



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: 4.5

Publication date: 2013-05-13

Table of Contents

- Welcome to Serena ALM 1
 - Accessing the Documentation 8
 - What is Serena Release Manager? 9
 - What is Serena Development Manager?..... 9
- Installation Prerequisites and Planning 11
 - Serena ALM Architecture 11
 - Installation Prerequisites and Order 12
 - Database Requirements 13
 - Software Compatibility Requirements..... 14
 - System Requirements..... 14
 - Installation Planning for SBM Connector for HP Quality Center / ALM Integration 15
 - Installing and Configuring Serena Requirements Manager 15
 - Installing and Configuring Serena Dashboard 15
- Serena ALM Installation 16
 - Installation Overview 16
 - Upgrade Overview 16
 - Installation and Configuration Checklist..... 18
 - Running the Installer 20
 - Upgrading the Agile Database 25
 - Importing and Promoting Serena ALM 25
 - Importing the Serena ALM Solutions 25
 - Creating an Environment for Serena ALM 26
 - Promoting the Serena ALM Snapshots 28
 - Configuring SSO in RESTgrid Widgets 31
 - Updating the Connection Details in the Workflows 32
 - Publishing and Deploying the Process Apps..... 33
- Serena ALM Configuration 35
 - Applying Licenses..... 35
 - Configuring the Admin User for Release Manager 35
 - Creating an Administrative User 36
 - Configuring the Administrative User Privileges 36
 - Enabling Serena Release Manager Project Roles 37
 - Creating the Service User for Development Manager 38
 - Using the Serena ALM Configurator..... 40
 - Configuring Access to the Configurator..... 40
 - Invoking the Serena ALM Configurator 40
 - Entering Information in the Serena ALM Configurator 40

ALF Client Connection Information.....	41
SBM Application Connection Information	41
BCR Provider Connection Information	42
RFC Provider Connection Information	43
Dimensions CM Client Connection Information	44
Serena Release Automation Client Connection Information	44
Serena Release Automation (Nolio) Client Connection Information	45
ZMF Client Connection Information	46
Accessing the Standard SBM User Interface	47
Managing Serena ALM User Interface Reports	47
Changing Existing Reports	48
Creating New Reports	48
Naming Convention and Reference Rules.....	48
List of Release Manager Reports.....	50
List of Development Manager Reports	53
Privileges Required for UI Reports	58
Release Manager Configuration	58
Serena Release Manager Runtime Communication	58
Adding Your Application Names in Serena Release Manager	59
Adding Your Server Names in Serena Release Manager	59
Enabling Scheduling of Deployment Tasks	60
Managing Release Manager Users	61
Managing Release Manager Notifications	63
Development Manager Configuration	67
Creating and Assigning the Development Manager Users	67
Agile Configuration	72
General Agile Configuration	72
Agile Connection Information	73
Enabling the Agile Functionality on IIS 7.5	74
Configuring Agile to Use Different Process Apps.....	74
Configuring Epic and Story Properties	75
Configuring Task Properties	76
Configuring the Project Context Properties	78
Configuring the Item Hierarchy Properties	78
Dimensions CM for Serena ALM Configuration	80
Dimensions CM Communication Configuration Overview	80
Configuring Communication on the Dimensions CM Server	81
Configuring Dimensions CM for ALF Events	81
Specifying Selection Criteria for Dimensions CM Events and Objects	83
Configuring Objects in Dimensions CM	84
Configuring the Dimensions CM Global Stage Lifecycle	84

Managing Dimensions CM Users	85
Configuring Dimensions CM Roles and Privileges	86
Assign Build Engineer Role	86
Assign Development Engineer Role as Delegation Candidate	87
Assign Development Engineer Role to TASK Lifecycle Transitions	88
Setting Privileges for the Service User	90
Configuring Dimensions CM Projects and Streams	92
Configuring Dimensions CM for Products and Components	92
ChangeMan ZMF for Serena ALM Configuration	94
ZMF Communication Configuration Overview	94
Configuring ZMF Communication on the Mainframe	94
Configuring the Notification URL.....	95
Configuring the SERNET HTTP Server	95
SERSERV Prerequisites	96
Installing SERSERV	98
Verifying the Installation of SERSERV	102
SERSERV Runtime Considerations	102
Configuring a ZMF Proxy User ID	104
Configuring TSO User IDs and Permissions	105
Configuring ZMF Approvers	105
Configuring Objects in ChangeMan ZMF	105
Configuring ZMF Communication in Serena ALM	106
Specifying ALF Event Manager Connection Information for ZMF	106
Specifying Client-Specific Information for ChangeMan ZMF	107
Configuring the Release Manager ZMF Web Page Widgets	108
Configuring the Development Manager ZMF Web Page Widgets	109
Release Automation for Serena ALM Configuration	110
Release Automation Communication Configuration Overview	110
Release Automation Architecture	110
Release Automation (Nolio) Architecture	111
Configuring Communication with Release Automation	111
Configuring Communication on the Release Automation Server	111
Configuring Release Automation Communication in Release Manager	112
Configuring Objects in Serena Release Automation	113
Configuring Communication with Release Automation (Nolio)	115
Configuring Communication on the Release Automation (Nolio) Server	115
Configuring Release Automation (Nolio) Communication in Release Manager	117
Configuring Objects in Release Automation (Nolio)	119
Provider Configuration	122
Release Manager Provider Overview	122
Telling Release Manager Which Providers to Use	122

Configuring Access to Requests for Change.....	123
Designating the Details for Each RFC Provider	124
Configuring Access to Business Change Requests	125
Designating the Details for Each BCR Provider	125
Configuring Access to Development Change Requests	126
Designating the Details for Each DCR Provider	126
Configuring Access to Deployment Units	129
Release Manager DU Provider Details	130
Configuring Objects in Serena Business Manager	130
Configuring Objects in Serena Service Manager.....	131
Development Manager Provider Overview	132
Telling Development Manager Which Providers to Use.....	132
Development Manager Dev Task Provider Details	133
Vault Request Configuration	139
Vault Request Purpose	139
Vault Request Architecture	140
The Vault in a Vault Request	140
Definitive Media Library (DML)	140
Vault Template Overview	140
Configuring Integrations with Orchestrated Ops Solutions	141
SSM and Release Manager Integration	141
Creating the Reports Used for the SSM Integration	142
Changing the Integration in SSM	143
Configuring the SSM Feed for the Release Train Calendar	144
Changing the Instance of SSM that Release Manager Uses.....	144
Demand Manager and Development Manager Integration	144
Troubleshooting Serena ALM	146
Troubleshooting Overview	146
Information from the User Interface	146
Error Messages	146
Activity Log	146
Activity Page	147
History	147
Information from the ALM Configurator	147
Information from Log Files	147
Execution Log File	147
Installer Log Files	147
Symptoms and Solutions	148
ALM Installer Errors	148
Snapshot Promotion Errors	149
Cannot log into Serena ALM.....	150

Cannot Log into Serena ALM Configurator	150
User Interface Display Issues	151
Create Release Fails with a Check Uniqueness Error	152
Unable to update the landing page layout	152
Matches Not Found for Selections	152
No Change Package Data Displayed in ZMF Deployment Tasks	153
Release Package Deployment Fails	153
Slow Response Time	154
Serena ALM Workflow Reference	156
Release Manager Workflows	156
Workflow Relationships	156
Release Train Workflow	158
Application Release Workflow	159
Release Package Workflow	160
Start and Development States	160
Integration State	161
Staging and Production States	162
Exceptions State	163
Deployment Task Workflows	163
Automation Deployment Task Workflow	164
Manual Deployment Task Workflow	166
Vault Deployment Task Workflows	167
Deployment Process Template Workflow	168
Environment Workflow	169
Vault Request Workflow	169
Vault Template Workflow	170
Development Manager Workflows	170
Configuration File Reference	171
Configuration Files on the Serena ALM Server	171
Files in the ALM Classes Folder	171
Files in Other Folders	173
Configuration Files on the Agile Server	174
Configuration Files on the Dimensions CM Server	174
Configuration Files on the Release Automation (Nolio) Server	175

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 \[page 8\]](#)

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

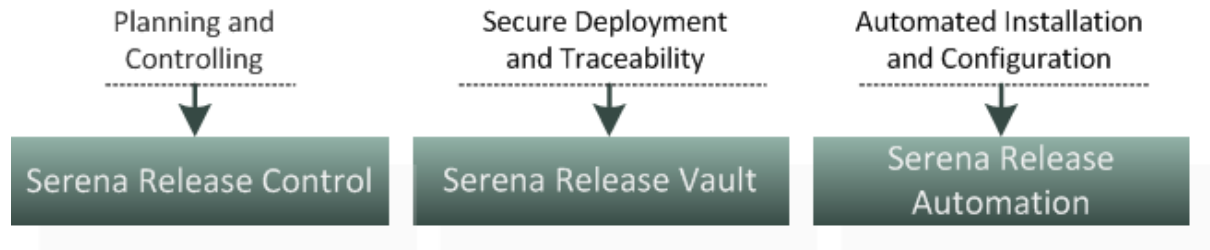
product. To view and download readme files and PDFs for all supported releases, select **Demos All Doc** in the online Help menu.

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 track 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 ALM Architecture \[page 11\]](#)

[Installation Prerequisites and Order \[page 12\]](#)

[Database Requirements \[page 13\]](#)

[Software Compatibility Requirements \[page 14\]](#)

[System Requirements \[page 14\]](#)

[Installation Planning for SBM Connector for HP Quality Center / ALM Integration \[page 15\]](#)

[Installing and Configuring Serena Requirements Manager \[page 15\]](#)

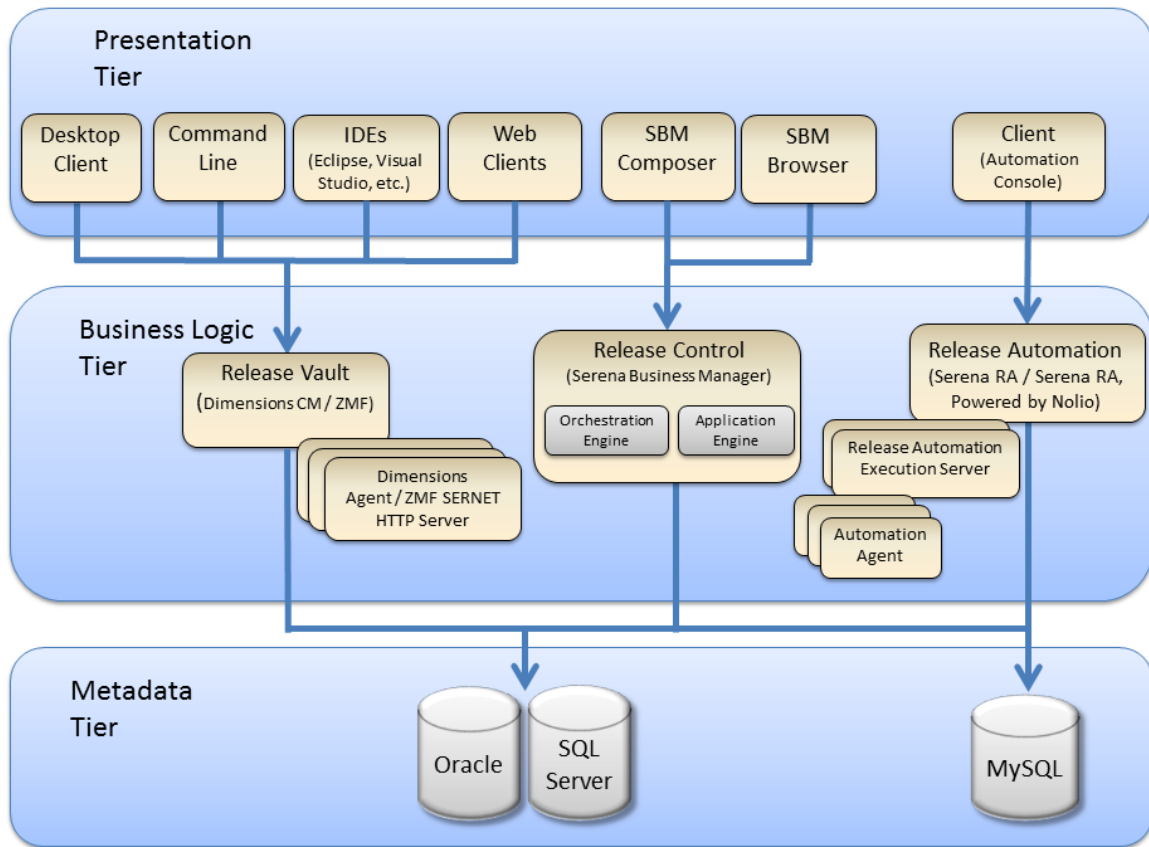
[Installing and Configuring the Serena Dashboard \[page 15\]](#)

Serena ALM 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*.



Installation Prerequisites and Order

Before you install the Serena ALM suite of products, make sure you have completed the prerequisites as follows. The installation and configuration of the required products 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 products that you plan to use with Serena ALM, you do not need to install them again. 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 Release Manager or Serena Development Manager.

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. **Database:** If you plan to use 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. See *Serena Dimensions CM and RM Installing the Serena-Supplied Runtime RDBMS*.
3. **Serena Business Manager** must be installed and configured before you can install Serena Development Control or Serena Release Manager.
 - Please familiarize yourself with the SBM Readme before proceeding.
 - For either new or existing SBM installations, ensure SSO is installed and enabled for SBM. Be aware that for SBM 10.1.2 and above, the SSO gatekeeper information has changed. This is documented in the SBM Readme.
 - If you are installing a new SBM installation, 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.
 - Please familiarize yourself with the Dimensions CM Readme before proceeding.
 - For either new or existing Dimensions CM installations, ensure SSO is enabled for Dimensions CM and point Dimensions CM to the SSO server you installed for SBM.
 - If you are installing a new Dimensions CM installation, see the "Fresh Installations Checklist" section of the *Serena Dimensions CM Installation Guide* for your operating system.

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.

IMPORTANT! This release of Serena ALM ships with a version of the common tools installer that 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.

5. **Connector for HP Quality Center (Optional)**

If you plan to install and use 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*

6. **Serena Release Automation (Optional)**

- If you plan to install and use Serena Release Automation with Release Manager, you will need to install the latest version to take advantage of all available integrating features.
- If you plan to install and use Serena Release Automation, Powered by Nolio, you will need to install version 4.0.1 or earlier versions supported by your version of Serena Release Manager.

7. **Serena ChangeMan ZMF (Optional)**

- If you are using Changeman ZMF as a Configuration Management provider, Development Manager requires ChangeMan ZMF.
- If you are using ChangeMan ZMF for your vault, Release Manager requires ChangeMan ZMF.

8. **Serena ALM** must be installed and configured after all the required prerequisites are met.

- The Serena ALM installer installs Serena Development Manager, Serena Release Manager, and Serena Requirements Manager components.
- **Common Supporting Files:** The installer for Serena ALM automatically installs additional supporting software, such as a common Tomcat Web server, if it does not detect that it already exists. If you install Serena ALM on the same machine as Dimensions CM, Serena ALM will use the Serena Common Tomcat already installed with Dimensions CM.

Database Requirements

Before beginning the installation, you must have database systems installed and ready for each of the foundational products as follows:

- If you are using a shared database, install the database before installing the rest of the products so that you can point each to the shared database.
- If you are installing SQL Server 2008 in a multi-server environment, the database processing requires the Microsoft SQL Server 2008 Command Line Utilities. This has to be located on the machine on which you are running the installer, not the database machine. This component can be found at the following link:

<http://www.microsoft.com/en-us/download/details.aspx?id=27596> search for ENU\x64\sqlncli.msi

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 14] .

Server Requirements

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

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

For prerequisites for the Quality Center integration, see the *Serena Development Manager DVM for HP Quality Connector 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 *Serena Development Manager DVM for HP Quality Connector 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*.

Installing and Configuring Serena Dashboard

The Serena Dashboard is a separately licensed option. For prerequisites, 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 \[page 16\]](#)

[Upgrade Overview \[page 16\]](#)

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

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

[Importing and Promoting Serena ALM \[page 25\]](#)

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

Upgrade Overview

If you are upgrading from Serena ALM version 4.0.1 to version 4.5 rather than doing a new installation, document any customization of your implementation and back up your system, upgrade any supporting systems such as SBM or Dimensions CM, and then run the installer.

Non-Typical Upgrades

If you are upgrading from Serena ALM versions earlier than 4.0.1 to version 4.5, or are upgrading from a system in which Requirements Manager, products and components, or HP Quality Center are also implemented, see the documentation for those solutions and the Serena Support website Knowledgebase. You must be logged on the Support website to view Knowledgebase solutions.

Typical Upgrades

To perform a typical upgrade of Serena ALM from version 4.0.1 to 4.5:

1. Document custom changes.

If you have a customized version of Serena ALM, 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. Upgrade any supporting systems such as SBM or Dimensions CM according to the [Software Compatibility Requirements \[page 14\]](#).

4. Install and configure Serena ALM as you would for a new installation according to the [Installation and Configuration Checklist \[page 18\]](#).

IMPORTANT! For upgrades:

- You **MUST** promote the Dev Change Requests and Global Process App process apps with the following settings:
 - Select **Merge Conflict** for each of the process app snapshots during promotion.
 - For *DVM - Development Change Requests*, for **Projects**, select **None**. The projects already exist, so they cannot be imported again.
 - For *Global Process App*, for Users and Groups, select **None**. The users and groups already exist, so they cannot be imported again.
- If you are using the Agile option for Development Manager, you must upgrade your Agile database using the provided SQL scripts. You must do this after running the installer but before promoting the solutions.

5. If you are using Development Manager, adjust the configuration of your groups, users, and roles as needed according to [Creating and Assigning the Development Manager Users \[page 67\]](#).

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 Running the Installer [page 20]</p>
2: Upgrade Your Agile Database (Upgrades Only)	<p>If you are using the Agile option for Development Manager and are upgrading from the previous version of Development Manager, you must upgrade your Agile database using the provided SQL scripts. You must do this after running the installer but before promoting the solutions.</p> <p>See Upgrading the Agile Database [page 25].</p>
3: Process Apps	<p>Import the Serena ALM SBM solutions and promote the snapshots of the process apps.</p> <ol style="list-style-type: none"> 1. Import the Serena ALM solutions you plan to use. These contain snapshots of the process apps and other supporting data. 2. Create a new application server environment for the process apps if needed. 3. Promote each of the snapshots, creating and selecting endpoints as needed. <ul style="list-style-type: none"> • Promote the <i>Global Process Apps</i> snapshot first if you want to use the sample data. • Promote the <i>DVM - Development Projects</i> snapshot next for Development Manager to automatically created necessary endpoints. • Select Security Token authentication for the Release Manager endpoints. 4. Select Merge Conflicts for the Development Manager snapshots to ensure the user roles are included. FOR UPGRADES ONLY, select none for users and groups for <i>Global Process Apps</i> to avoid importing that information again. 5. Configure the process apps in SBM for Serena Release Manager and/or Serena Development Manager. <ul style="list-style-type: none"> • If needed, select the SSO Authentication option in forms with RESTgrid widgets. • Update the connection details in the Development Manager workflows if you are not using the default service user and localhost. • Publish and deploy the process apps to upgrade the structure of the RESTGrid widgets and apply any other changes. <p>See Importing and Promoting Serena ALM [page 25].</p>

4: 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 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, groups, roles, reports, and notifications ◦ Release Manager applications and environments (servers) (specific to your organization; add to the respective auxiliary tables) ◦ Development Manager Agile properties <p>See Serena ALM Configuration [page 35].</p>
5: 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 80].</p>
6: ChangeMan 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 94].</p>
7: 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 110].</p>

8: 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: `r1mzmf_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 \[page 122\]](#) 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 \[page 151\]](#)

Running the Installer

If you are installing in a multi-server environment, you must run the Serena ALM installer on the machine where the SBM Application Engine resides. This can be run either locally or remotely. This installs the Serena Common Tomcat, the Web services, and the Serena ALM SBM components on the Application Engine machine.

