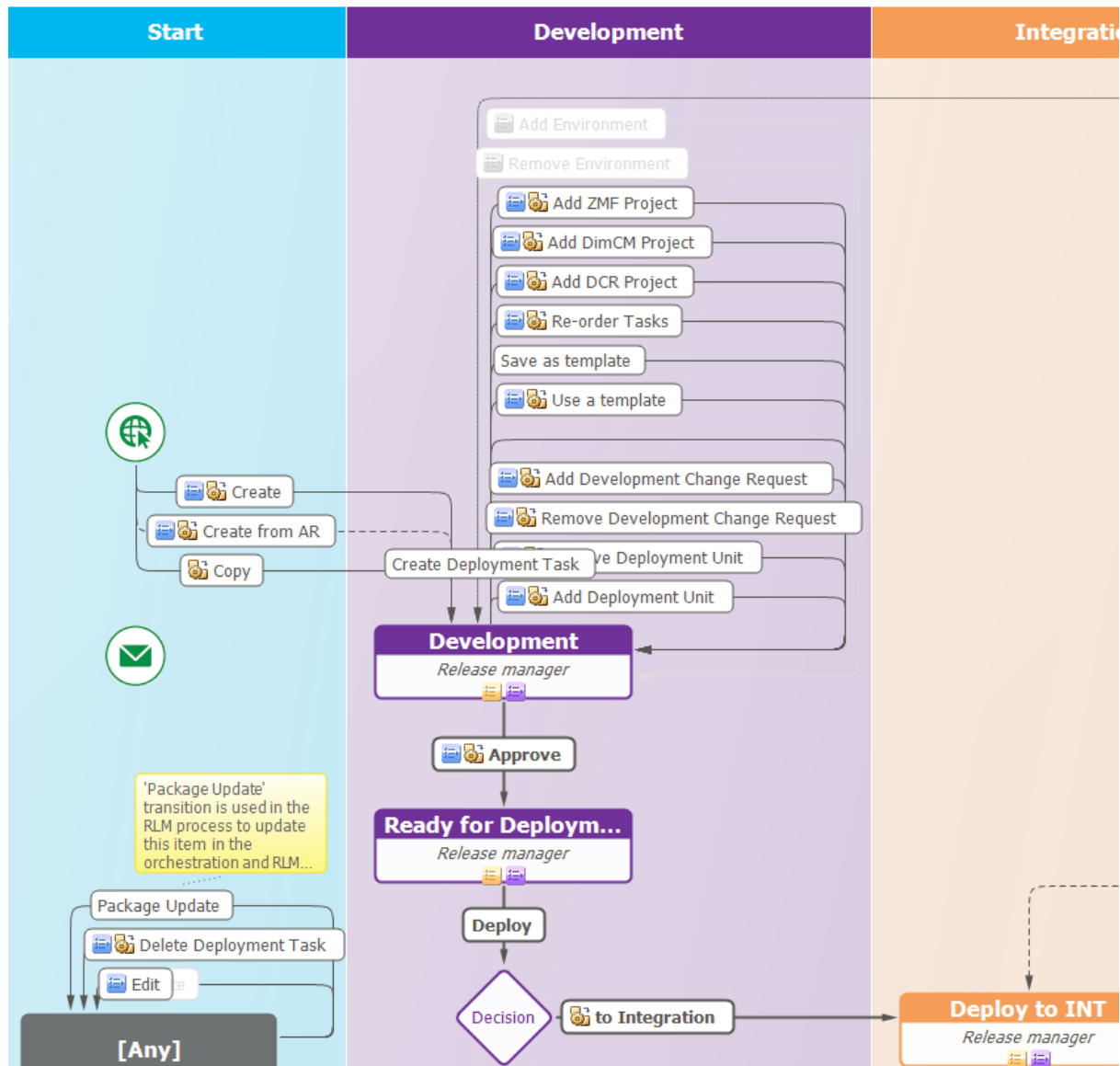


Release Package Workflow

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.