To install Serena ALM:

1. Download the full installation package from the Serena Support Web site, My Downloads page. 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.

NOTE If you are installing in a multi-server environment, you must run the Serena ALM installer on the machine where the SBM Application Engine resides.

3. 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.

4. Click **Next**.

The **License Agreement** page appears.

5. Confirm and click **Next**.

The **Destination Folder** page appears.

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

7. Click **Next**.

The **Product selection** page appears.

8. To install the *Agile features* of Serena Development Control, select **Development Control**.

The shell and database schema for the Agile features of Development Manager are installed.

NOTES

- Selecting the Development Manager option does not require you to use the Agile option with Development Manager, but if you do not select it, the components required for Agile are not installed. If you are undecided, select it.
- You cannot deselect **Release Control**. Whether or not you have access to Release Control will be determined by which solutions you import and which snapshots you promote.

9. 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.

10. 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 and 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 the user experience.
- **ALM Process Application**: SBM solution files. The solution files must be imported into an SBM application repository.

11. 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 Beginning with Serena ALM version 4.0.1, the installation includes a version of the Serena Common Tomcat Web server that 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 or later.

The Serena ALM 4.5 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, and then installs the new common tools.

12. 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**.

13. If you selected the **Agile Configuration** option, the **Connection details** page appears.

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

Database Server Type	Option	Notes
----------------------	--------	-------

Oracle	Host	Host name where your database resides. Example sbm_dbhost
	Oracle SID	Oracle system ID (SID) Example dim12
	Port	Port on the host where your database resides. Example 1521
	Oracle Home	Home directory for your Oracle files. Example c:\oracle\ora11g
	SBM Database user	SBM database user name Example sample
	Password	Password for the SBM database user Example sample
SQL Server	Host	Host name where your database resides. Example sbm_dbhost
	Database Instance	Name of the database instance. Example sample
	Port	Port on the host where your database resides. Example 1521
	Database Name	SBM database name Example sample
	SBM Database user	SBM database user name Example sample
	Password	Password for the SBM database user Example sample

- b. Click **Test Connection**.

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

- c. Click **Next**.

The **Service Execution Login Credentials** page appears.

14. If you plan to use the Agile option for Development Manager, enter the application and administrative service user credentials and click **Next**.

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

15. 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. 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.

16. 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 (contains the shell zip file)

 \solution(contains the solution files)

 \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 SBM solution files to the solutions folder for the Serena Common JBoss.
- Copies Agile database upgrade files to the solutions/Agile folder.

17. 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.

Upgrading the Agile Database

To manually upgrade your Agile database:

1. Navigate to the directory where your Agile database upgrades scripts were copied during the installation. For example: `..\Program Files\Serena\Solutions\Agile`
2. Execute `update_oracle.sql` or `update_mssql.sql` against the database you specified when running the installer, depending on whether you are using an Oracle or Microsoft SQL Server database system.
3. Continue with the installation.

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 ALM suite is packaged as a set of SBM solutions. Included in the solutions are the following:

- Process apps for each solution
- Runtime configuration information
- Reports

- Notifications
- Auxiliary table information

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

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

1. Select **Start | All Programs | Serena | Serena Business Manager | SBM Application Repository**.

2. Enter your login details.

3. In the navigation pane, click **Solutions**.

4. The Serena ALM solution packs are listed as available. For example:

- serena_requirements_manager_4_5_0_0_bbb
- serena_development_manager_4_5_0_0_bbb
- serena_release_manager_4_5_0_0_bbb
- dvm___products_and_components_4_5_0_0_bbb

where bbb is build number

5. For each solution that you want to implement, 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 process app snapshots should be listed.

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 [Promoting the Serena ALM Snapshots \[page 28\]](#).

Create the environment and supporting target servers as follows:

- [Creating the Environment \[page 27\]](#)
- [Specifying the BPEL Server for SBM Orchestrations \[page 27\]](#)
- [Specifying the SBM System Event Manager \[page 27\]](#)

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

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

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

Specifying the BPEL Server for SBM Orchestrations

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

To set up the target server for SBM orchestrations:

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

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

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

Port 8085 is the default, unless you specified a different port in the 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, overwrite 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.

Promoting the Serena ALM Snapshots

You must promote each of the snapshots required for Serena ALM solutions you plan to use, Release Manager, Development Manager, or Requirements Manager, or a combination thereof.

When promoting the snapshots, make sure to create and select the endpoints as needed. See additional notes in the following procedure for solution-specific and upgrade selections.

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. Examine the following table to see which snapshots are required for your implementation:

Snapshots Required for Implementations of DVM and/or RLM	
Snapshot	Notes
Global Process App	Promote this first if you want to use the sample user data; this creates the sample users, groups, and roles
ALM - Common	<ul style="list-style-type: none"> • Promote this only once, although it is listed multiple times if you import more than one solution • Promote this before other snapshots that use the DVMUtilService, RLMUtilService, and ReleaseRequestService endpoints; otherwise, these endpoints must be manually created as part of the promotion procedure
DVM - Development Projects	

Snapshots Required for Implementations of DVM and/or RLM	
Snapshot	Notes
DVM - Development Change Requests	This has a predefined project that is created as an example; choose the option appropriate to your situation according to the promotion procedure
DVM - Development Tasks	
DVM - Development Packages	
DVM - Provider Connector	
DVM - Products and Components	Promote only if you plan to use the Products and Components option with DVM
DVM - HP Quality Center Connector	Promote only if you plan to use the HP Quality Center Connector with DVM
RLM - Release Train	
RLM - Application Release	
RLM - Release Package	
RLM - Deployment	
RLM - Environment	
RLM - Release Template	
RLM - Vault Request	
<p>LEGEND:</p> <p>ALM = Serena ALM (Application Lifecycle Manager)</p> <p>DVM = Development Manager</p> <p>RLM = Release Manager</p>	

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.

- For all Development Manager snapshots, select **Merge Conflicts**. This enables the user information to be included in the process apps. This is not necessary for the Release Manager snapshots because you will configure the users manually.
- For the *DVM - Development Change Requests* 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 the *Global Process App*, if you do not want to import the sample groups and users into your environment, select:

Groups: None

Users: None

NOTE For upgrades, do NOT import the sample groups and users.

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

Source	Destination Endpoint
ReleaseAutomationService	http://almhost:8080/alm/services/ReleaseAutomationService
DeploymentAutomationService	http://almhost:8080/alm/services/DeploymentAutomationService
VaultService	http://almhost:8080/alm/services/VaultService
DVMUtilService	http://almhost:8080/alm/services/DVMUtilService

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

Configuring SSO in RESTgrid Widgets

You must select the **Use SSO authentication** check box for each RESTgrid widget that still uses the SOAP Web services with HTTP binding. This enables the security token to be included in the URL.

RESTgrid Widgets to Update

If you are not using Serena Release Automation or Serena Release Automation, Powered by Nolio, you may skip this section.

The only form controls in which you must update the check box are shown in the following table.

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

Updating the Connection Details in the Workflows

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

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. You may continue to [Publishing and Deploying the Process Apps \[page 33\]](#).

The default settings are:

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

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 workflows in the **DVM - Provider Connector** process app that may need updating are as follows:

- CM_BuildComplete

The workflows in the **DVM – HP Quality Center Connector** process app that may need updating are as follows:

- Create_DVM_Issue
 - Update_DVM_Issue
1. Open the **DVM - Provider Connector** process app and update the connection information as follows:
 - a. Under **Orchestration Workflows**, select *CM_BuildComplete*
 - b. In the Property Editor, select **Data Mapping**.
 - c. In *Working data*, under *SBMAuth*, change the *Default value* for *userId*, *password*, and *hostname* to the actual server name (if not localhost) and user credentials you will be using.
 2. Open the **DVM – HP Quality Center Connector** process app and update the connection information as follows:
 - a. Under **Orchestration Workflows**, select *Create_DVM_Issue*.
 - b. In the Property Editor, in **Data Mapping**, update the Default values for *SBM_User*, and *SBM_Password* to the values for the connection user.
 - c. Repeat this for the *Update DVM_Issue* orchestration workflow.

Publish and deploy each process app when you have completed the updates.

Publishing and Deploying the Process Apps

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

If you have not made any changes to the process apps since promoting the snapshots, you may skip this procedure and continue with [Serena ALM Configuration \[page 35\]](#).

When deploying the Serena ALM process apps you need to do so in the following order:

1. Global Process App
2. ALM Common
3. RQM - Requirements Approval Management (if you are using Requirements Manager)
4. DVM - Projects (if you are using Development Manager)
5. DVM - Development Change Requests (if you are using Development Manager)
6. DVM - Development Tasks (if you are using Development Manager)
7. DVM - Development Packages (if you are using Development Manager)
8. DVM - Provider Connector (if you are using Development Manager)
9. DVM - Products and Components (if you are using Development Manager)
10. DVM - HP Quality Center Connector (if you are using HP Quality Center)

11. RQM - Dimensions RM Connector (if you are using Requirements Manager)
12. RLM - Release Train (if you are using Release Manager)
13. RLM - Application Release (if you are using Release Manager)
14. RLM - Release Package (if you are using Release Manager)
15. RLM - Deployment (if you are using Release Manager)
16. RLM - Environment (if you are using Release Manager)
17. RLM - Release Template (if you are using Release Manager)
18. RLM - 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.

NOTES

- If you are using Requirements Manager with your implementation of Development Manager, see the *Serena Requirements Manager Readme* and *Installation and Configuration Guide* for information on how to activate the integration elements.

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:

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

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 \[page 35\]](#)

[Configuring the Admin User for Release Manager \[page 35\]](#)

[Creating the Service User for Development Manager \[page 38\]](#)

[Using the Serena ALM Configurator \[page 40\]](#)

[Accessing the Standard SBM User Interface \[page 47\]](#)

[Release Manager Configuration \[page 58\]](#)

[Development Manager Configuration \[page 67\]](#)

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, `almsadmin`.
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 `almsadmin`. Edit `almsadmin` to give the additional unique privileges needed for Serena Release Manager.

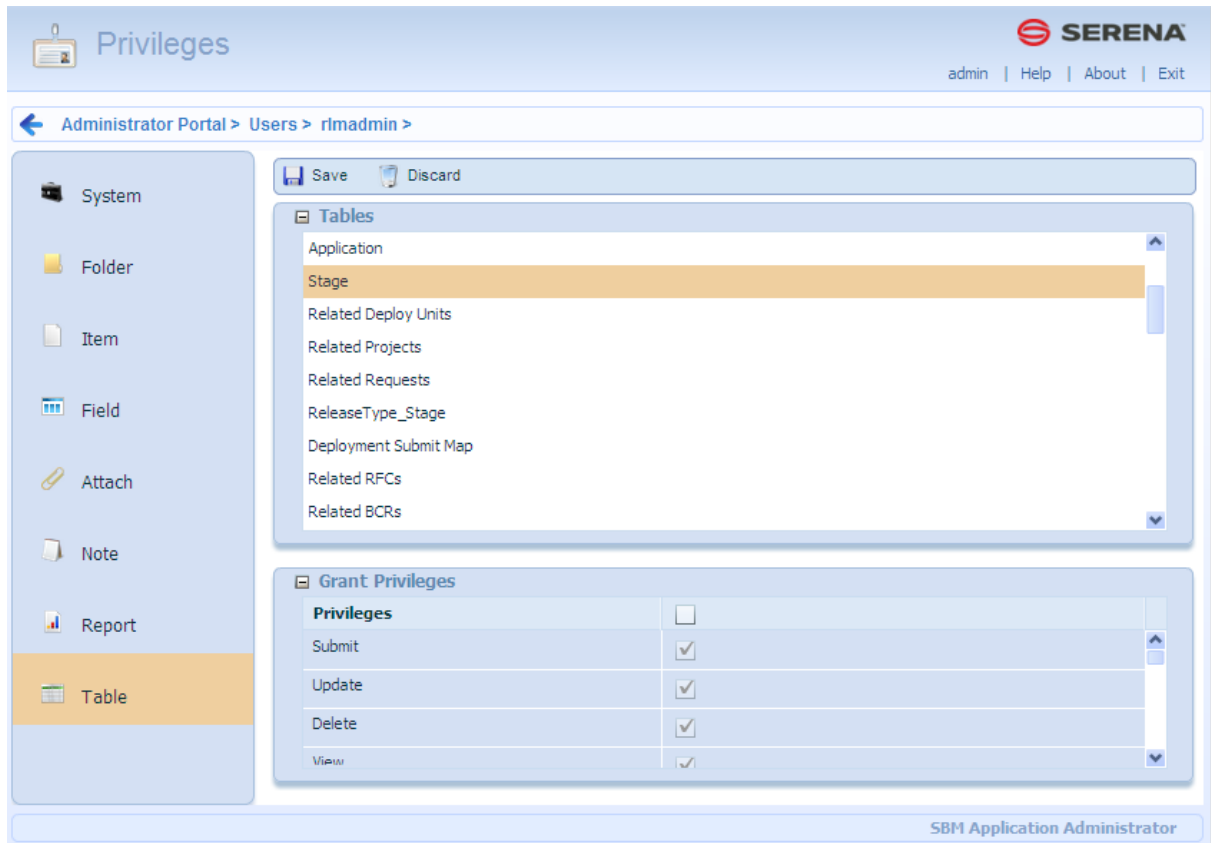
NOTE A matching administrative user ID must be set up in Dimensions CM, because single sign-on is required for the communication to work. A similar TSO user ID is required if you are implementing ChangeMan ZMF with Serena Release Manager.

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 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, a1admin.
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 role that you want to enable for this user. The role 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. The general actions and privileges controlled by each of the default Release Manager roles, Release Manager and Release Engineer, are shown in the following table.

Example

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

The screenshot shows the 'Administrator Portal > Users >' page. On the left is a sidebar with menu items: General, Roles (highlighted), Membership, Privileges, User Preferences, Notifications, Passwords, and Administration. The main area has a 'Save' and 'Discard' button, a search bar for 'Project Name', and a 'Projects' section. Under 'Projects', there's a list of project names: ALM Projects, Dev Change Requests Project, Dev Tasks Project, and Release Package (highlighted). Below this is a table with columns for 'Name' and 'Enabled'. The 'Release Manager' row is highlighted, and the 'Enabled' checkbox is checked. The user 'rlmadmin' is also listed with a checked 'Enabled' checkbox. At the bottom right, it says 'SBM Application Administrator'.

Documentation References

Complete documentation on managing role assignments in SBM is in the *Serena Business Manager 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 \[page 90\]](#).

Creating the Service User

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 *API/Script*.
- **System Privileges:** Set this to *All*.

Selecting Roles for the Service User

To select roles for the service user:

1. Assign the new service user the Administrator role for all Development Manager process apps: ALM Project, Dev Package Project, Dev Tasks Project, and Dev Change Requests 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
2. Select **Membership** and add the user to the *Everyone* group.
 3. Save your changes.

Assigning Privileges to the Service User

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://aehost: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 Serena 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 ALM, as some of the functions

implemented in Serena ALM 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 check uniqueness error.

NOTE This version of the `sbm-client-connection.properties` file is under the ALM Web services directory and is NOT the same as the same-named file under the Agile Web services directory.

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 application services, in the form of: <code>http://<sbm_hostname>:<port>/gsoap/gsoap_ssl.dll?sbmappservices72</code>
sbm.service.user	SBM service user ID with administrative privileges and access to the appropriate SBM projects and tables controlled by Development Manager. This user ID must exist in SBM.
sbm.service.password	Password for the SBM service user ID.

Example

Set the SBM provider connection for DCRs. This example gives Serena ALM the URL to connect to the SBM application services on port 80 of host aehost.

`sbm-client-connection.properties`

```
SBM_APPWS_URL = http://aehost:80/gsoap/gsoap_ssl.dll?sbmappservices72
sbm.service.user = serviceuser
sbm.service.password = serviceuserpassword
```

NOTE Although the connection URL for the application services is used by Release Manager and Development Manager, the service user ID and password is not used by Release Manager. The administrative user for Release Manager is set in the ALF client connection file. See [ALF Client Connection Information \[page 41\]](#).

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: http://<sbm_hostname>:<port>/gsoap/ gsoap_ssl.dll?sbmappservices72
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 aehost.

```
bcr-connection.properties
```

```
bcr.ws.app.url=http://aehost: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 aehost.

```
itsm-connection.properties
```

```
itsm.ws.app.url=http://aehost: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 104] .
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 95] .

Example

```
zmf-client-connection.properties
```

```
ZMF_WS_URL = http://aehost: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://aehost/tmtrack/tmtrack.dll?shell=alm
```

your Serena ALM standard SBM user interface URL would be:

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

Managing Serena ALM User Interface Reports

Reports help stakeholders stay informed of status, history, and other process-specific information information. Serena ALM 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 Serena ALM. The default reports configured for Serena ALM 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 ALM 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 ALM.

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 \[page 48\]](#).

Creating New Reports

You can create new reports and register them in the UI if you need a report for information the default reports don't provide.

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 ALM 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 \[page 49\]](#)
- [Report Reference Rules \[page 49\]](#)

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 ALM. 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 shows the Serena Business Manager interface. At the top, there are navigation tabs for 'Inbox', 'Release Trains', 'Application Releases', 'Release Packages', 'Deployment Tasks', and 'Environments'. Below these is a table with columns: 'Item ID', 'Status', 'Author', 'Official release name', 'Description', 'Release type', and 'State'. The 'Release Trains' tab is active, and a sidebar menu on the left shows 'Release Trains | View All' highlighted in yellow. The main content area displays a list of reports under the heading 'Release Train New Project (Release Train)'. The table has columns: 'Title', 'Privilege Category', 'Created', 'Author', 'Last Modified', 'Last Modifier', and 'Last Run'. The 'Release Trains | View All' report is highlighted in yellow in the list.

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 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 Release Manager 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.

List of Development Manager Reports

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

Report Title / Name in SBM Composer	Reference Name	Description
Inbox		
ALM Projects.MyWork.Inbox / Inbox.AL M Projects	ALM.MyWork.Inbox.AL MProjects	ALM projects assigned to the current user.
DVM.MyWork.Inbox.DevCRs / Inbox.DevChgReq.DevCRs	DVM.MyWork.Inbox.DevCRs	Development Change Requests assigned the current user.
DVM.MyWork.Inbox.DevPackage / Inbox.Dev Package	DVM.MyWork.Inbox.DevPackage	Development packages assigned the current user.
DVM.MyWork.Inbox.DevTasks / Inbox.Dev Tasks	DVM.MyWork.Inbox.DevTasks	Development tasks assigned the current user.
ALM Projects		
ALM Projects.View.All / View.All		All ALM projects in active status.
ALM Projects.View.Mine / View.Mine		ALM projects assigned to the current user.
ALM Projects.View.Closed / View.Closed		All ALM projects in the Closed state.
All Active ALM Projects JSON /		All ALM projects in active status, JSON version, used for what?
DVM.View All Projects /		All ALM projects in active status, used for what?
RQM.View.Project Documents / Projects for Requirement Approvals	ALMProjects.json. ProjectsForRequirementApprovals	All RQM project documents, used for integration with Requirements Manager.
Development Change Requests		

Report Title / Name in SBM Composer	Reference Name	Description
Dev CR.View.All / View.All	DevCR.View.All	All Development Change Requests in active status.
Dev CR.View.Mine / View.Mine	DevCR.View.Mine	Development Change Requests assigned to the current user.
Dev CR.View.Closed / View.Closed	DevCR.View.Closed	All Development Change Requests in the Closed state.
Dev CR.View.In Test / View.In Test	DevCR.View.InTest	All Development Change Requests in the In Test state.
Dev CR.View.In Development / View.In Development	DevCR.View.InDevelopment	All Development Change Requests in the In Development state.
Dev CR.View.Waiting to Start / View.Waiting to Start	DevCR.View.WaitingtoStart	All Development Change Requests in the Waiting to Start state.
CR(s) associated to this Dev Package / CRs associated to this Dev Package	CR4DP	All Development Change Requests associated with the currently-selected development package.
/ CRs with no associated Dev Package		All Development Change Requests that are not currently associated with a development package.
CR(s) associated to this Project / CRs associated to this Project	CR4Proj	All Development Change Requests associated with the currently-selected project.
Development Packages		
Dev Packages.View.All / View.All	DevPackages.View.All	All development packages in active status.
Dev Packages.View.Mine / View.Mine	DevPackages.View.Mine	Development packages assigned to the current user.
Dev Packages.View.Closed / View.Closed	DevPackages.View.Closed	All development packages in the Closed state.