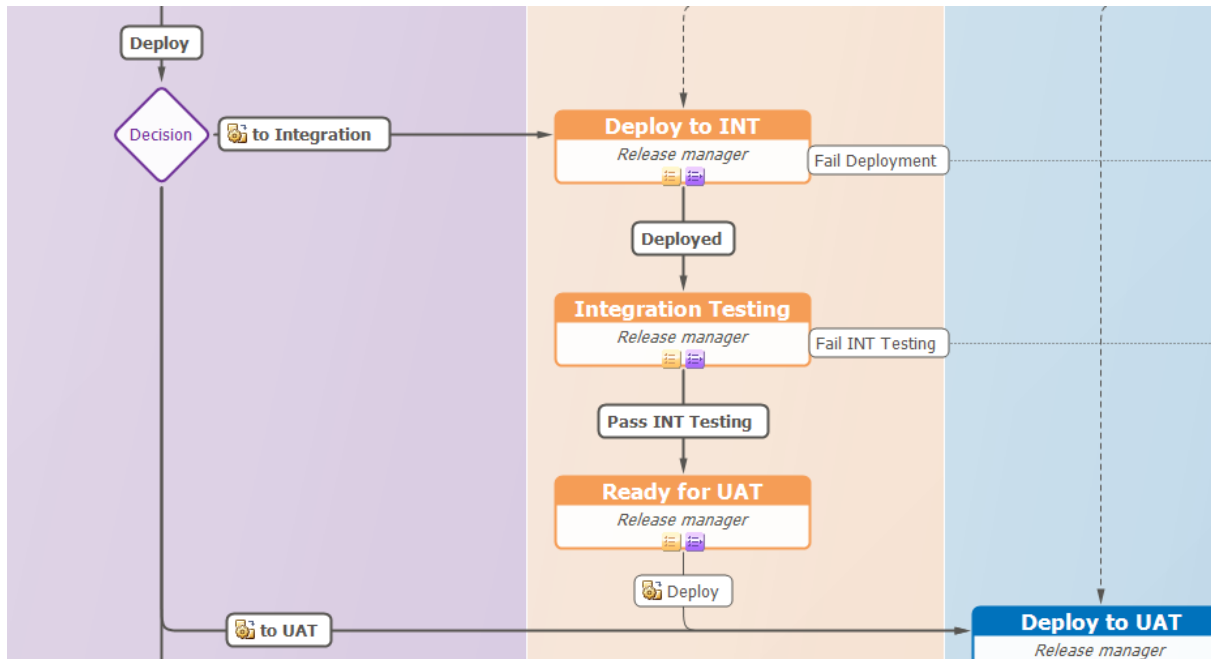
Start and Development States

The Start and Development states swimlanes of the Release Package workflow in the default version of Serena Release Manager are shown in the following figure.



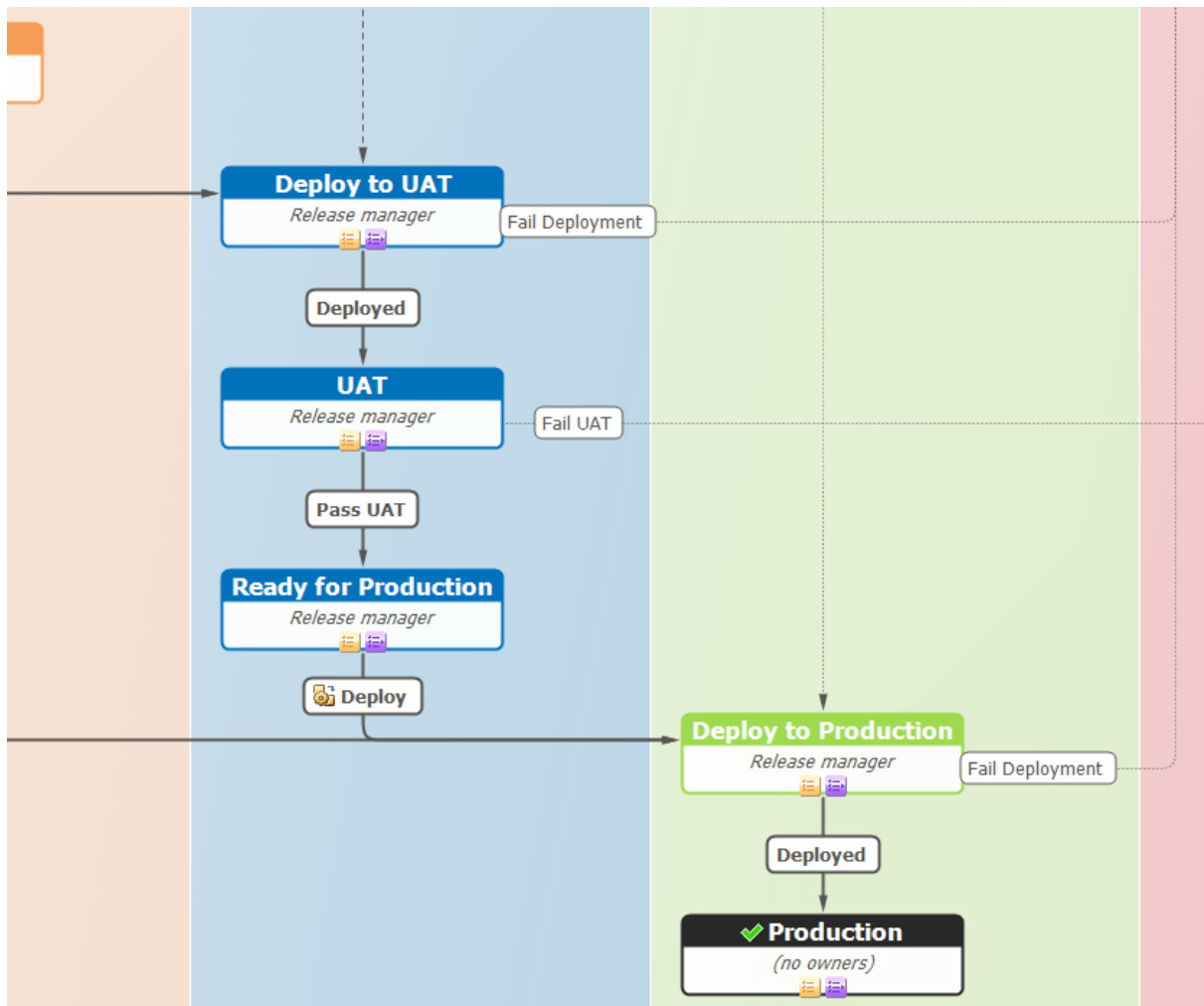
Integration State

The Integration state swimlane of the Release Package workflow in the default version of Serena Release Manager is shown in the following figure.