Report Title / Name in SBM Composer	Reference Name	Description
Dev Packages.View.Package Creation / View.Package Creation	DevPackages.View. PackageCreation	All development packages in the Package Creation state.
Dev Packages.View.In Build / View.In Build	DevPackages.View.InBuild	All development packages in the In Build state.
Dev Packages.View.In Test / View.In Test	DevPackages.View.InTest	All development packages in the In Test state.
Dev Packages.View.Ready to Turnover / View.Ready to Turnover	DevPackages.View. ReadytoTurnover	All development packages in the Ready to Turnover state.
Dev Packages associated with this Project / Dev Packages associated with this Project	DP4Proj	All development packages associated with the currently-selected project.
Dev Packages associated with this Change Request / Dev Packages associated with this Change Request	DVM.embedded. DevPackagesAssociatedWith ThisChangeRequest	All development packages associated with the currently-selected change request.
Development Tasks		
Dev Tasks.View.All / View.All	DevTasks.View.All	All development tasks in active status.
Dev Tasks.View.Mine / View.Mine	DevTasks.View.Mine	Development tasks assigned to the current user.
Dev Tasks.View.Completed / View.Completed	DevTasks.View.Completed	All development tasks in the Completed state.
Dev Tasks.View.In Progress / View.In Progress	DevTasks.View.InProgress	All development tasks in the In Progress state.
Dev Tasks.View.Waiting to Start / View.Waiting to Start	DevTasks.View.WaitingtoStart	All development tasks in the Waiting to Start state.

Report Title / Name in SBM Composer	Reference Name	Description
CM Task(s) associated with this Dev Package / Tasks associated with this Dev Package	Task4DP	All development tasks associated with the currently-selected development package.
Task(s) associated with this CR / Tasks associated with this CR	Task4CR	All development tasks associated with the currently-selected development change request.
Task(s) associated with this Baseline / Tasks associated with this Baseline	Task4BL	All development tasks associated with the currently-selected baseline.
/ Tasks with no associated Dev Package		All development tasks that are not currently associated with a development package.
Task(s) associated with this Project / Tasks associated with this Project	DVM.embedded. TasksAssociatedWithThisProject	All development tasks associated with the currently-selected project.
ZMF Task(s) associated with this Dev Package /		All ChangeMan ZMF development tasks associated with the currently-selected development package.
Epics		
NOTE The Epic report usage is defined as part of the Agile configuration. The default process app is <i>DVM - Development Change Requests</i> , but any process app may be mapped to the Epic process in your implementation. See Agile Configuration [page 72] .		
Dev CR.View.Epics /		All epics associated with a dev change requests.
Epic Storie(s) to associate this Change Request to /		Epic stories to associate with this dev change request.
DVM.View All Epics / View.All		All epics in active status.

Report Title / Name in SBM Composer	Reference Name	Description
/ View.Mine		Epics assigned to the current user.
/ View.Closed		All epics in the Closed state.
/ View.In Test		All epics in the In Test state.
/ View.In Development		All epics in the In Development state.
/ View.Waiting to Start		All epics in the Waiting to Start state.
CRs associated to this Dev Package / CRs associated to this Dev Package		All epics associated with the currently-selected development package.
/ CRs with no associated Dev Package		All epics that are not currently associated with a development package.
CR(s) associated to this Project / CRs associated to this Project	CR4Proj	All epics associated with the currently-selected project.
CR(s) associated to this Dev Package / CRs associated to this Dev Package	CR4DP	All epics associated with the currently-selected development package.
/ CRs with no associated Dev Package		All epics that are not currently associated with a development package.
CR(s) associated to this Project / CRs associated to this Project	CR4Proj	All epics associated with the currently-selected project.

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

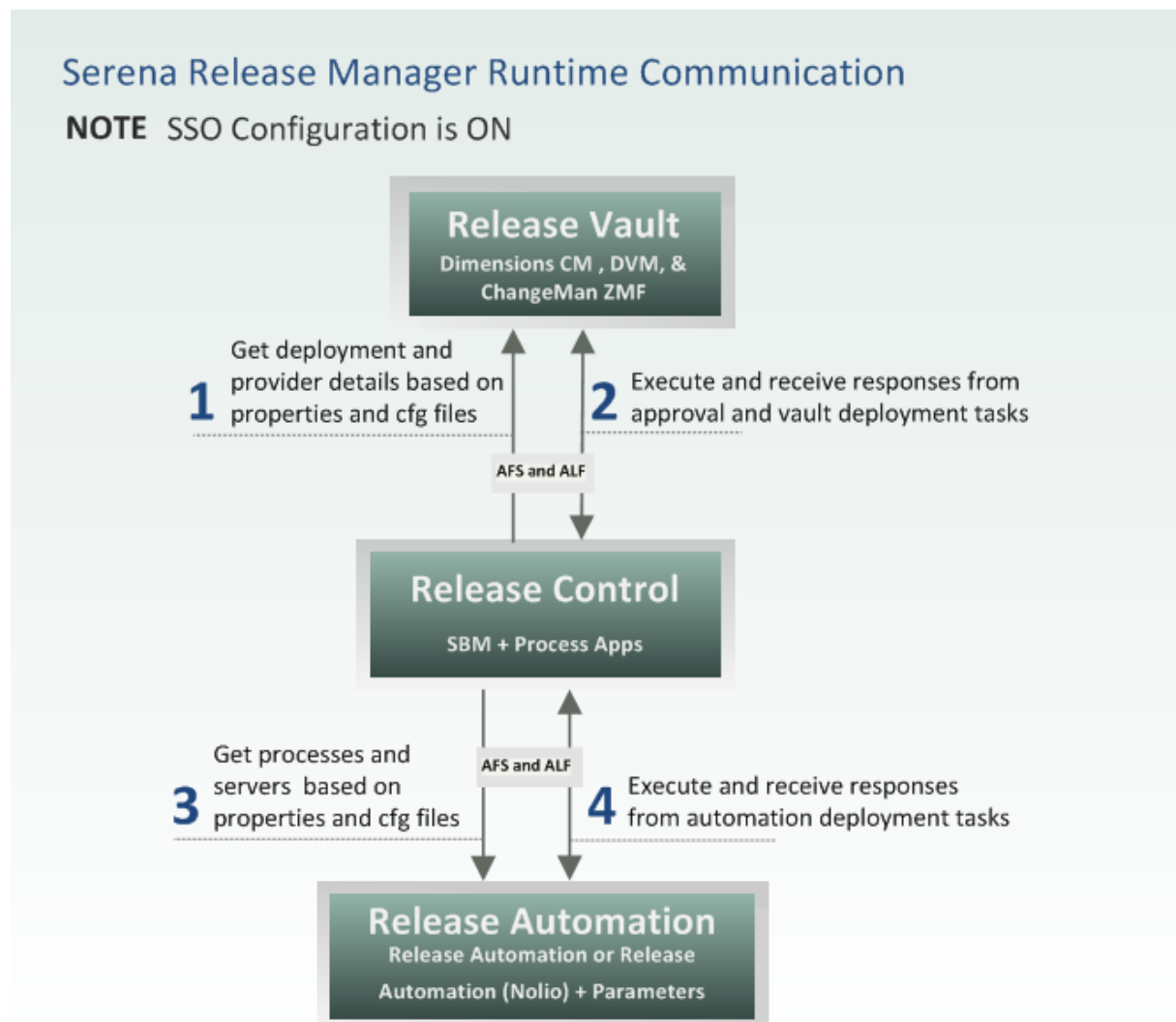
Release Manager Configuration

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

Example

The default roles defined in Serena Release Manager are as follows:

- **Release Manager** role is required for most of the projects. It is the primary owner for most of the workflows and secondary for all except the vault deployment tasks and the environment workflows.
- **Release Engineer** role is required for deployment task workflows for a user to deploy baselines, as it is the only owning role of the vault deployment task.

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 Application Administrator Guide* in "Managing Notifications".

Development Manager Configuration

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

Most of the configuration for Development Manager is done as part of the Installer scripts and during promotion of the snapshots for a sample implementation, but for an actual implementation, you'll need to configure your users, groups, roles, and privileges according to your organization's needs.

Creating and Assigning the Development Manager Users

Sample users and groups, role assignments, and project roles are included as part of the packaged snapshot of Development manager. To take advantage of this configuration jumpstart, you must be careful to promote and deploy the *Global Process App* snapshot first during the promotion and deployment processes. If you have already promoted and did not do this, you'll need to create the users, groups, and roles and assign and configure them as documented in this section.

Use the following lists and procedures if you need to add or modify your Development Manager user, group, role, and privilege information.

- [List of Default Development Manager Users, Groups, and Roles \[page 67\]](#)
- [List of Default Roles for Each Project \[page 68\]](#)
- [Creating Groups in SBM \[page 70\]](#)
- [Assigning Groups to Roles in Projects \[page 70\]](#)
- [Creating Users in SBM \[page 70\]](#)
- [Assigning Group Membership \[page 70\]](#)
- [Assigning Group Privileges to the ALM UI Settings Table \[page 71\]](#)
- [Specifying Group Properties for Agile \[page 71\]](#)

List of Default Development Manager Users, Groups, and Roles

The default installation of Development Manager includes a set of users that have the appropriate groups and roles assigned in the process apps. If you have promoted and deployed the *Global Process App* as the first process app, you will not need to create and assign these users.

The default set of users, groups, and roles are given in the following table. You may use this information as an example for your own user, group, and role configuration.

User	Group	Role
Alexander	Build Engineers	Build Engineer
Amy	Build Managers	Build Manager
Ashley	QA Engineers	QA Engineer
Dave	Release Managers	Release Manager

Dinesh	QA Managers	QA Manager
Emma	Development Engineers	Development Engineer
Eva	Component Managers	Component Manager
Josh	Development Managers	Development Manager
Kaci	Product Owners	Product Owner
Matt	Business Analysts	Business Analyst
Mira	Project Managers	Project Manager
Oscar	Release Engineers	Release Engineer
Sam	Scrum Masters	Scrum Master
serviceuser	Administrators	Administrator

List of Default Roles for Each Project

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 roles you must assign in each of the default projects are shown in the following table.

Project	Role
For All Development Manager	
ALM Projects	Administrator
	Build Engineer
	Build Manager
	Business Analyst
	Development Engineer
	Development Manager
	Product Owner (If using the Agile option)
	Project Manager
	QA Engineer
	QA Manager
	Release Engineer
	Release Manager
	Scrum Master (If using the Agile option)
	User

Dev Change Requests Project	Administrator
	Build Engineer
	Build Manager
	Business Analyst
	Development Engineer
	Development Manager
	Product Owner (If using the Agile option)
	Project Manager
	QA Engineer
	QA Manager
	Release Engineer
	Release Manager
	Scrum Master (If using the Agile option)
	User
Dev Tasks Project	Administrator
	Build Engineer
	Build Manager
	Business Analyst
	Development Engineer
	Development Manager
	Product Owner (If using the Agile option)
	Project Manager
	QA Engineer
	QA Manager
	Release Engineer
	Release Manager
	Scrum Master (If using the Agile option)
	User
Dev Tasks Project	Administrator
	Build Engineer
	Build Manager
	Business Analyst
	Development Engineer
	Development Manager
	Product Owner (If using the Agile option)
	Project Manager
	QA Engineer
	QA Manager
	Release Engineer
	Release Manager
	Scrum Master (If using the Agile option)
	User
For Products and Components option only (Requires the DVM - Products and Components solution)	

Component Lifecycle Project

Product Manager

NOTE For Agile, Team Member does not have a specific role; any user may be a Team Member.

Creating Groups in SBM

If you do not have the groups you need already, such as the sample groups, you may create your own groups in SBM.

To create groups in SBM:

1. Navigate to the **Administrator Portal**.
2. Click **Groups** in the main window.
3. Click **Add** and provide the group details.
4. Click **Save**.

Assigning Groups to Roles in Projects

When you create your own groups, you will need to assign the correct roles for those groups.

To assign groups to roles in SBM:

1. In the **Administrator Portal**, select the group.
2. Select **Roles**.
3. Double-click *Base Project*.
4. Select a project and select the required role. For example for the Development Manager user, select Development Manager. See the tables in the preceding topics.
5. Select the **User** role.
6. Repeat the previous steps for each of the groups, projects, and roles in the preceding tables.
7. Click **Save**.

Creating Users in SBM

Create your own users and assign them to groups as needed.

To create users in SBM:

1. Navigate to the **Administrator Portal**.
2. Click **Users** in the main window.
3. Click **Add**, and enter the details for the user, selecting the option **Regular User**.
4. Click **Save**.

Assigning Group Membership

To assign users to groups in SBM:

1. In the **Administrator Portal**, select the group.

2. Select **Membership**.
3. Assign the users to this group according to the [List of Default Development Manager Users, Groups, and Roles \[page 67\]](#).
4. Give each user membership in a group that has the required privileges for ALM_USER_SETTINGS auxiliary table. All users must have Submit, Update, and View privileges to this table. Otherwise, the Serena ALM UI won't appear as it should to those users. See [Assigning Group Privileges to the ALM UI Settings Table \[page 71\]](#).
5. Repeat the previous steps for each of the projects where the role is listed.
6. Click **Save**.

TIP Roles can be assigned to users or groups. They can be assigned from the Users, Groups, or Project details view. See the *Serena Business Manager Application Administrator Guide*.

Assigning Group Privileges to the ALM UI Settings Table

To set 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 group that Serena ALM users share. You may use the Everyone group for this, depending on how you like to control privileges for your groups and users.
2. Select Tables, and select the ALM_USER_SETTINGS table.
3. Select Submit, Update, and View under Grant Privileges.

Specifying Group Properties for Agile

For each Development Manager Agile configuration, in addition to setting up and assigning the Product Owner and Scrum Master groups and roles as shown in the preceding sections, you must ensure the XML configuration file information for these users, groups, and roles is correct.

1. Navigate to the Development Manager Agile Web services classes directory. For example:


```
..\Program Files\Common\tomcat\6.0\webapps\alm#agile\WEB-INF\classes
```
2. Open the `dvm-project.properties.xml` file.

3. Make sure that the group names match the default group names in the associated XML tags. For example:

```
<dvm-project-context>
  <solution>
    <unique-name>ALM_PROJECTS</unique-name>
    <update-transition>ALM_PROJECTS.UPDATE1</update-transition>
    <submit-transition>ALM_PROJECTS.SUBMIT</submit-transition>
    <role-product-owners-group-name>Product Owners</role-product-owners-group-name>
    <role-scrum-masters-group-name>Scrum Masters </role-scrum-masters-group-name>
    <projects>
      <project>
        <project>ALM Projects</project>
      </project>
    </projects>
    <source-reports>
      <source-report>
        <search-by-name>DVM</search-by-name>
        <include-sub-projects>>true</include-sub-projects>
        <category>ALL</category>
      </source-report>
    </source-reports>
  </solution>
</dvm-project-context>
```

Documentation References

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

Agile Configuration

This section tells you how to configure Agile for Development Manager.

General Agile Configuration

If you are using the Agile features of Development Manager, you may change default settings in the configuration properties file for the following behaviors:

- Showing text of SQL commands (Default = true)
- Agile database auto-generation on application startup (Default = false)
- SSO internal settings used for Web services (Default = true)
- Items per page returned by the Agile web services (Default = 50)
- Role names shown in the Agile Web UI. The names must be valid roles in SBM.

Default:

- role.PRODUCT_OWNER= Product Owner
- role.SCRUM_MASTER = Scrum Master
- role.TEAM_MEMBER= Development Engineer,QA Engineer

- Allowing burndown chart updates on weekends (Default = false)

To update the general Agile properties:

Edit the configuration properties file is located in:

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

When you have completed any change to this file, restart Tomcat.

Agile Connection Information

Specify the instance of SBM if you are using the Agile Development feature of Development Manager. This information is added as part of the installer. If you entered this information correctly while running the installer, you do not need to update it.

NOTE This version of the `sbm-client-connection.properties` file is under the Agile Web services directory and is NOT the same as the same-named file under the ALM Web services directory.

To update the Agile connection information:

1. In `<tomcat home>\webapps\alm#agile\WEB-INF\classes\sbm-client-connection.properties`, make sure the variables are set to the correct connection information for your installation of Agile as follows:

Parameter	Value
SBM_APPWS_URL	URL to the SBM server that provides the application services, in the form of: http://<sbm_hostname>:<port>/gsoap/gsoap_ssl.dll?sbmappservices72
SBM_ADMINWS_URL	URL to the SBM server that provides the administrative services, in the form of: http://<sbm_hostname>:<port>/gsoap/gsoap_ssl.dll?sbmadminservices72
sbm.service.user	SBM service user ID with administrative privileges and access to the appropriate SBM projects and tables controlled by Agile for Development Manager. This user ID must exist in SBM.
sbm.service.password	Password for the SBM service user ID.

Example

Set the SBM provider connection for Agile. This example gives Agile the URL to connect to the SBM application services and SBM administrative services on port 80 of host aehost.

```
sbm-client-connection.properties
```

```
SBM_APPWS_URL = http://aehost:80/gsoap/gsoap_ssl.dll?sbmappservices72
SBM_ADMINWS_URL = http://aehost:80/gsoap/gsoap_ssl.dll?sbmadminservices72
sbm.service.user = serviceuser
sbm.service.password = serviceuserpassword
```

Enabling the Agile Functionality on IIS 7.5

To get Agile functionality fully working on IIS 7.5, you must allow the Serena Common Tomcat Web Server to pass through HTTP error messages directly to the client.

If this setting is not configured properly, the user interface interactions that change Agile objects do not work properly. For example, you will be unable to edit backlogs or re-rank backlog items.

This setting is the default beginning with SBM 10.1.2.1, so if you are using SBM 10.1.2.1 or above, you may skip this configuration. If you are using SBM 10.1.2 or prior, you must ensure this setting is applied.

You must change the configuration of the ISAPI redirector that is installed as an ISAPI filter so it will pass through HTTP error messages from Tomcat directly to the client.

To configure the ISAPI filter to pass the messages directly to the client:

1. In the root node of SBM Application Engine, open the Web.config file. For example, `..\Program Files\Serena\SBM\Application Engine\bin\Web.config`.
2. In the configuration section, add or update the system.webServer section as follows:

```
<configuration>
...
  <system.webServer>
    <httpErrors existingResponse="PassThrough"/>;
  </system.webServer>
...
</configuration>
...
```

Configuring Agile to Use Different Process Apps

The architecture of the Development Manager Agile components is designed so that implementers can replace the default agile project and story and task process apps and process definitions with their own. Epics share a process app and configuration file with stories.

Configuring the Properties

- [Configuring the Project Context Properties \[page 78\]](#)
- [Configuring Epic and Story Properties \[page 75\]](#)
- [Configuring Task Properties \[page 76\]](#)
- [Configuring the Item Hierarchy Properties \[page 78\]](#)

CAUTION! The agile object type names in the story and task property files have to match the names in the item hierarchy file; if you change these in any of the property files, you must also change them to match in the other files.

Examples and More Information

For examples and more details on mapping the XML information to the related user interface, see the Serena Support website Knowledgebase.

Configuring Epic and Story Properties

Epics and stories can be configured to work with your own workflow, or SBM Process App.

- State transitions are defined by the workflow.
- Epic and Story states are workflow states. SBM states can be mapped to Agile phases.
- By default, Development Manager Epics and Stories are configured as Dev Change Requests.

The epic and story property definitions are stored in the following XML file.

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

Modifying the Epic Mapping

In the `story.properties.xml`

under:

```
<solution>
```

```
<type-name>EPIC</type-name>
```

specify the process app, update transition, and submit transition as shown in this example:

```
<external-tool-type>SBM</external-tool-type>
<unique-name>your-process-app-name</unique-name>
<update-transition>your-process-app-update-transition</update-transition>
<submit-transition>your-process-app-submit-transition</submit-transition>
</solution>
```

Make sure the project hierarchy tags point to the epic project in your process app. For example:

```
<projects>
<project>
<project>your-project-name</project>
</project>
</projects>
```

TIP Open your process app in SBM Composer and select the transition you want to map in the workflow. In the Property Editor, click **General** and use the Internal name in the corresponding transition in the XML file.

You can also map priority and story points, which are shown in a header on the form.

Mapping the Epic Overview Report to a Specified Project

An epic is simply a special kind of story. Therefore, it does not have a separate process app. However, it does have its own project within the Story process app. This example shows mapping the Epic Overview UI report to the IDM project. This assumes that the project information has already been configured in the `story.properties.xml` file.

1. From the native SBM interface, select the project for which you want to display epics in the UI. For example, IDM Project.
2. Click **Reports**.
3. Create a new report.

4. Set the **Report Project** to IDM Project.
5. Click **Save**.
6. Set the **Report Title** to DVM.View All Epics.
7. Set **Privilege Category** to User.
8. Click **Finish**.

Modifying the Story Mapping

Modifying the Story mapping is similar to modifying the Epic mapping, except that you change the tags `<story-context>` under `<type-name>` STORY.

To make the Story Panel and Epic Overview panes work properly, you need to configure the reports. The reports show the columns you want to show in the UI.

For example, IM Project has columns selected to display:

Issue Id
Summary
Owner
State

In the `<view-report>` section, specify the report reference name and title for each of the UI elements.

For example, replace the default reports with your process app's reports:

```
<view-report>  
<report-view-all-reference>DevCR.View.All</report-view-all-reference>  
<report-inbox-projectparm></report-inbox-projectparm>  
<report-inbox-title>DVM.MyWork.Inbox.DevCRs</report-inbox-title>  
<report-inbox-reference>DVM.MyWork.Inbox.DevCRs</report-inbox-reference>  
<report-view-all-projectparm>F_TS_ASSOCIATED_PROJECT</report-view-all-projectparm>
```

Configuring Task Properties

Tasks can be configured to connect to any workflow, or SBM Process App.

- State transitions are defined by the workflow.
- Task states are workflow states. SBM states can be mapped to Agile phases.
- By default, Development Manager Tasks are configured as Dev Tasks.

The task property definitions are stored in the following XML file.

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

Modifying the Task Mapping

Task process app mapping is very similar to Epic and Story process app mapping except that there are more mappings for states.

To map the agile task states:

For example, replace the default states with your process app's states:

```
<agile-states>
<state-map>
<mapped-state-display-name>Defined</mapped-state-display-name>
<mapped-state-internal-name>DEFINED</mapped-state-internal-name>
<state>
<mapping>DEV_TASKS.RAISED</mapping>
</state>
</state-map>
<state-map>
<mapped-state-display-name>In Progress</mapped-state-display-name>
<mapped-state-internal-name>IN_PROGRESS</mapped-state-internal-name>
<state>
<mapping>DEV_TASKS.UNDER_WORK</mapping>
</state>
<state>
<mapping>DEV_TASKS.PEER_REVIEW</mapping>
</state>
</state-map>
<state-map>
<mapped-state-display-name>Done</mapped-state-display-name>
<mapped-state-internal-name>DONE</mapped-state-internal-name>
<state>
<mapping>DEV_TASKS.COMPLETE</mapping>
</state>
<state>
<mapping>DEV_TASKS.CLOSED</mapping>
</state>
</state-map>
```

To map the agile task reports:

To make the task UI elements work properly, you need to configure the reports.

In the <view-report> section, specify the report reference name and title for each of the UI elements.

For example, replace the default task reports with your process app's reports:

```
<view-report>
<report>Built-In: All Items</report>
<report-view-all-title>DevTasks.View.All</report-view-all-title>
<report-view-all-reference>DevTasks.View.All</report-view-all-reference>
<report-inbox-projectparm></report-inbox-projectparm>
<report-inbox-title>DVM.MyWork.Inbox.DevTasks</report-inbox-title>
<report-inbox-reference>DVM.MyWork.Inbox.DevTasks</report-inbox-reference>
<report-view-all-projectparm>U_278528c1-4fdc-4606-9611-1212cc1290bb</report-view-
all-projectparm>
```

Understanding Task Priority and Effort

The Priority, Estimated Efforts, Actual Efforts, and Remaining Efforts are not associated with the process apps. These are stored in the Agile DB. This is in the task header.

Configuring the Project Context Properties

Agile items are submitted into an SBM project and identified uniquely as Agile in the Serena ALM user interface. In the project context properties file, you specify the name of the process app's base project, subprojects, and submit and update transitions that you want to use for Agile items. You also specify the SBM group names and report identifiers for the Agile UI.

If you change the process app and project structure for Agile items, you must change the project context file accordingly.

The project context properties are stored in the following XML file.

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

For example:

```
<dvm-project-context>
  <solution>
    <unique-name>ALM_PROJECTS</unique-name>
    <update-transition>ALM_PROJECTS.UPDATE1</update-transition>
    <submit-transition>ALM_PROJECTS.SUBMIT</submit-transition>
    <role-product-owners-group-name>Product Owners</role-product-owners-group-name>
    <role-scrum-masters-group-name>Scrum Masters </role-scrum-masters-group-name>
    <projects>
      <project>
        <project>ALM Projects</project>
      </project>
    </projects>
    <source-reports>
      <source-report>
        <search-by-name>DVM</search-by-name>
        <include-sub-projects>true</include-sub-projects>
        <category>ALL</category>
      </source-report>
    </source-reports>
  </solution>
</dvm-project-context>
```

NOTE *ALM Projects* is the internal name for the *DVM - Development Projects* process app.

Configuring the Item Hierarchy Properties

The item hierarchy for agile defines the items types that are used for agile. If you change these, you must also change them in the story and task properties files.

CAUTION! Before making changes to this file, you should first understand the entire agile architecture, as there are many dependencies among the agile objects.

The item hierarchy properties are stored in the following XML file.

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

For example:

```
<item-hierarchy-context>
  <items>
    <item>
      <!-- The item name must be a valid story item type as defined in the
           AG_STORY_ITEM_TYPE table. The related-solution name must point to a valid
           solution name in the story.properties.xml file. -->
      <name>EPIC</name>
      <related-solution>EPIC</related-solution>
    </item>
  </items>
  <item>
    <name>STORY</name>
    <related-solution>STORY</related-solution>
  </item>
</items>
</item-hierarchy-context>
```

NOTE Epic and Story share the same process app in the default implementation.

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 Communication Configuration Overview \[page 80\]](#)

[Configuring Communication on the Dimensions CM Server \[page 81\]](#)

[Configuring Objects in Dimensions CM \[page 84\]](#)

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 both Serena Release Manager and Serena Development Manager (DVM) installed as part of the Serena ALM suite, your implementation of Dimensions CM is shared between them.

Configuring Dimensions CM Properties in Serena ALM

You must configure Dimensions CM properties on the Serena ALM 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 Dimensions CM Objects for Use in Serena ALM

Before you begin using the Serena ALM features that integrate with Dimensions CM, you must configure the Dimensions CM objects needed such as the Global Stage Lifecycle, Users, and Projects and Streams.

Specifying the Available Selection of Requests and Baselines

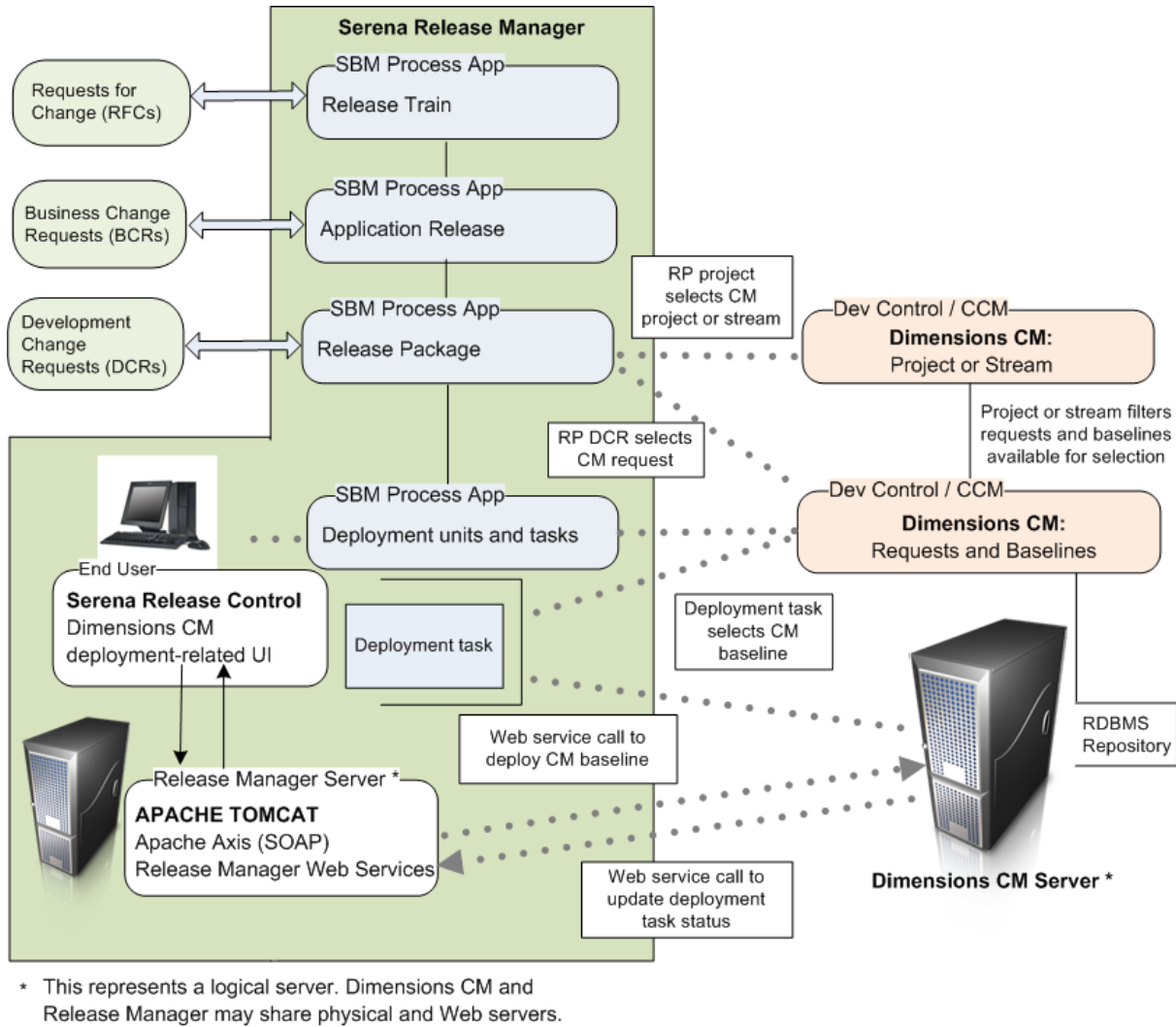
Dimensions CM requests and baselines may be used for Release Manager and/or Development Manager development change requests and Release Manager deployment units and/or Development Manager baselines respectively.

- Dimensions CM requests are displayed for selection in Serena ALM only if they are in one of the statuses defined by your Serena ALM administrator.
- Dimensions CM baselines are displayed for selection in Serena ALM only if they are in release mode and are in one of the statuses defined by your Serena ALM administrator.

For more information, see [Provider Configuration \[page 122\]](#).

Serena ALM to Dimensions CM Architecture

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



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 and Development 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
- [Specifying Selection Criteria for Dimensions CM Events and Objects \[page 83\]](#)

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. 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.

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

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. Update the related ALF information.
4. Restart the Dimensions CM Listener service.

Release Manager and Development Manager 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
```

Development Manager Only 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
```

NOTE The DM_ALF_USER and DM_ALF_PASSWORD are used for Release Manager only. Development Manager uses the service user ID and password set in the SBM client connection file and in Development Manager orchestrations.

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 \[page 122\]](#).

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 and if you are using Release Manager, also specify baseline type and deploy event.
3. Restart the Serena Dimensions Listener service.

Release Manager and Development Manager 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 Objects in Dimensions CM

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

The following sections explain the configuration and administration needed.

- [Configuring the Dimensions CM Global Stage Lifecycle \[page 84\]](#)
- [Managing Dimensions CM Users \[page 85\]](#)
- [Configuring Dimensions CM Projects and Streams \[page 92\]](#)

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 Roles and Privileges

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.

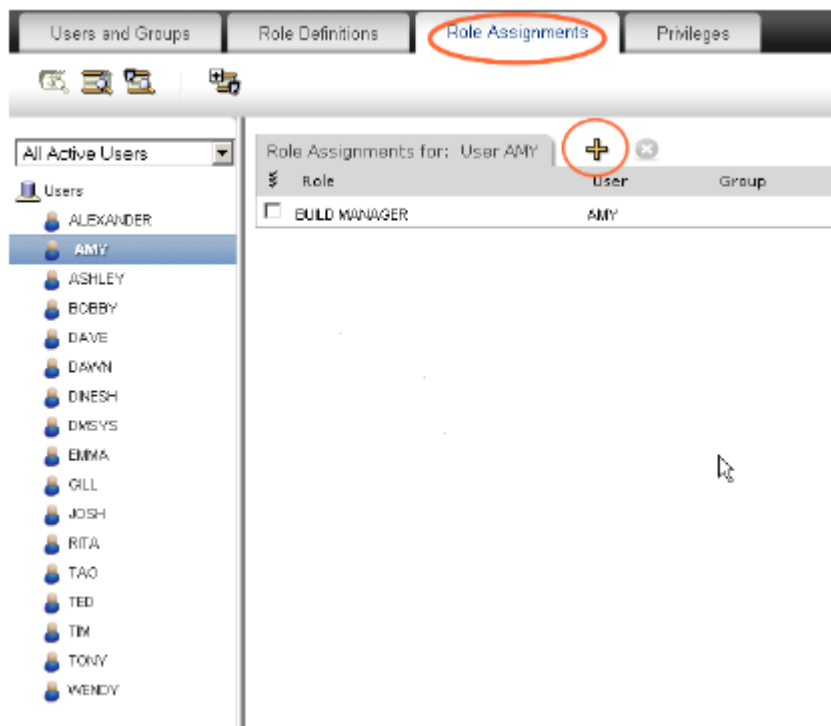
To do this:

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

In a browser, enter the URL. For example:

`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.

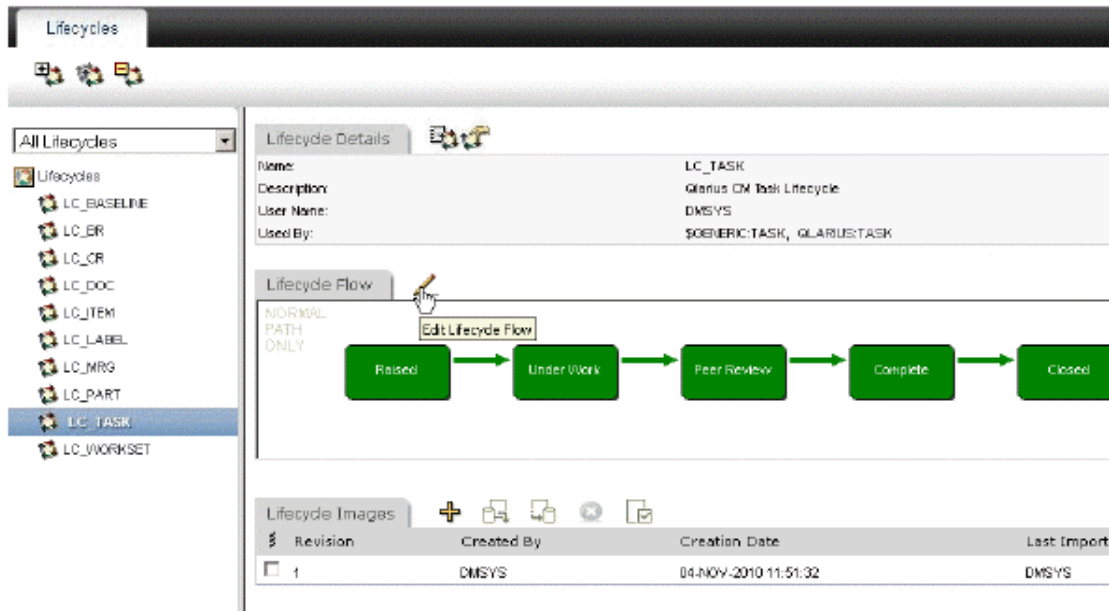
To do this:

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**.

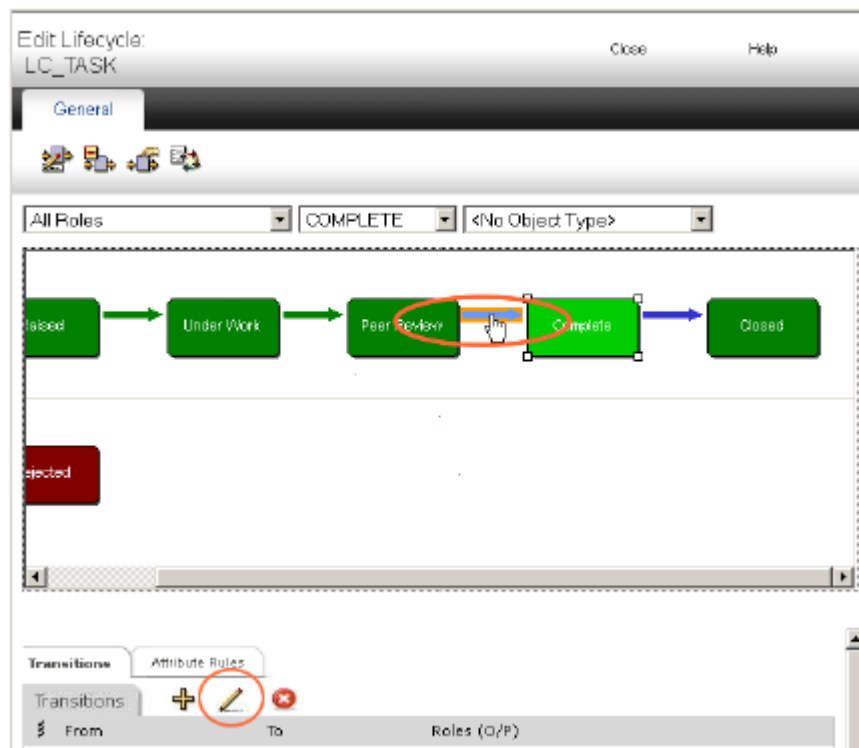
Assign Development Engineer Role to TASK Lifecycle Transitions

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.