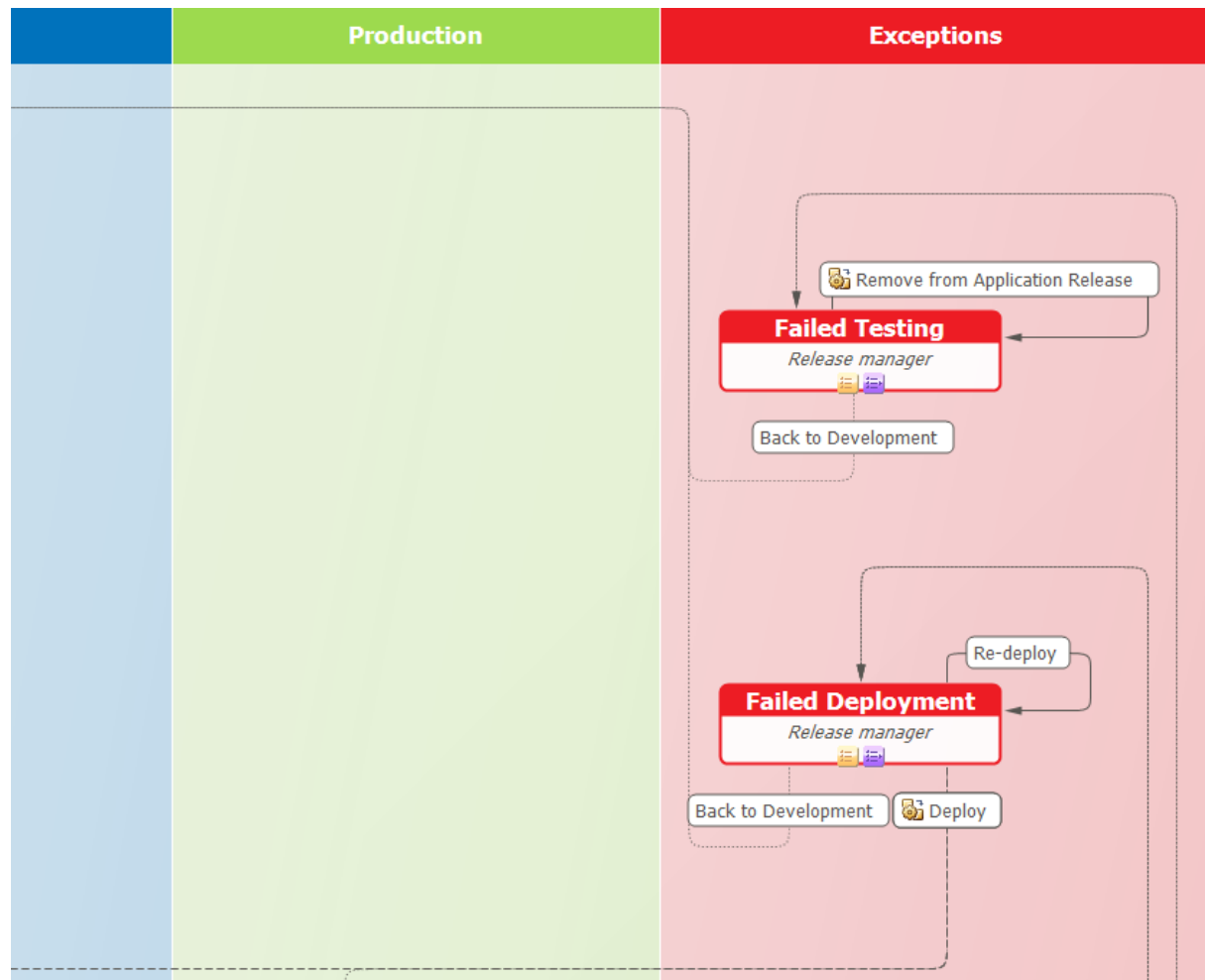
Staging and Production States

The Staging and Production states swimlanes of the Release Package workflow in the default version of Serena Release Manager are shown in the following figure.



Exceptions State

The Exceptions state swimlane of the Release Package workflow in the default version of Serena Release Manager is shown in the following figure.

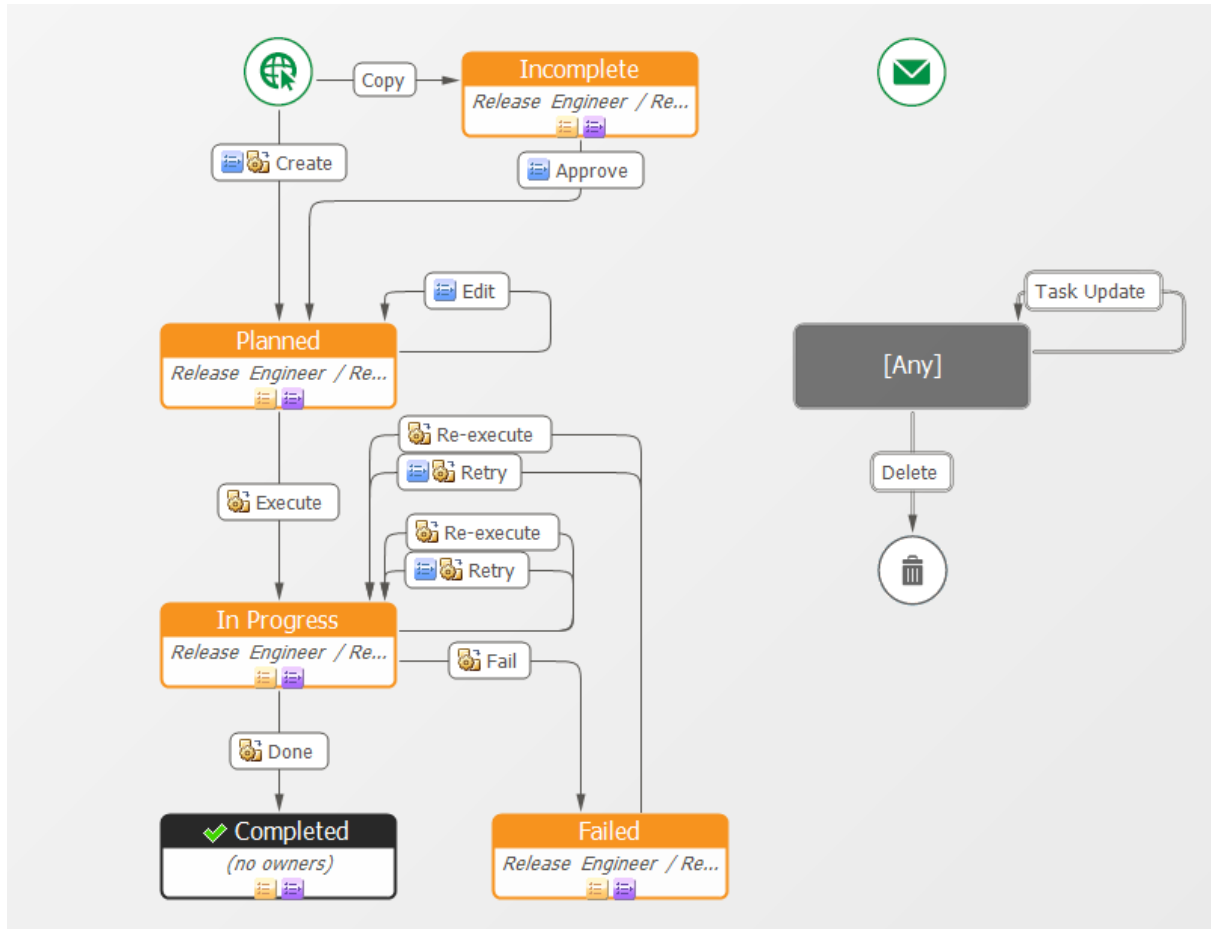


Deployment Task Workflows

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

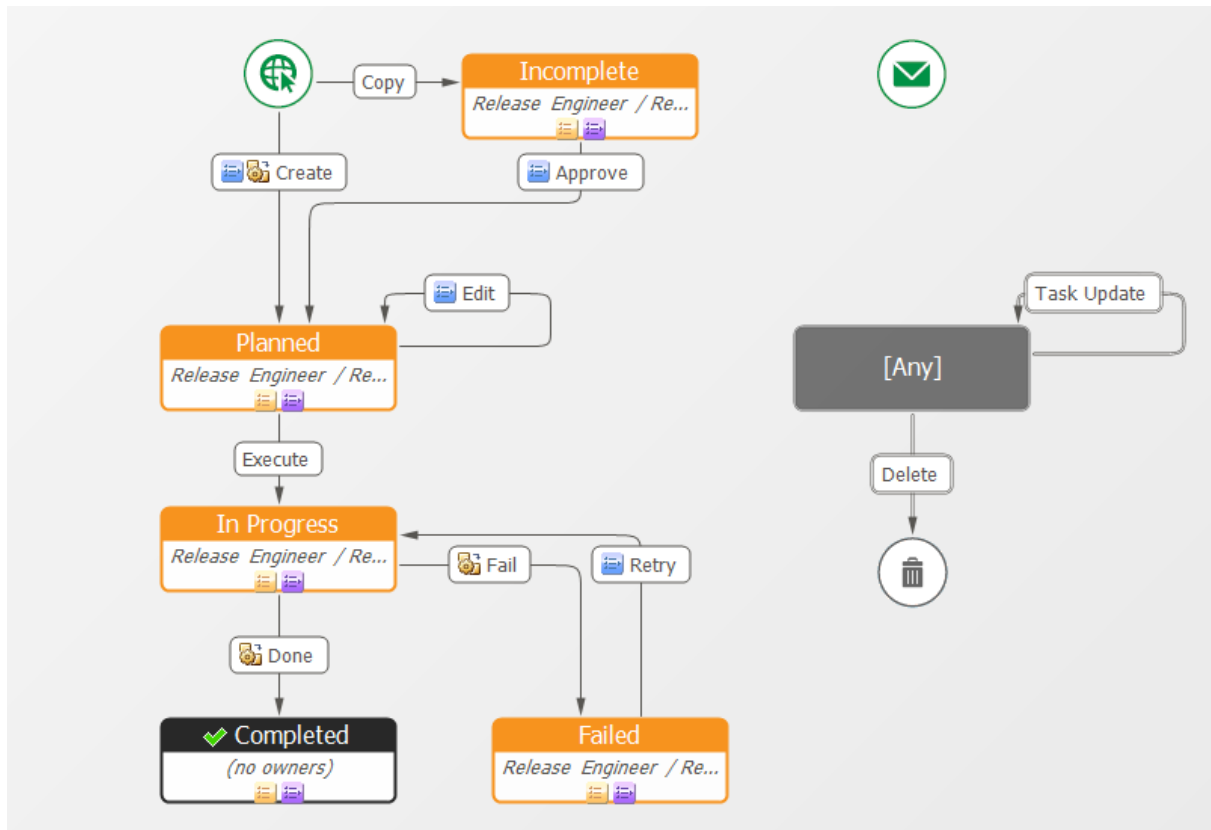
Automation Deployment Task Workflow

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



Manual Deployment Task Workflow