Edit Transition: PEER REVIEW:COMPLETE

From State: PEER REVIEW To State: COMPLETE

Is a Normal Transition

Roles	Name	Optional	Pending
<input type="checkbox"/>	ORIGINATOR	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BUILD ENGINEER	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BUILD MANAGER	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	CHANGE-MANAGER	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	DEPLOY-MANAGER	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	DEVELOPER	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	DEVELOPMENT ENGINEER	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	DEVELOPMENT MANAGER	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	DOCUMENTER	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	GRAPHIC ARTIST	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	IMPLEMENTOR	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	NETWORK-ADMINISTRATOR	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	PARTS-CONTROLLER	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	PRODUCT-MANAGER	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	PROJECT-MANAGER	<input type="checkbox"/>	<input type="checkbox"/>

7. Click OK.
8. Select 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.
11. Click OK.

Edit Lifecycle:
LC_TASK

General

All Roles | COMPLETE | <No Object Type>

NORMAL

OFF-NORMAL

Rejected

Transitions | Attribute Rules

Transitions + - ✖

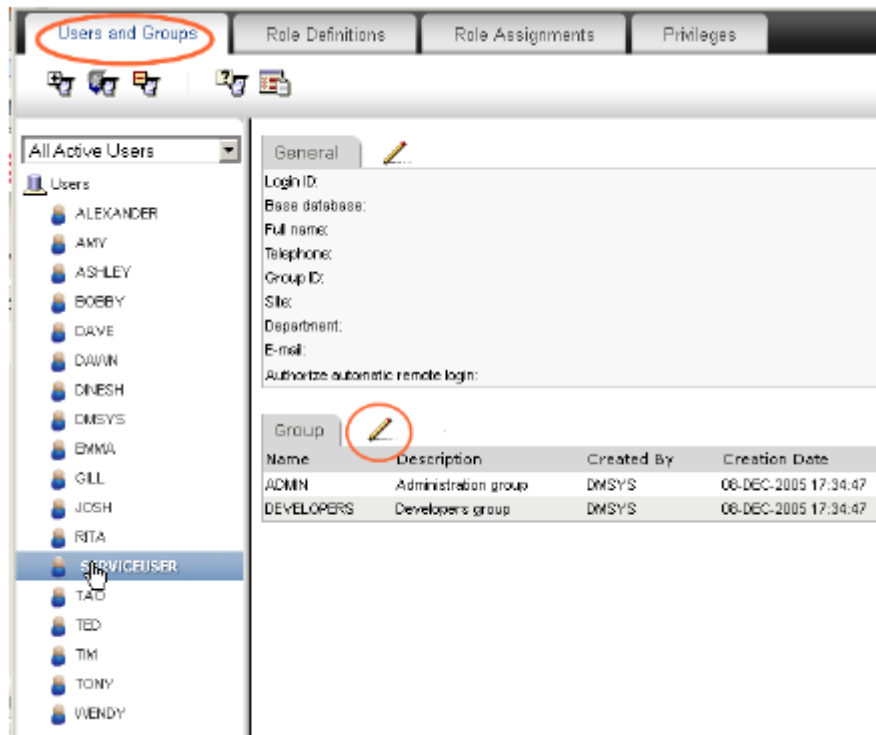
From	To	Roles (O/P)
<input checked="" type="checkbox"/> COMPLETE	CLOSED	RELEASE MANAGER P QA MANAGER P DEVELOPMENT MANAGER P DEVELOPMENT ENGINEER P BUILD MANAGER P \$ORIGINATOR P
<input type="checkbox"/> PEER REVIEW	COMPLETE	REVIEWER P DEVELOPMENT MANAGER P DEVELOPMENT ENGINEER P DEVELOPER P

Filtered by All Roles, COMPLETE, <No Object Type>

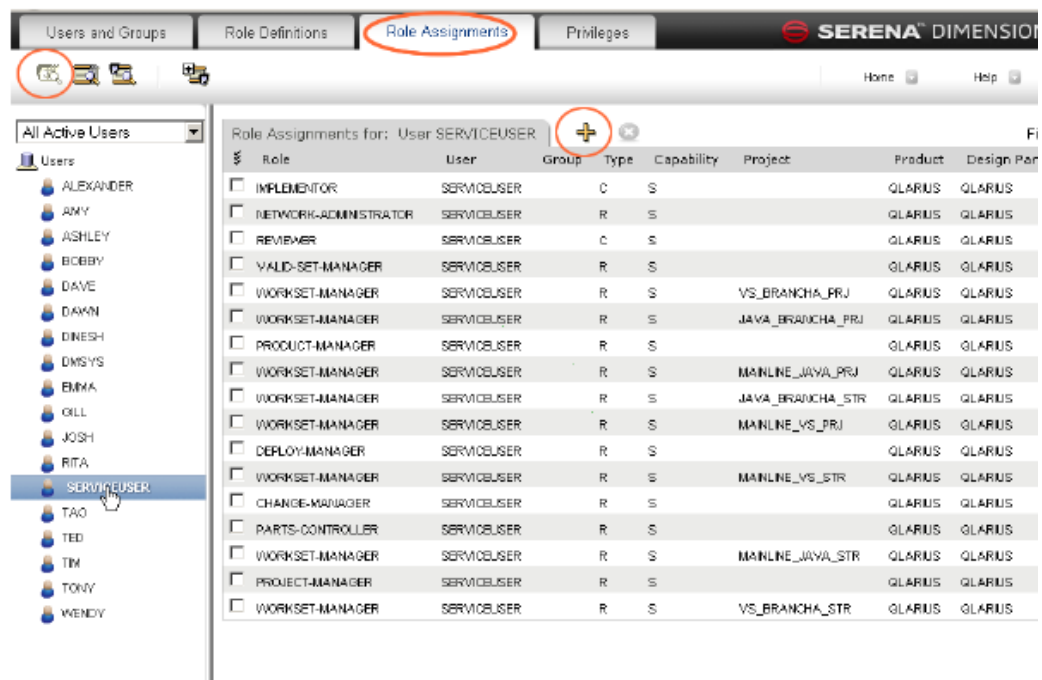
Setting Privileges for the Service User

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.

1. Open the Administration Console.
2. Select Users and Groups.
3. 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.
4. 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 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".

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.

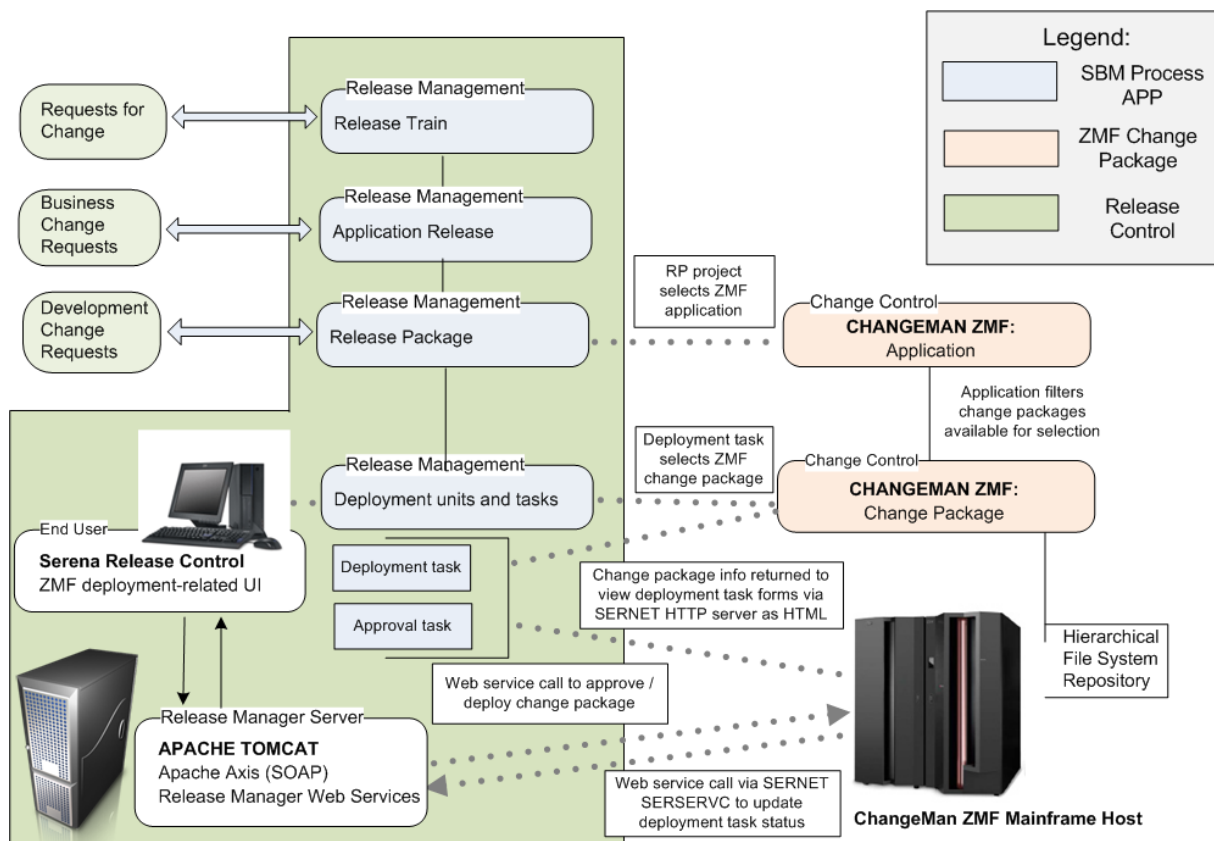
[ZMF Communication Configuration Overview \[page 94\]](#)

[Configuring ZMF Communication on the Mainframe \[page 94\]](#)

[Configuring ZMF Communication in Serena ALM \[page 106\]](#)

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 JESMSGLOG (JES message log).

Documentation References

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

Configuring 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 \[page 108\]](#).
- 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 \[page 122\]](#).

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 Prequisites

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 JESMSGLG of the job output from a previous execution of ChangeMan ZMF.
 - b. At the top of the message log, usually next to the message


```
$HASP373 STARTED
```

 find message IEF695I. This message includes an ASSIGNED message line that identifies the SERNET user ID.
2. Retrieve the OMVS segment for the SERNET user ID.
 - a. At the TSO command line, issue the following command:


```
LU userid OMVS
```

 where `userid` is the SERNET user ID found above, such as `sernet`.
 - b. If no OMVS segment is returned, ask your systems programmer or security administrator to add one. (See [SERNET User ID \[page 97\]](#) 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:

```

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

Pathname . : /u/sernet

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

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

Press Enter.

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

For example:

```

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

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

Permissions . . 755 (Octal)

Link . . . . .          +

File Type . . . 1  1. Directory          Options
                   2. Regular file      Set sticky bit
                   3. FIFO              Copy...
                   4. Symbolic Link     Edit...
                   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/serservc', * 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 serservc 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(SERSERVP)
//SYSPRINT DD  SYSOUT=*
. . .

```

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

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

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

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

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

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

5. Run SERSERVI.

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

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

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

Verifying the Installation of SERSERVC

To verify the installation of SERSERVC, do the following:

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

```
/S SERSERVC
```

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

- a. At the TSO command line, enter:

```
SERCMD PING
```

- b. You should receive the response ok.

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

- a. From any Web browser, type

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

where

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

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

- b. You should receive the response ok.

SERSERVC Runtime Considerations

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

Startup and Shutdown

Console Commands

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

```
/S 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 SERSERVC \[page 98\]](#), substituting another name for this instance, such as SERSERVD, in place of SERSERVC.
- Specify a different directory and port for this instance according to your company's installation standards.

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

- Production SERSERVC example:

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

- Development SERSERVC example:

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

I. Select a ZMF form II. Select the ZMF info tab III. Select a sub-tab IV. Select the widget

V. Change localhost to your hostname

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 \[page 110\]](#)

[Configuring Communication with Release Automation \[page 111\]](#)

[Configuring Communication with Release Automation \(Nolio\) \[page 115\]](#)

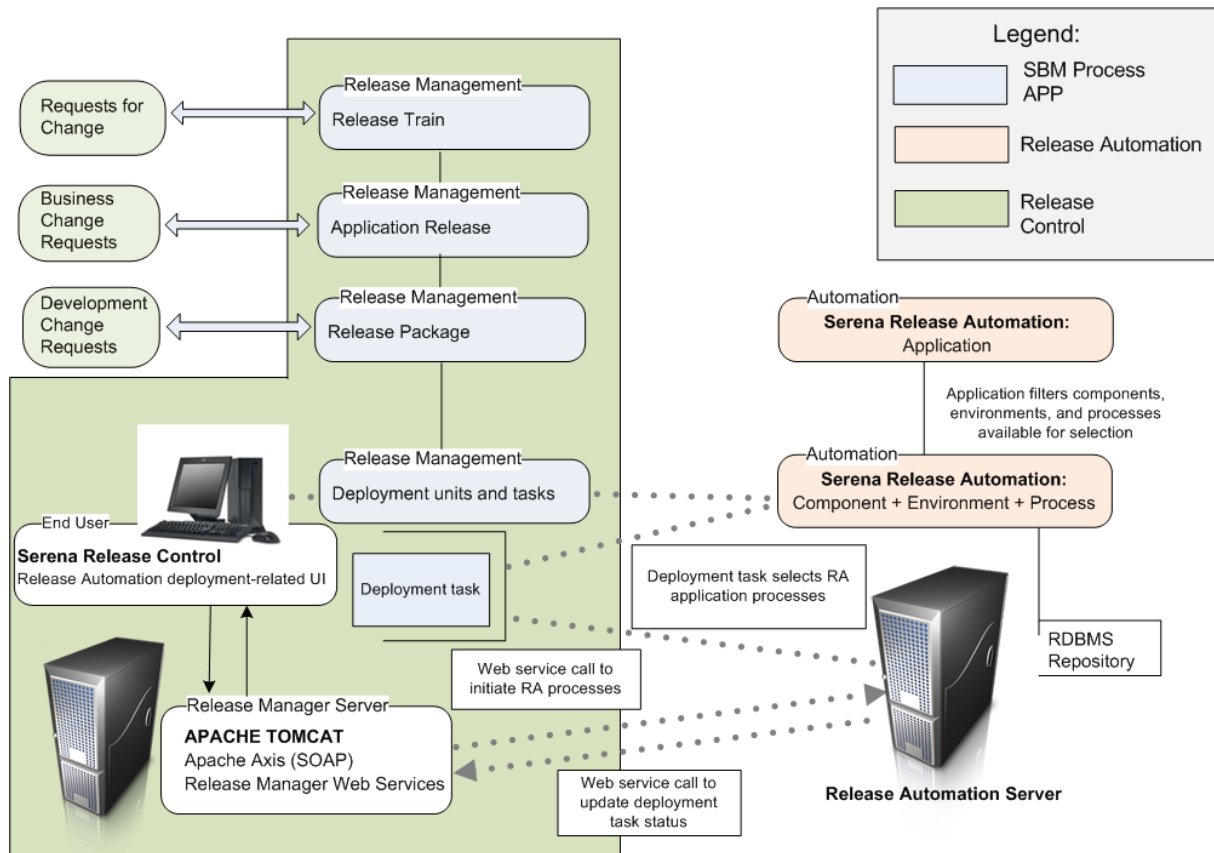
Release Automation Communication Configuration Overview

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

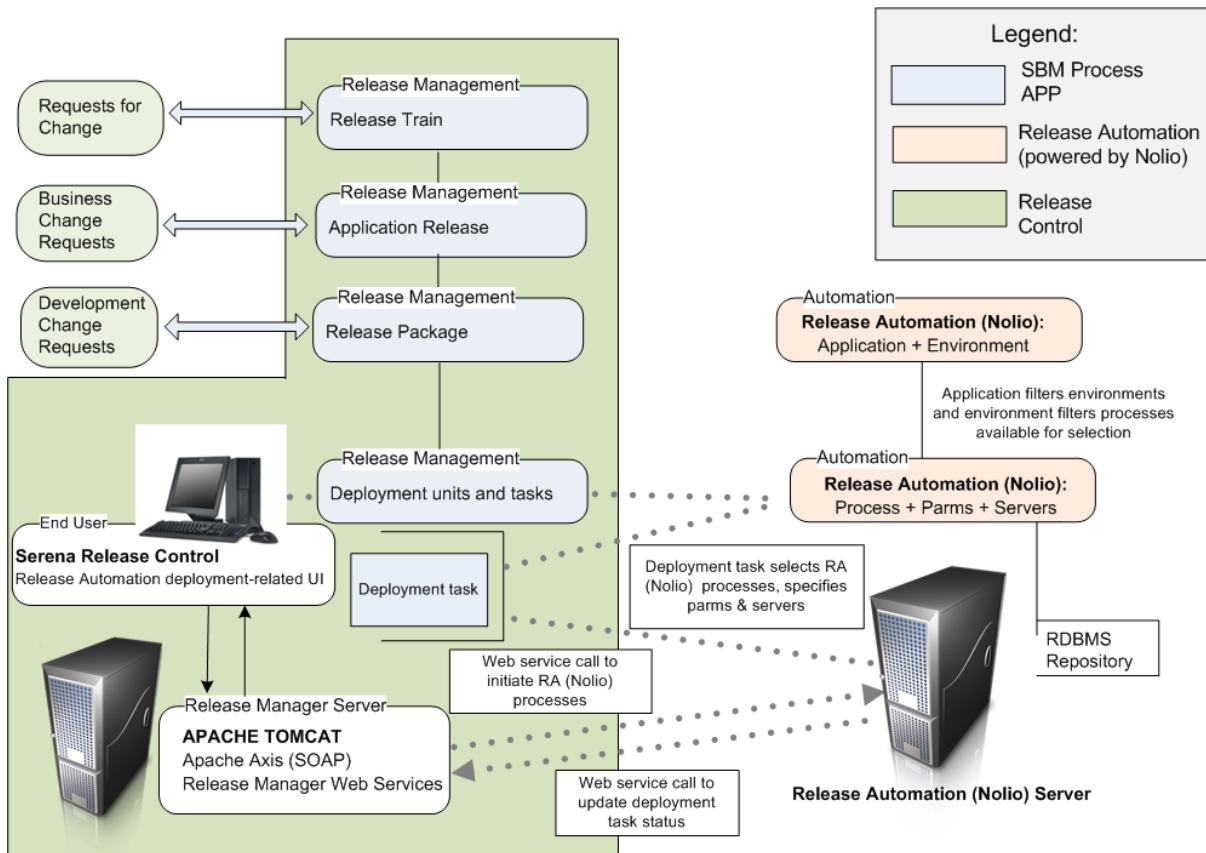
The architecture that supports the Serena Release Automation integration is shown in the following figure:



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>	<p>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:</p> <pre>http://almhost:8080/alm/servlet/ranotification/</pre> <p>You must specify the actual ALM application environment server name for the host; localhost is an invalid value.</p>
<code>ra.server.postDeployMessage</code>	<p>Expressions for the process status message, in the form of:</p> <pre>\${p:request.id}:\${p:finalStatus}</pre>
<code>ra.server.childStatusPutUrl</code>	<p>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:</p> <pre>http://almhost:8080/alm/servlet/ranotification/</pre> <p>You must specify the actual ALM application environment server name for the host; localhost is an invalid value.</p>
<code>ra.server.childStatusMessage</code>	<p>Child status messages, in the form of:</p> <pre>requestId:\${p:parentRequest.id} component:\${p:component.name} resource:\${p:resource.name} status:\${p:status} result:\${p:result}</pre>

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 \[page 113\]](#)
- [Configuring Serena Release Automation Processes and Servers \[page 114\]](#)