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

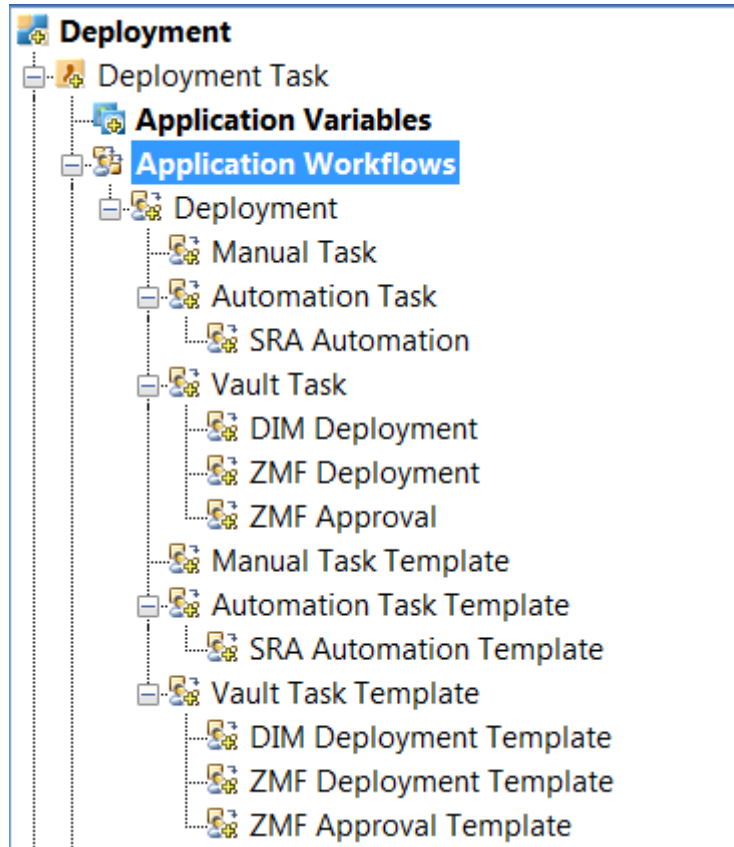


Vault Deployment Task Workflows

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

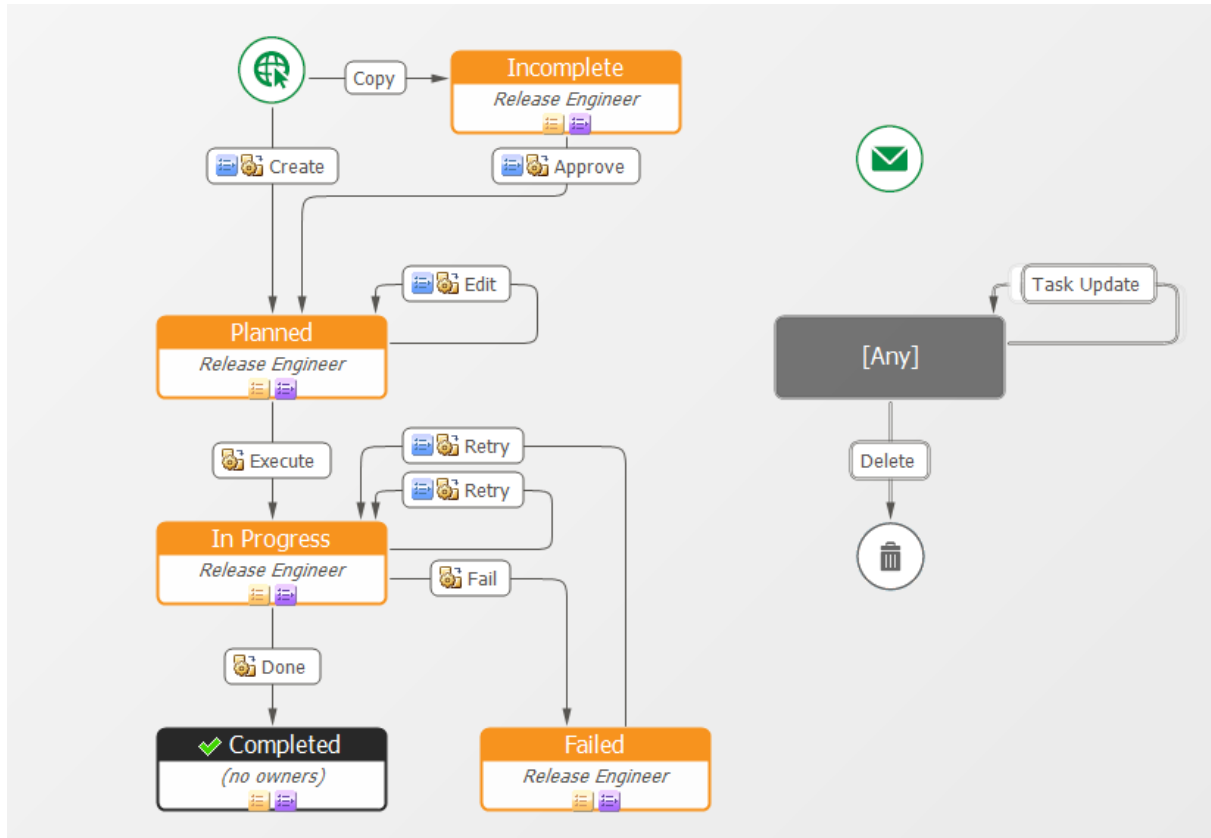
Deployment Task Hierarchy