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
```

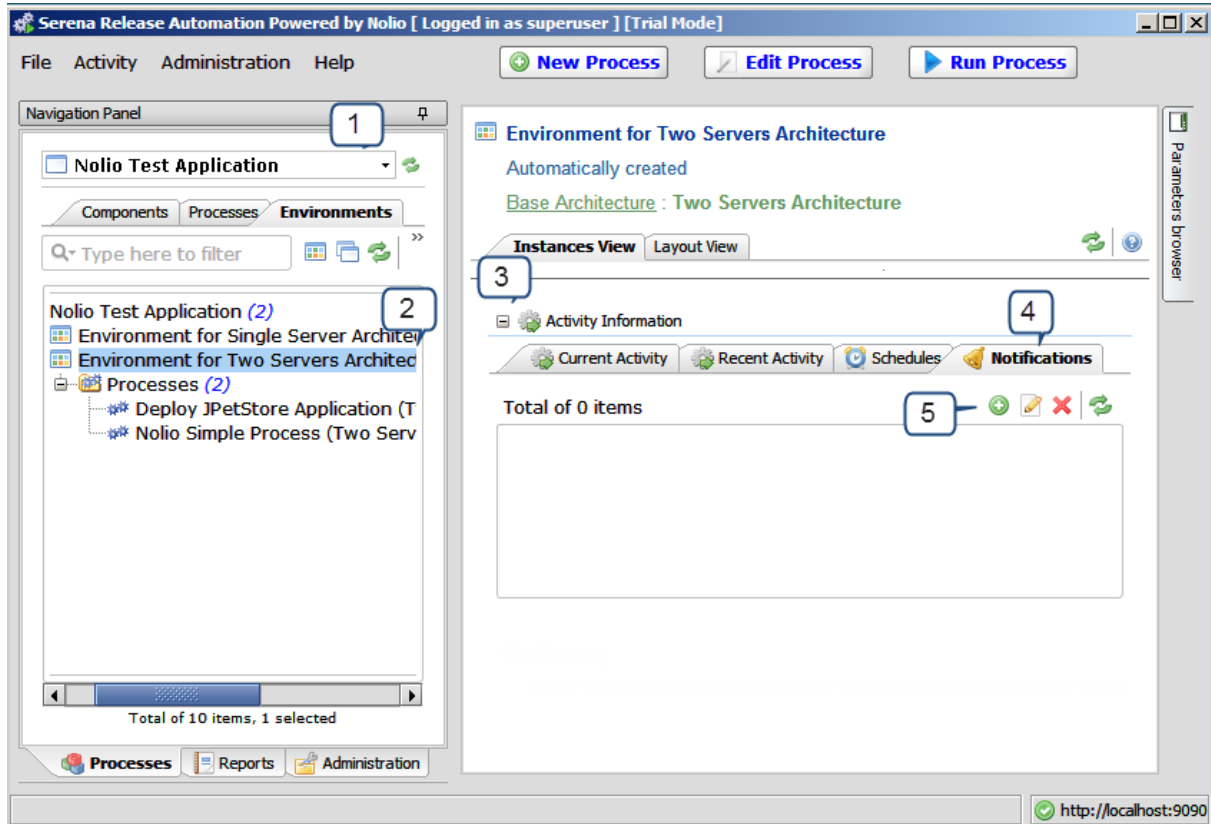
4. Restart the Nolio Server and Nolio Agent services.

IMPORTANT! Make sure to use the exact case given in [Step 3 \[page 115\]](#) 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 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 :
 ...

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 \[page 120\]](#)
- [Configuring Release Automation \(Nolio\) Processes and Servers \[page 120\]](#)

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 \[page 122\]](#)

[Development Manager Provider Overview \[page 132\]](#)

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.

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 \[page 130\]](#).

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 \[page 124\]](#)
2. [Telling Release Manager Which Providers to Use \[page 122\]](#)

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
- 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 \[page 125\]](#)
2. [Telling Release Manager Which Providers to Use \[page 122\]](#)

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 \[page 126\]](#)
2. [Telling Release Manager Which Providers to Use \[page 122\]](#)

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 \[page 127\]](#)
- [Designating DCR Provider Details for SBM Issues \[page 128\]](#)
- [Designating DCR Provider Details for SBM Incidents \[page 128\]](#)

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 \[page 130\]](#)
2. [Telling Release Manager Which Providers to Use \[page 122\]](#)

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 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 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.

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 \[page 44\]](#).

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).</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 \[page 46\]](#).

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. Vault templates are used in integration with Development Manager to automate the process of bringing component sets, or baselines, into Release Manager as Definitive Media. See the following sections for details.

[Vault Request Purpose \[page 139\]](#)

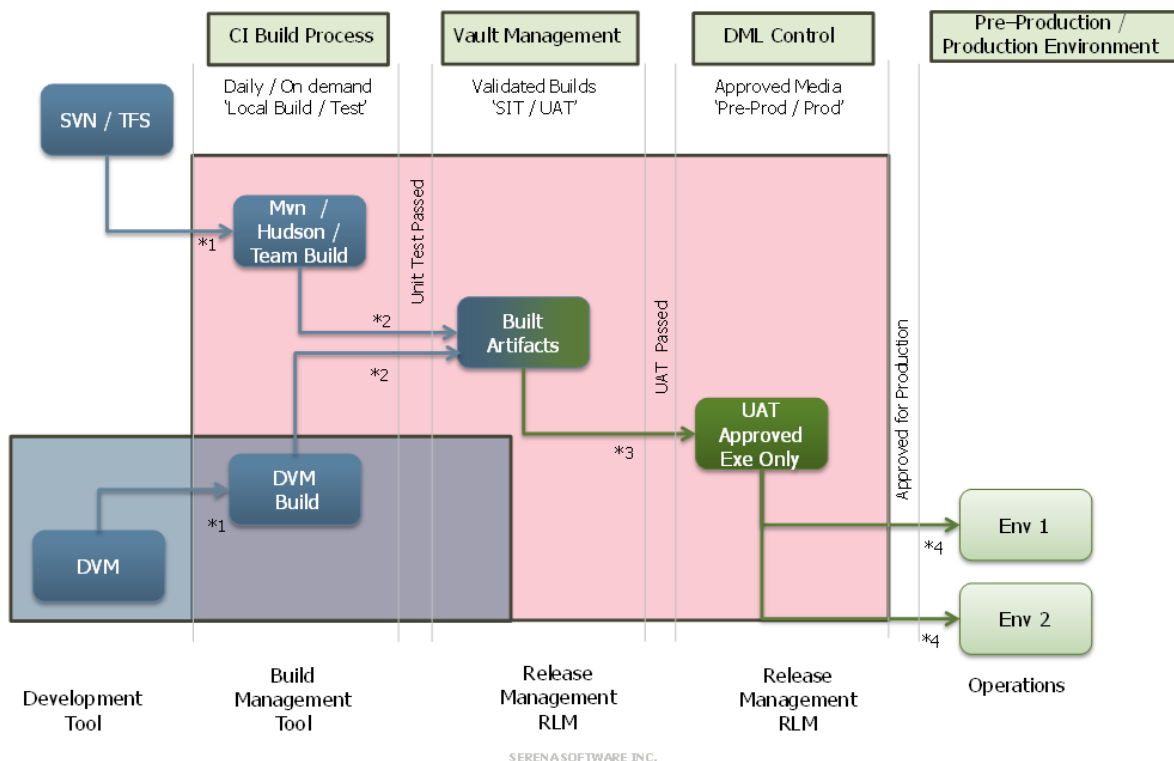
[Vault Request Architecture \[page 140\]](#)

[Vault Template Purpose \[page 140\]](#)

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 management 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 be 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 Overview

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.

Configuring Integrations with Orchestrated Ops Solutions

Serena ALM solutions integrate with Serena Orchestrated Ops solutions to provide a full end-to-end DevOps solution for your organization.

See the following sections for details.

[SSM and Release Manager Integration \[page 141\]](#)

[Demand Manager and Development Manager Integration \[page 144\]](#)

SSM and Release Manager 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

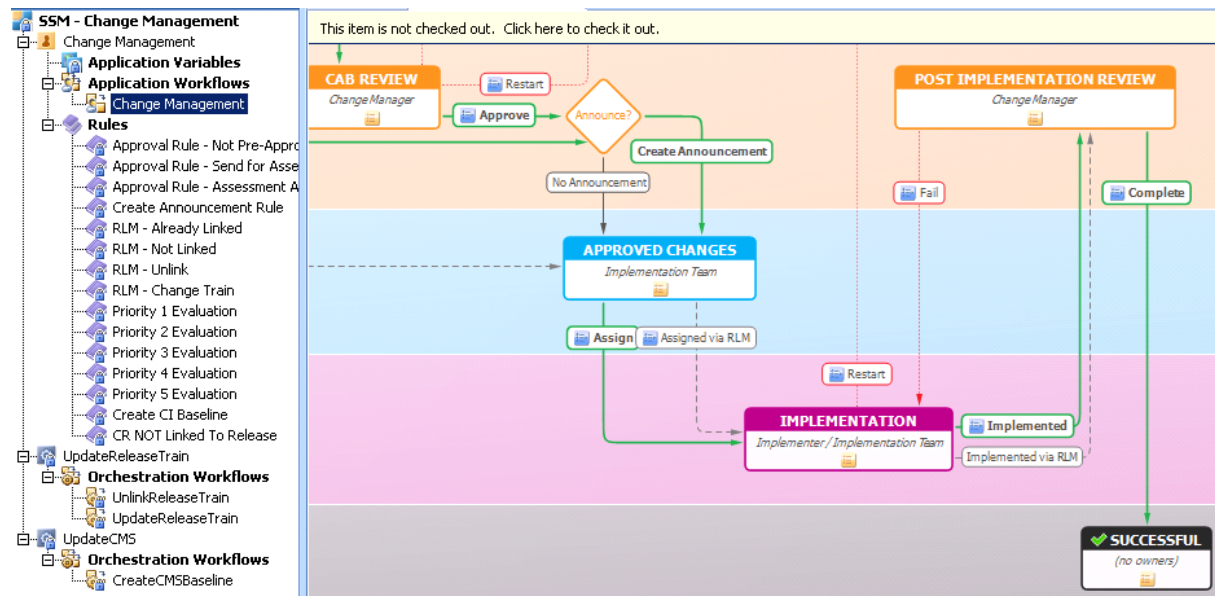
Privilege Category:	Guest
Report Project:	Base Project
Columns to Display:	Official release name
	Production deployment start date
	Production deployment end date
Include items from Sub-projects	(Select)
Use Basic Conditions:	State in Planning
	Production deployment start date = (Query At Runtime)
	Production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

Demand Manager and Development Manager Integration

You can associate Demand Manager work items with Serena Development Manager demand items using the provided features in the solutions.

Demand Manager can be easily integrated with Serena Development Manager if both solutions are installed.

For best results, use Dev Change Requests and the Dev Tasks as work item process apps. These process apps contain the Demand and Time Capture widgets, along with rules that display the widgets to users who have privileges to use Demand Manager.