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



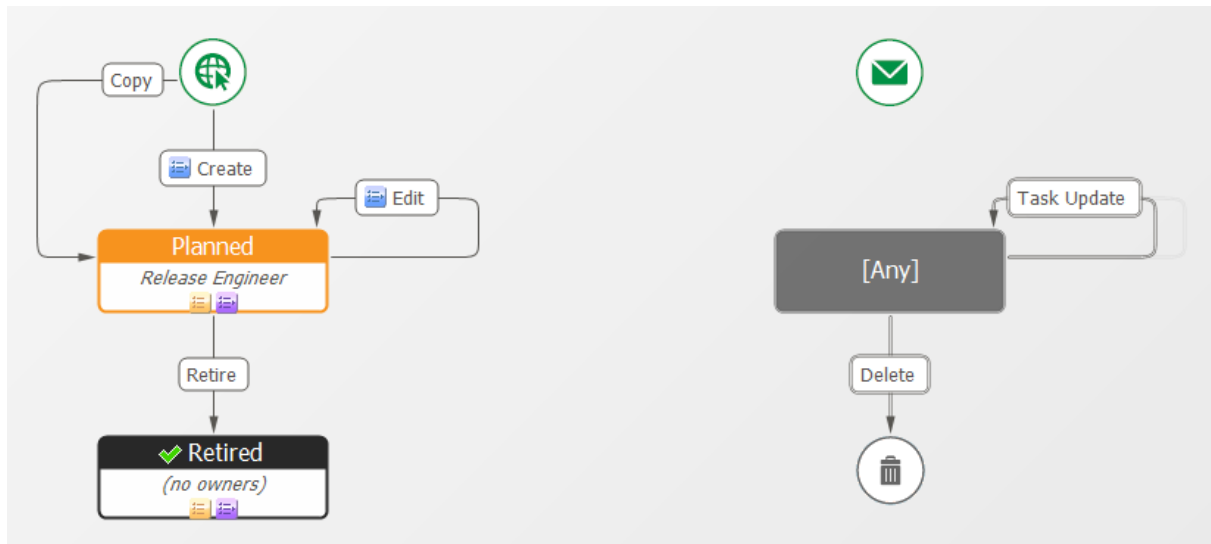
Main Vault Task Workflow

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.



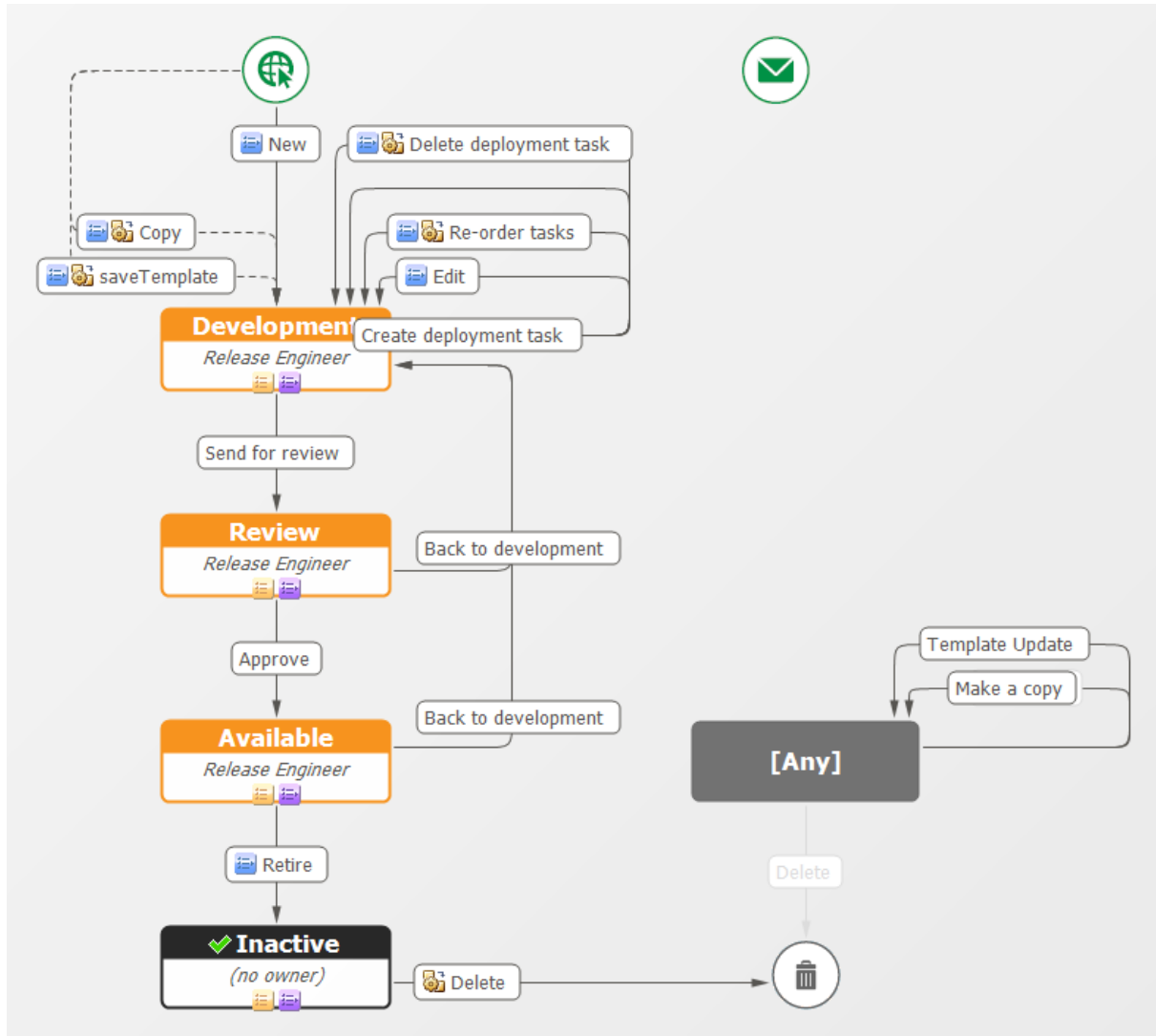
Main Vault Task Template Workflow

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.