Development Manager users can then associate their change requests and tasks, which are work items, to demand items, such as those in the Business Requests process app. Users can also record the amount of time spent on each work item. This information is used to calculate "actuals" in Demand Center.

Demand Manager's project linking feature also enables you to create new projects in either Development Manager process app or link existing projects to plans.

To prepare the Dev Change Requests process app for Demand Manager, see the *Serena Orchestrated Ops Getting Started Guide*.

Troubleshooting Serena ALM

This section gives information on troubleshooting issues in Serena ALM.

[Troubleshooting Overview \[page 146\]](#)

[Information from the User Interface \[page 146\]](#)

[Information from the ALM Configurator \[page 147\]](#)

[Information from Log Files \[page 147\]](#)

[Symptoms and Solutions \[page 148\]](#)

Troubleshooting Overview

When you encounter an issue, 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`.
- Known Issues published in the Readme for your solution, such as the Development Manager Readme and the Release Manager Readme.
- Readme Known Issues and troubleshooting information for integrating products and solutions.
- The Serena Support website Knowledgebase. You must be logged on the Support website to view Knowledgebase solutions.

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 the UI 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 \[page 148\]](#) for more assistance in these situations.

Activity Log

In Release Manager, 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

In Release Manager, 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

In Release Manager, 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.

In Development Manager, to see the change history of an Agile Epic, Story, or Task, a Dev Change Request, or a Dev Change Task, click the **History log** 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 ALM 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 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 \[page 148\]](#)
- [Snapshot Promotion Errors \[page 149\]](#)
- [Cannot log into Serena Release Manager \[page 150\]](#)
- [Cannot Log into Serena ALM Configurator \[page 150\]](#)
- [User Interface Display Issues \[page 151\]](#)
- [Create Release Fails with a Check Uniqueness Error \[page 152\]](#)
- [Unable to update the landing page layout \[page 152\]](#)
- [Matches Not Found for Selections \[page 152\]](#)
- [No Change Package Data Displayed in ZMF Deployment Tasks \[page 153\]](#)
- [Release Package Deployment Fails \[page 153\]](#)
- [Slow Response Time \[page 154\]](#)

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 generally do 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.

Warning messages similar to one of the following can 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 table fields referenced in process apps that have not yet been promoted. SBM tracks those missing references and as the process apps that are with the fields that are referenced are promoted, the missing references are resolved.

Cannot log into Serena ALM

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

Verifying the Web Services Connection

To verify the Serena ALM 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 ALM Web services are listed and in active status as follows:

- Release Manager
 - **RLMUtilService**
 - **ReleaseRequestService**
 - **DeploymentAutomationService**
 - **VaultService**
- Development Manager
 - **DVMUtilService**

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. See the Serena Support Knowledgebase for more information.

User Interface Display Issues

If the Serena ALM 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 in SBM.

Examples:

- If you added Release Manager 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 Release Manager 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.
- If you are not seeing the Development Manager Agile interface when signed on as a user who should have an Agile role, you may not have all of the group and role assignments, privileges, and associated XML file entries configured properly.