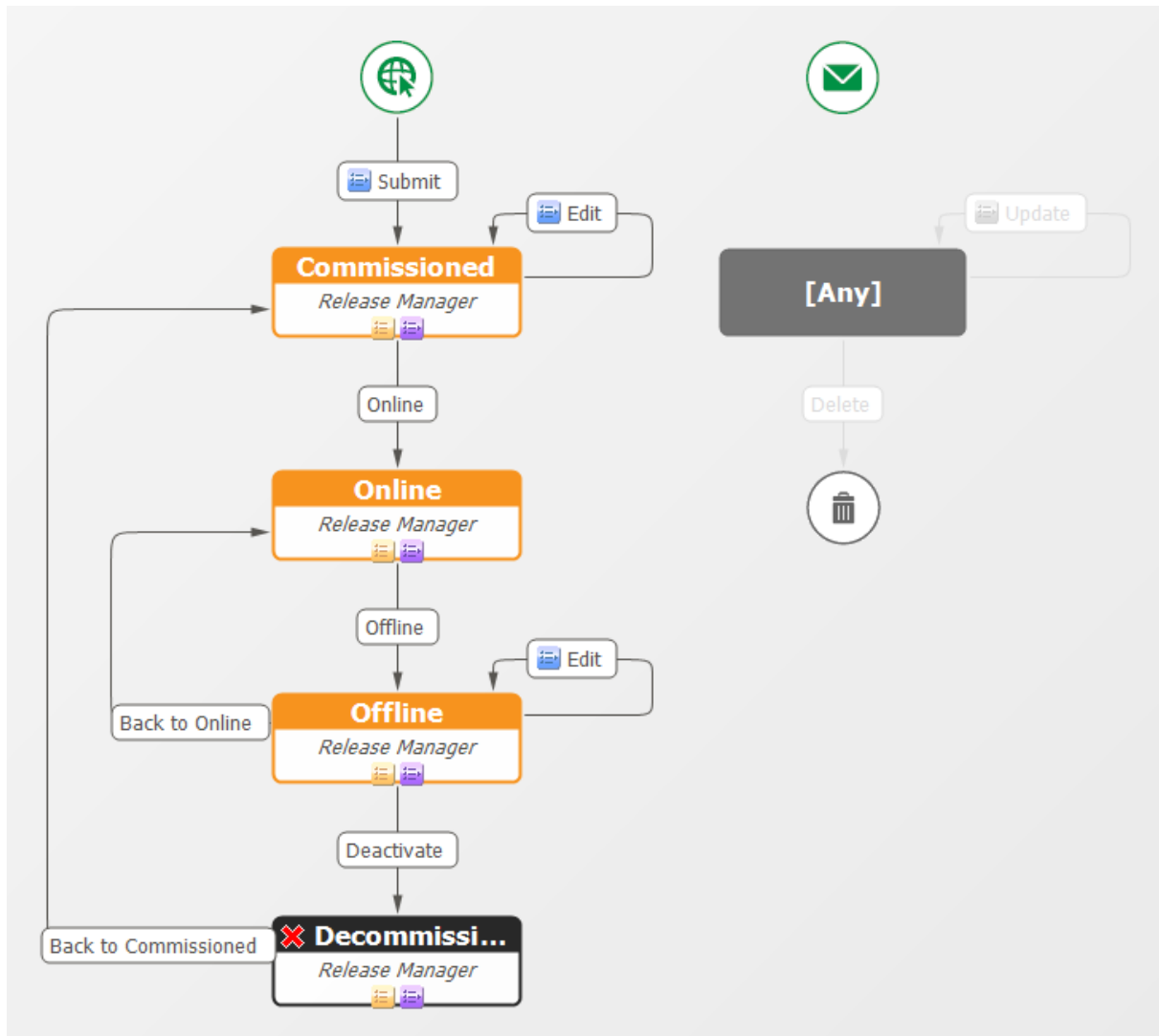
Deployment Process Template Workflow

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.



Vault Request Workflow

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



The diagram illustrates the lifecycle of a resource in a system. It starts with a resource being either 'Available' (green box) or 'Inactive' (black box). Both states are labeled '(no owner)' and have a small icon of a resource. Transitions are as follows:

- Available State:**
 - Can be reached from a 'Save from request' box or a 'Submit' box (with a green globe icon).
 - Can transition to 'Inactive' via a 'Retire' box.
 - Can be reached from 'Inactive' via an 'Activate' box.
- Inactive State:**
 - Can transition to 'Available' via an 'Activate' box.
 - Can be deleted via a 'Delete' box, leading to a trash can icon.
- [Any] State:**
 - Can be reached from an 'Update' box (with a green envelope icon).
 - Can be deleted via a 'Delete' box, leading to a trash can icon.

At the bottom, there is a faded box labeled '(no owner)' with a small icon of a resource.

Development Manager Workflows

Development Manager workflows are documented in the *Serena Development Manager Getting Started Guide*.

Configuration File Reference

Serena ALM uses configuration files to specify information for the Web services to use. These configuration files contain client connection information, filters, and other information necessary to execute a comprehensive set of release management operations.

This section lists and explains the use of the configuration files.

[Configuration Files on the Serena ALM Server \[page 239\]](#)

[Configuration Files on the Dimensions CM Server \[page 242\]](#)

[Configuration Files on the Release Automation \(Nolio\) Server \[page 242\]](#)

Configuration Files on the Serena ALM Server

Configuration files are used by Serena ALM to set implementation-specific details. Most of the configuration files reside on the Serena ALM server.

Files in the Classes Folder

Most of the configuration files on the Serena ALM server are located under the alm Web service folder in the WEB-INF\classes folder. For example:

```
..\Program Files\Serena\common\tomcat\6.0\webapps\alm\WEB-INF\classes
```

The configuration file names and purpose are given in the following table.

File Name	Purpose
alf-client-connection.properties	Use this file to specify credentials for the ALF server to be used for notifications sent from the Serena Release Automation server. When the automation deployment task is completed, Serena Release Manager sends an ALF event to SBM server. This requires the credentials to access the ALF server, because Serena Release Automation isn't implemented to use SSO. See ALF Client Connection Information [page 54] .
bcr.properties	Provides provider definitions for Business Change Requests (BCRs). See Configuring Access to Business Change Requests [page 124] .
bcr-connection.properties	Provides connection information for Business Change Requests (BCRs). See BCR Provider Connection Information [page 55] .
commons-logging.properties	Apache log file installed with this software.
dm_qlarius.properties	Provides filtering information for Dimensions CM objects accessed by Release Manager. See Configuring Access to Development Change Requests [page 125] and Configuring Access to Deployment Units [page 128] .
dm-client-connection.properties	Provides connection information for Dimensions CM. See Dimensions CM Client Connection Information [page 56] .

dvmzmf_packages.properties	Provides filtering information for accessing ChangeMan ZMF information from Development Manager. See Development Manager Dev Task Provider Details [page 132] .
itsm.properties	Provides provider definitions for ITSM (for RFCs). See Configuring Access to Requests for Change [page 122] .
itsm-connection.properties	Provides connection information for ITSM (for RFCs). See RFC Provider Connection Information [page 56] .
LICENSE.txt	Apache licensing information for use with this software.
log4j.properties	Tells the location and behavior for the primary message log file for Serena ALM, alm.log. See Information from Log Files [page 187] .
logback.xml	This file is used for logging confirmation. Do not change this file.
messages.properties	This file is planned for future use to define strings used in the Serena ALM Configurator UI.
nolio-client.properties	Provides client-specific information for Serena Release Automation, powered by Nolio. See Configuring Release Automation (Nolio) Communication in Release Manager [page 116] .
nolio-client-connection.properties	Provides connection information for Serena Release Automation, powered by Nolio. See Serena Release Automation (Nolio) Client Connection Information [page 58] .
nolio-client-queries.properties	Provides filtering information for Serena Release Automation, powered by Nolio. See Configuring Release Automation (Nolio) Communication in Release Manager [page 116] .
NOTICE.txt	Notice that Apache software is used and distributed with this software.
providers.properties	Specifies the providers to be used for this implementation of Serena ALM. See Telling Release Manager Which Providers to Use [page 121] .
ra-client.properties	Provides client-specific information for Serena Release Automation. See Configuring Release Automation Communication in Release Manager [page 111] .
ra-client-connection.properties	Provides connection information for Serena Release Automation. See Serena Release Automation Client Connection Information [page 57] .

rlm.properties	<p>This file is used for global alm.war settings. Most settings are system settings for Web services. The only setting that you should change is:</p> <pre>rlm.config.service.authdUsers=<list of user IDs delimited by commas></pre> <p>This defines the list of users who can access the Serena ALM Configurator through SSO. See Configuring Access to the Configurator [page 53].</p>
rlm-sbm.properties	<p>Sets values for the Serena ALM Web services used for Serena Release Manager. See Setting Maximum Associations for Release Manager Objects [page 184].</p> <p>CAUTION! The only values that should be changed in this file are the maximum object association limits for release trains, application releases, and release packages.</p>
rlmzmf_packages.properties	Provides filtering information for accessing ChangeMan ZMF information from Release Manager. See Configuring Access to Deployment Units [page 128] .
sbm.properties	Sets values for the Serena ALM Web services.
sbm_incidents.properties	Provides provider definitions for SBM Incidents, typically associated with BCRs. See Designating the Details for Each BCR Provider [page 124] .
sbm_issues.properties	Provides provider definitions for SBM Issues, typically associated with DCRs. See Designating the Details for Each DCR Provider [page 125] .
sbm-client-connection.properties	Provides connection information for SBM. See SBM Application Connection Information [page 54] .
zmf-client.properties	Provides client-specific information for ChangeMan ZMF. See Configuring ZMF Communication in Release Manager [page 105] .
zmf-client-connection.properties	Provides connection information for ChangeMan ZMF. See ZMF Client Connection Information [page 58] .

Files in Other Folders

Some of the configuration files on the Serena ALM server are located in other folders. The configuration file names, location, and purpose are given in the following table.

File Name	Location and Purpose
alfzmf_resource.properties	<p>..\Program Files\Serena\common\tomcat\6.0\webapps\almzmfalf\WEB-INF\conf</p> <p>ALF event manager information for ChangeMan ZMF. See Specifying ALF Event Manager Connection Information for ZMF [page 105].</p>

Configuration Files on the Dimensions CM Server

Configuration files on the Dimensions CM server are used to configure return communication to Serena ALM from Dimensions CM. The configuration file names, location, and purpose are given in the following table.

File Name	Location and Purpose
dm.cfg	..\Program Files\Dimensions 12.2\CM You must update this file on the Dimensions CM server with Dimensions CM ALF event configuration information Serena ALM needs. See Specifying Dimensions CM ALF Event Configuration Information [page 80] .
ALF_EVENTS_CONFIG.XML	..\Program Files..\Dimensions\12.2\CM\dfs You must update this file on the Dimensions CM server to specify selection criteria for Dimensions CM information Serena ALM accesses. See Specifying Selection Criteria for Dimensions CM Events and Objects [page 81]

Configuration Files on the Release Automation (Nolio) Server

Configuration files on the Serena Release Automation, powered by Nolio, server are used to configure return communication to Serena ALM from Serena Release Automation, powered by Nolio. The configuration file names, location, and purpose are given in the following table.

File Name	Location and Purpose
rest.integration.properties	..\Program Files\Serena\Serena Release Automation\conf Tells Serena Release Automation, powered by Nolio, what server to notify when an event occurs. See Specifying the Release Automation (Nolio) Server to Notify [page 114] .