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 ALM 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 in Release Manager 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.

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.

For information on configuring Serena ALM to use a different port number, please see the Serena Support Knowledgebase solution S139664. You must be logged on the Support website to view Knowledgebase solutions.

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 the 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 \[page 31\]](#) 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 \[page 122\]](#).

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.

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 \[page 156\]](#)

[Development Manager Workflows \[page 170\]](#)

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 \[page 156\]](#)

[Release Train Workflow \[page 158\]](#)

[Application Release Workflow \[page 159\]](#)

[Release Package Workflow \[page 160\]](#)

[Deployment Task Workflows \[page 163\]](#)

[Deployment Process Template Workflow \[page 168\]](#)

[Environment Workflow \[page 169\]](#)

[Vault Request Workflow \[page 169\]](#)

[Vault Template Workflow \[page 170\]](#)

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.

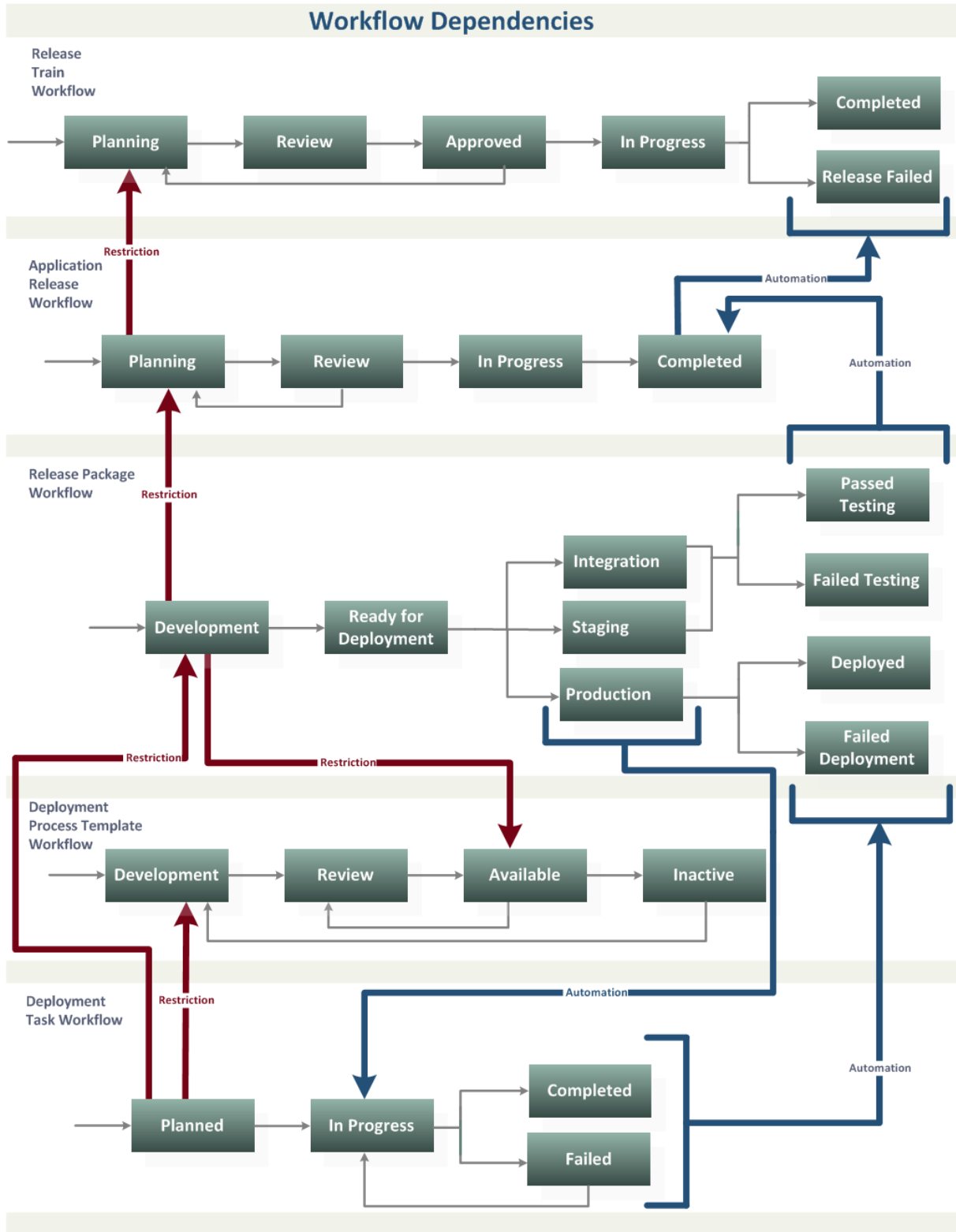
Workflows for Deployment Process Templates are standalone, although deployment process templates populate the deployment tasks for release packages and in that respect have a link to

the workflows for both Deployment Tasks and Release Packages. Workflows for environments are completely standalone by default.

Workflow Automations

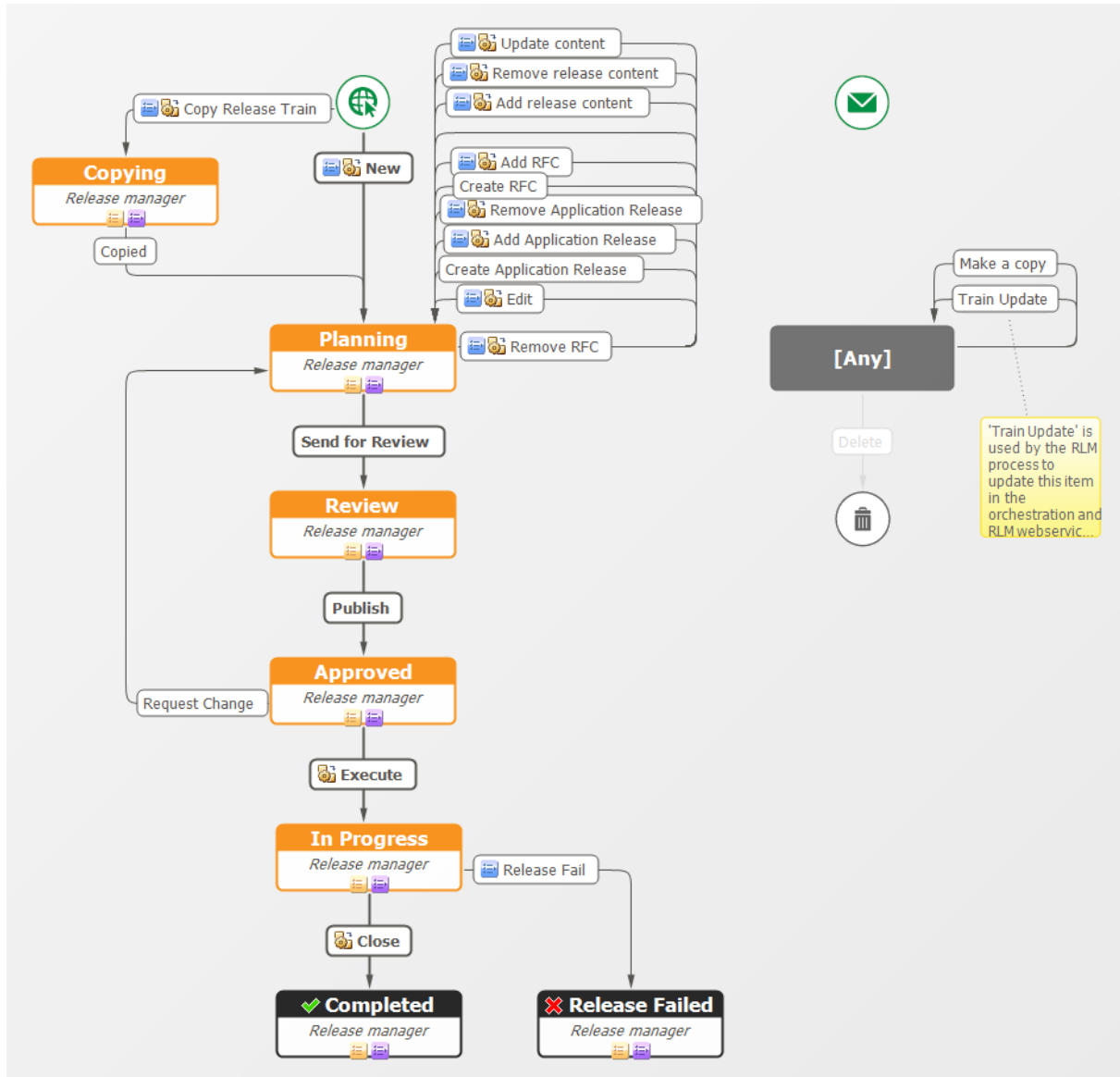
Many of the workflows are moved to completion states automatically when associated objects are complete. The automation is usually initiated at the lowest level of association and moves up the hierarchy as each level of release activities are completed. For example, after all deployment tasks for a release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



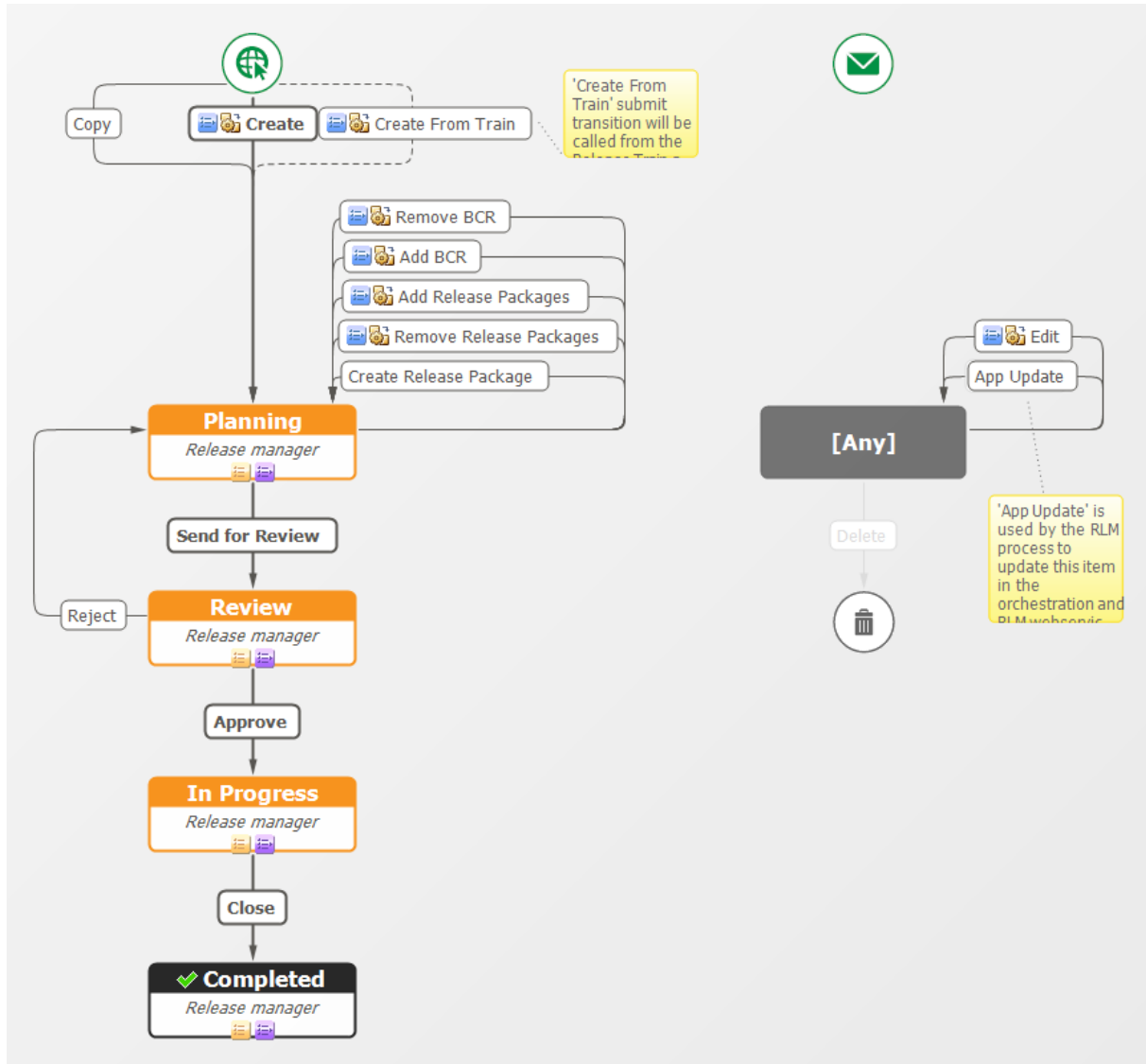
Release Train Workflow

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.

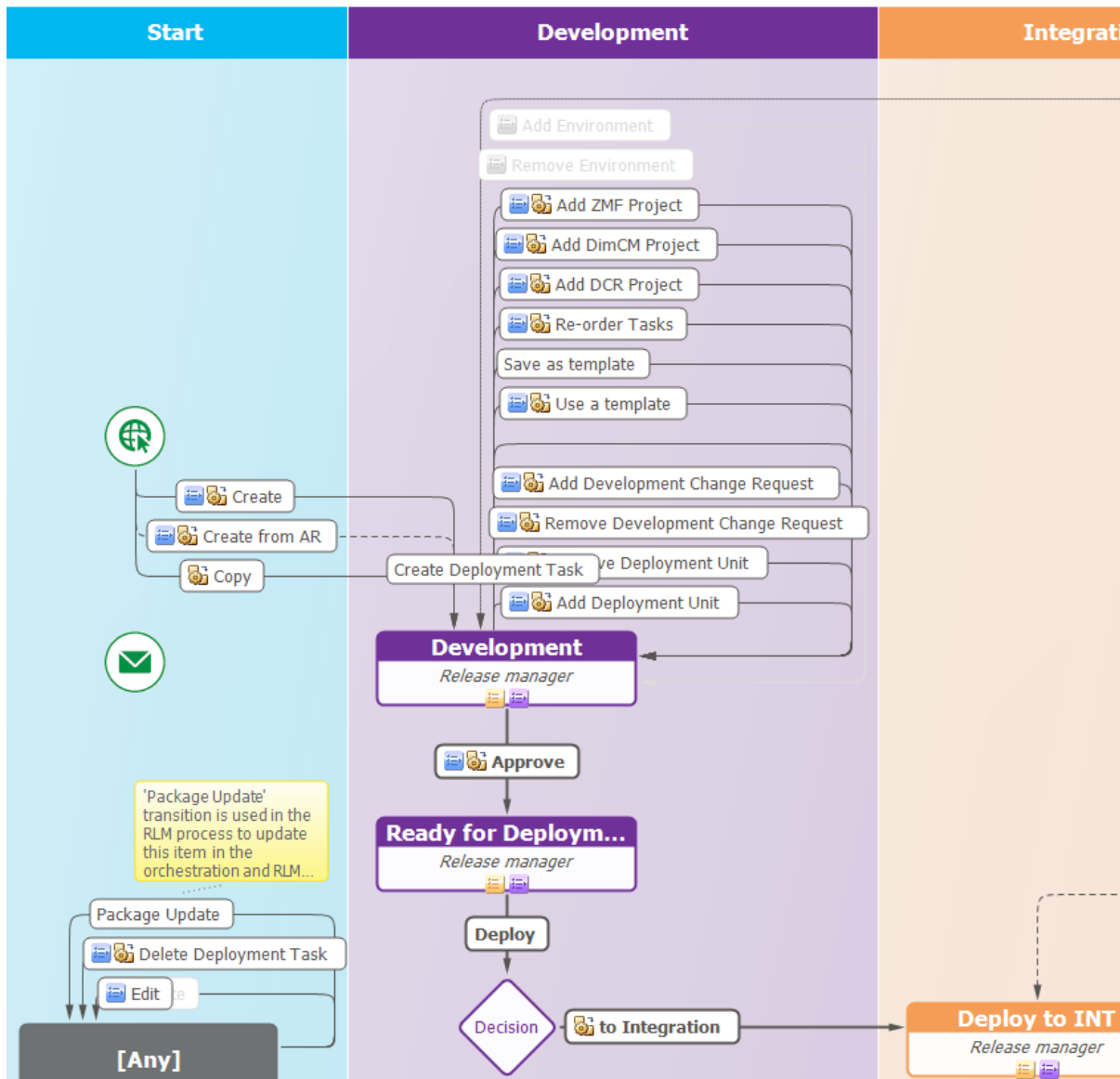


Release Package Workflow

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.

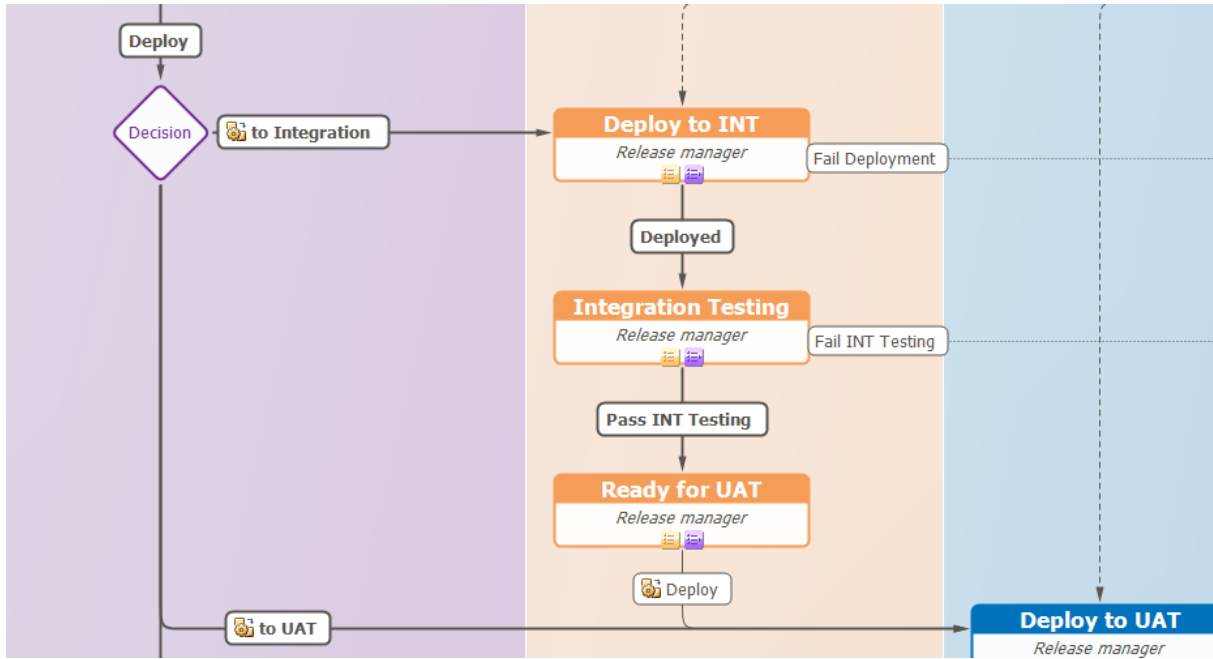
Start and Development States

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.



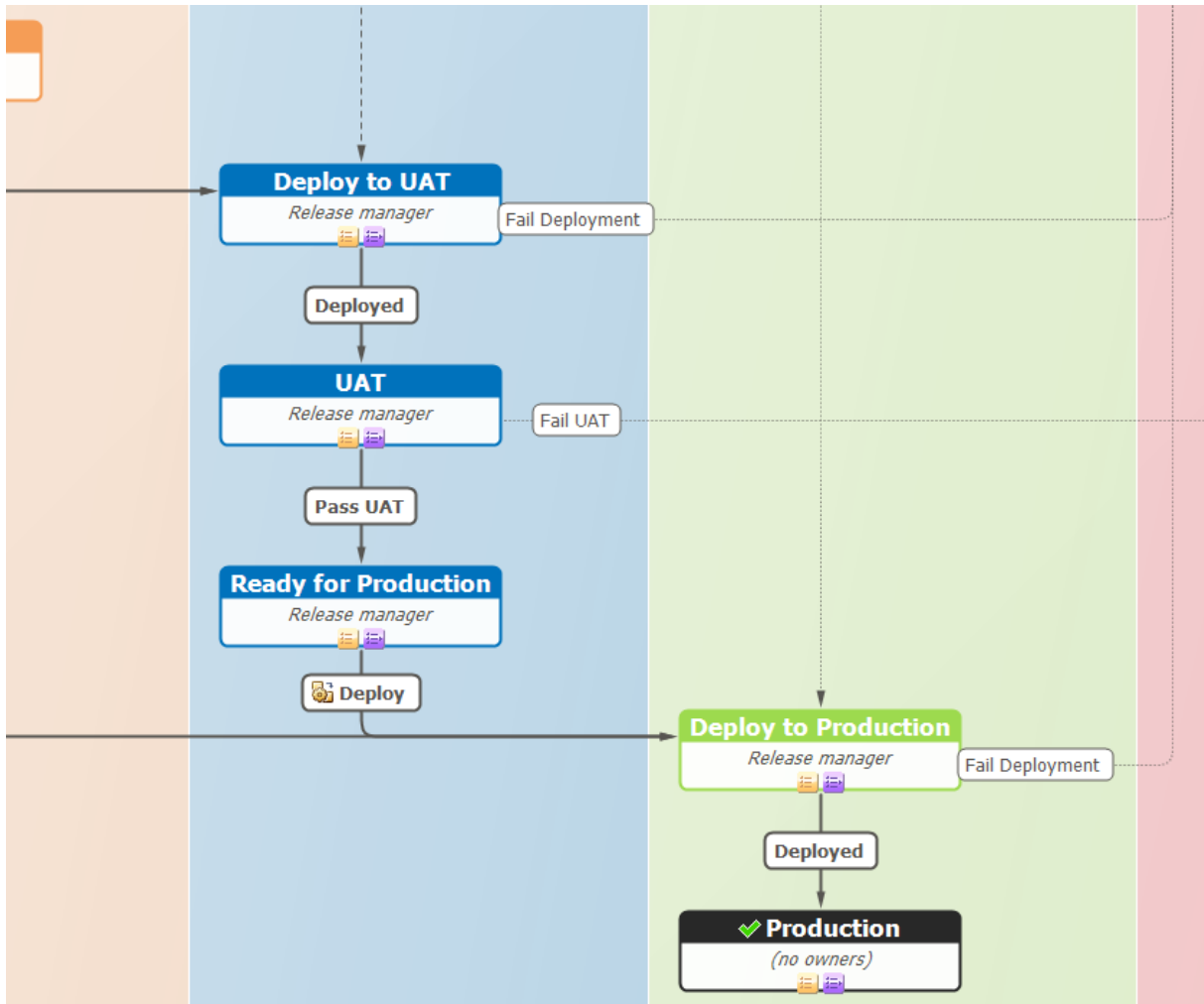
Integration State

The Integration state swimlane of the Release Package workflow in the default version of Serena Release Manager is shown in the following figure.



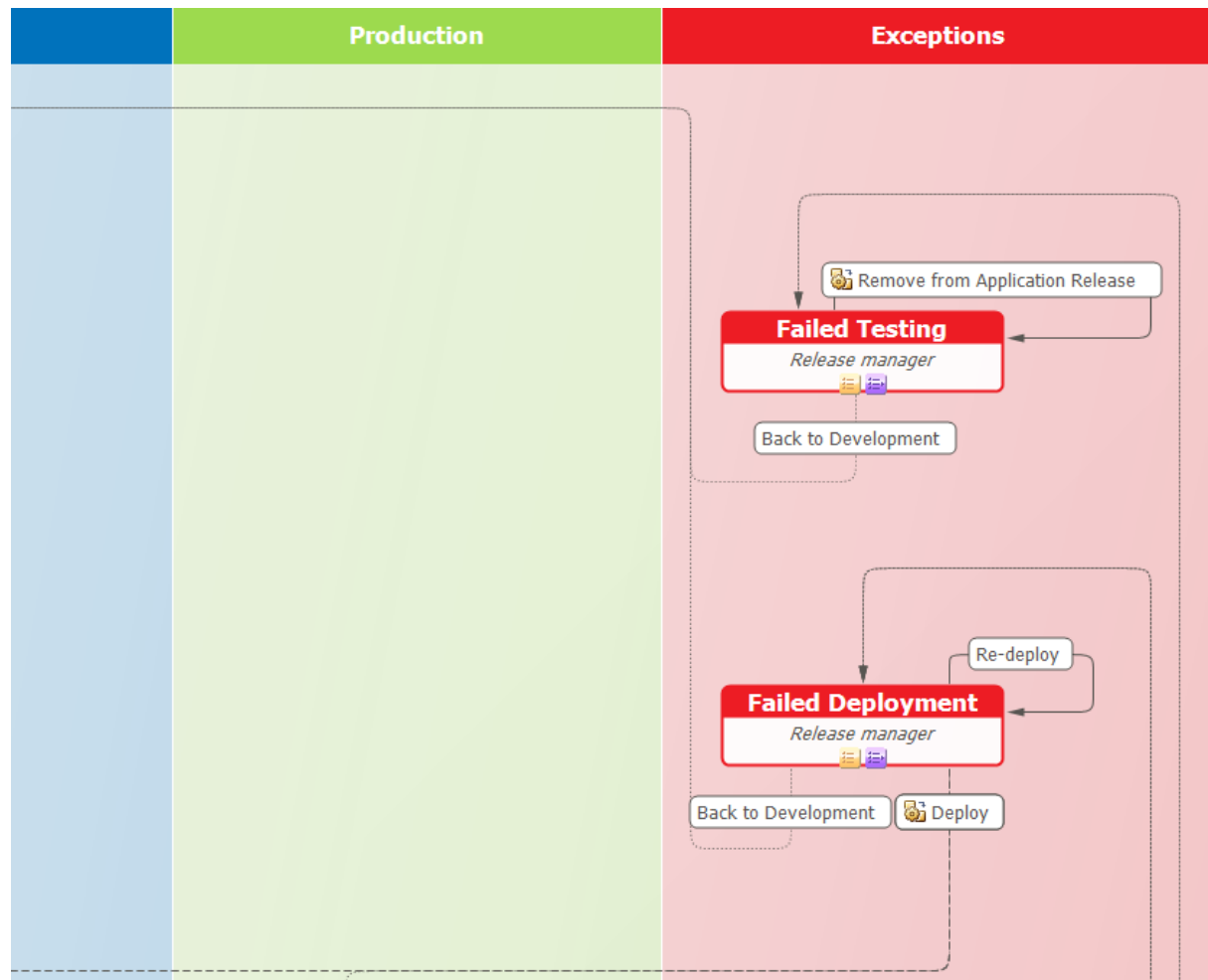
Staging and Production States

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.

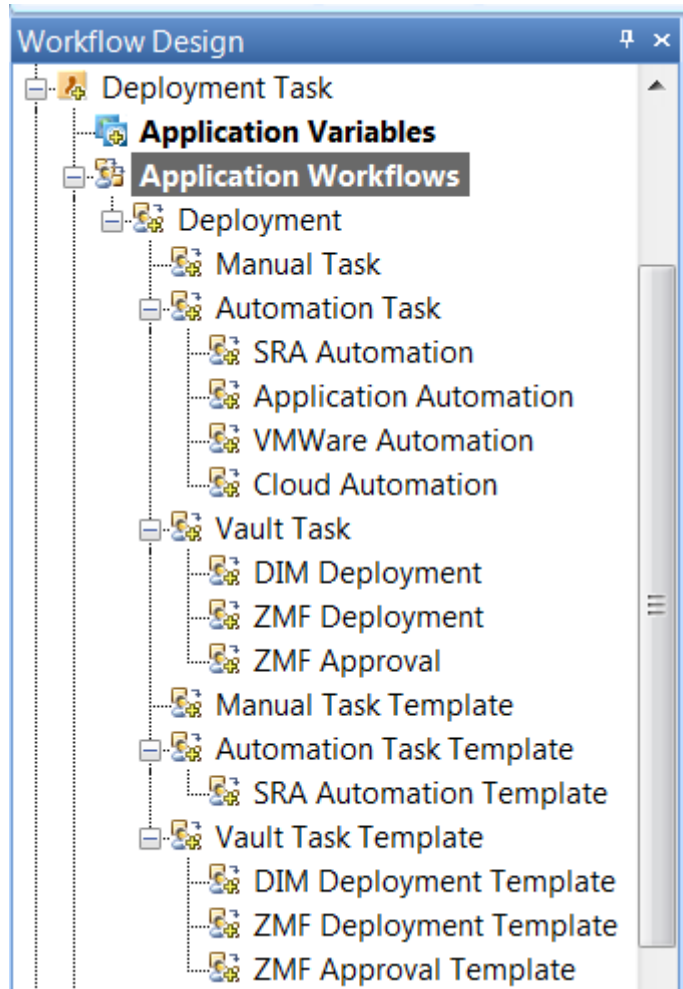


Deployment Task Workflows

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

Deployment Task Hierarchy

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

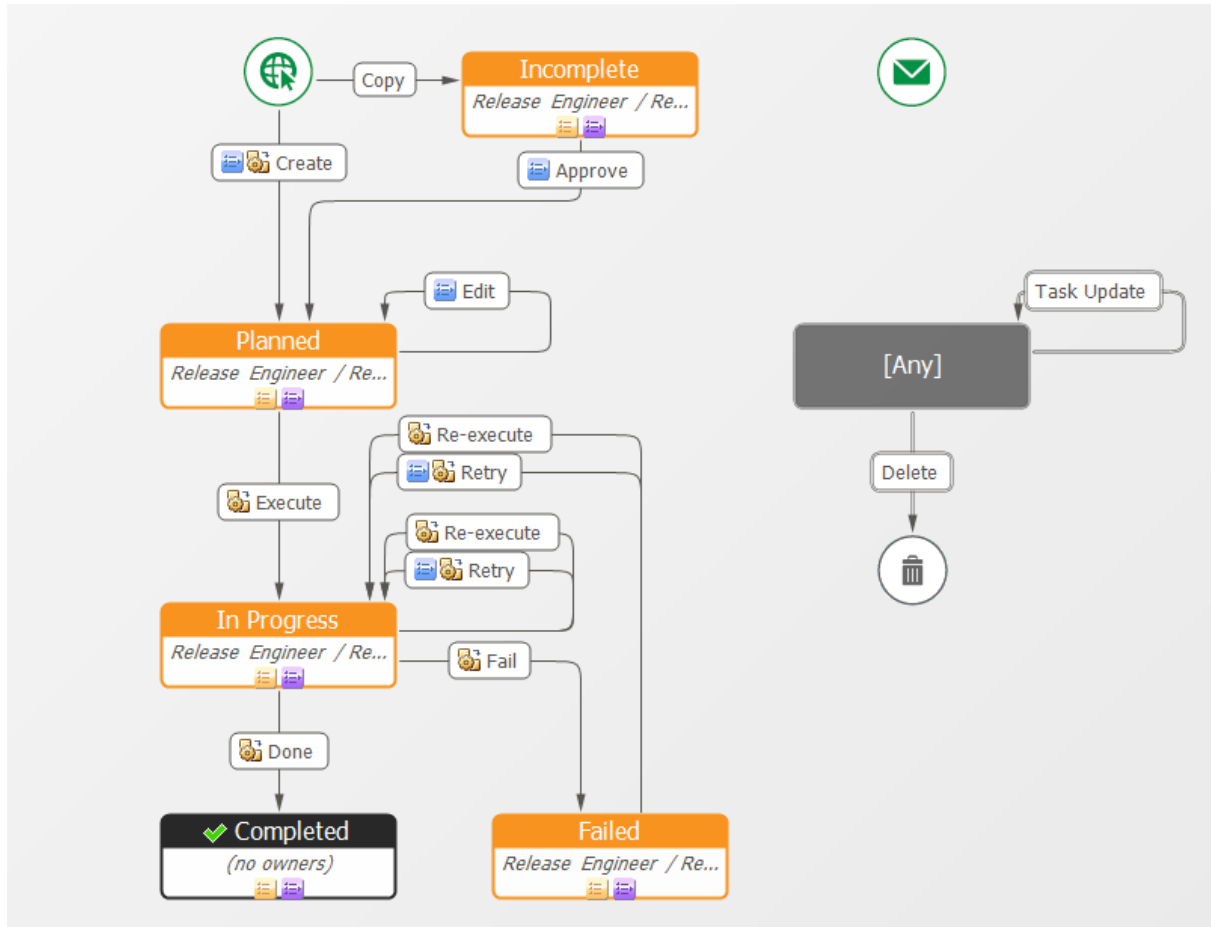


Automation Deployment Task Workflow

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

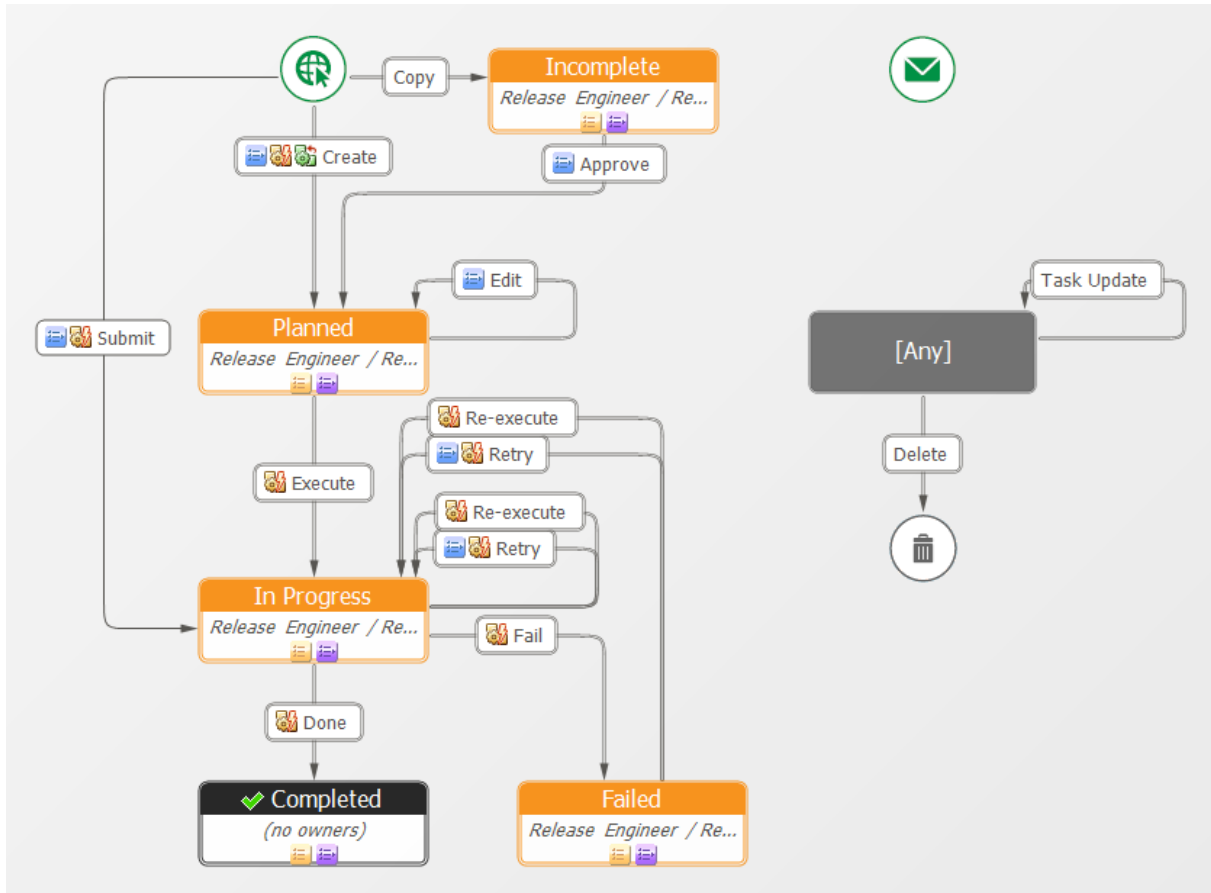
Main Automation Task Workflow

The main Automation Task workflow is shown in the following figure.



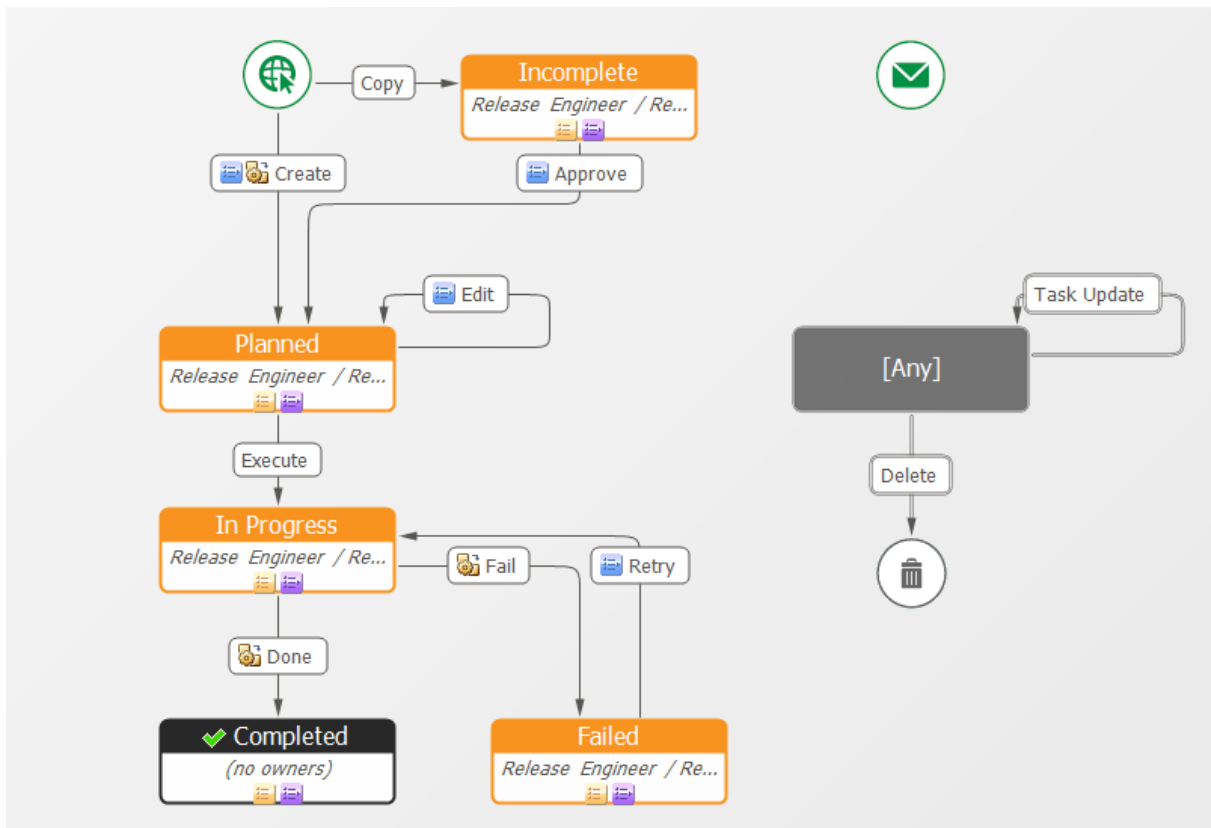
Release and Application Automation Task Workflows

The SRA Automation, Application Automation, VMWare Automation, and Cloud Automation sub-workflows inherit from the main workflow for Automation Task, but have an extra Submit transition with associated forms and orchestration workflows.



Manual Deployment Task Workflow

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

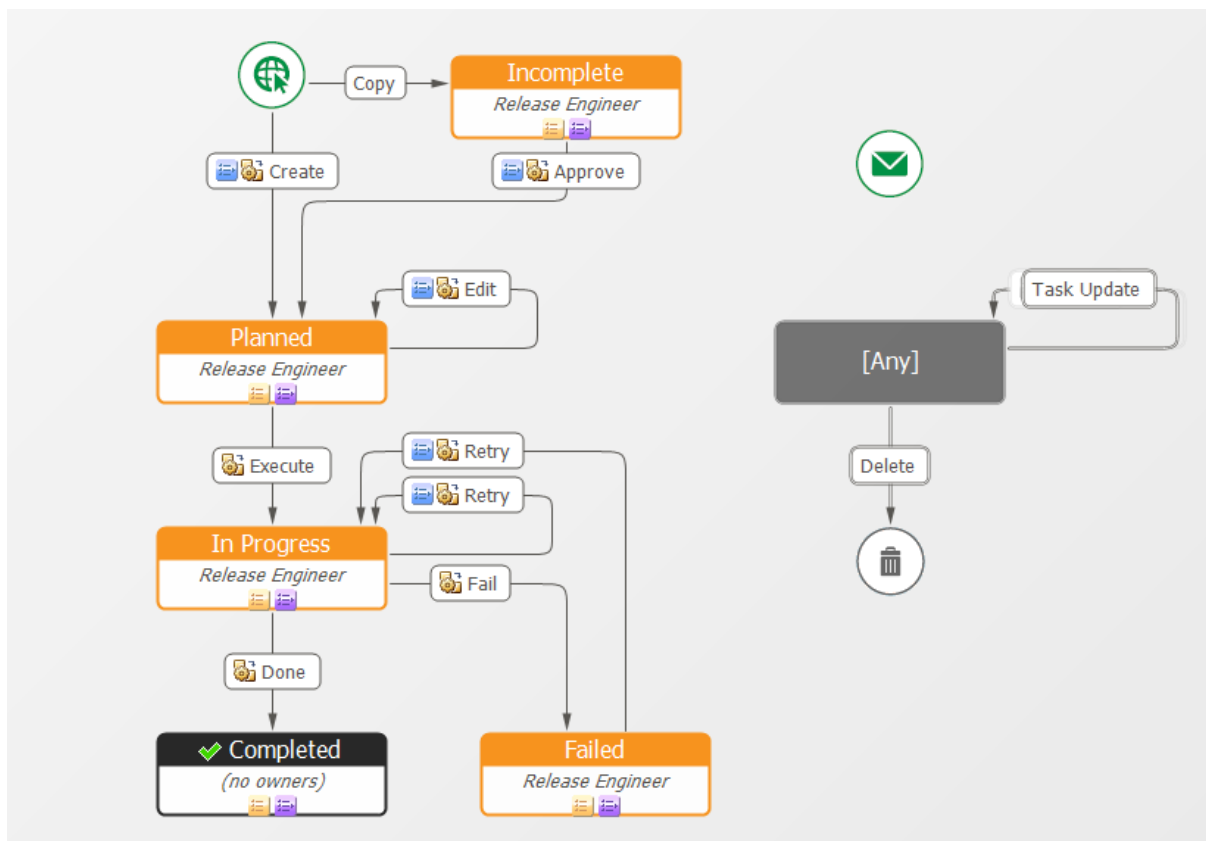


Vault Deployment Task Workflows

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

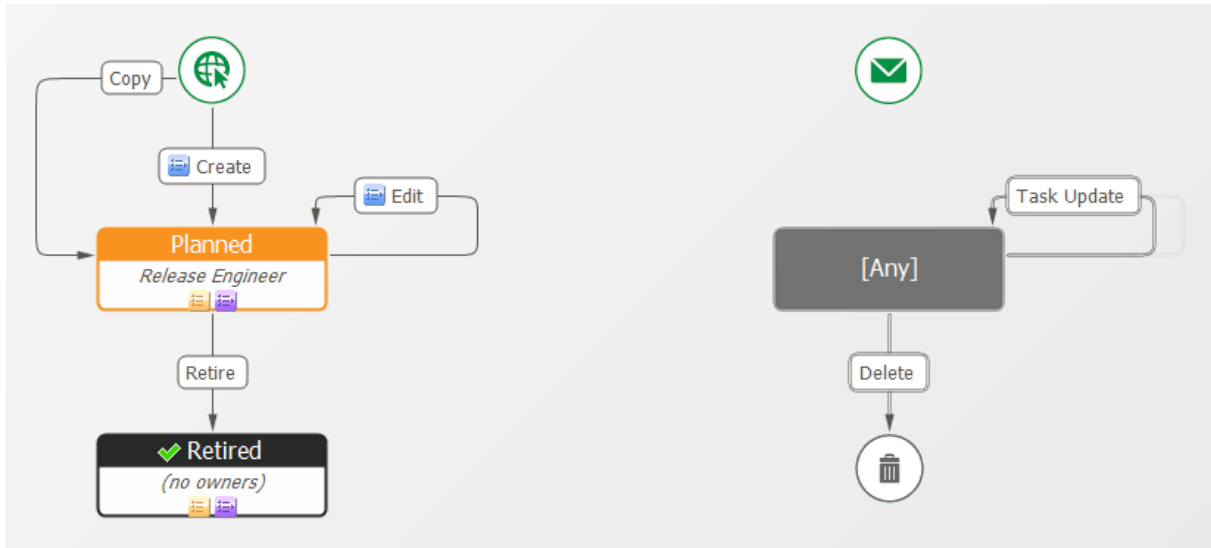
Main Vault Task Workflow

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.



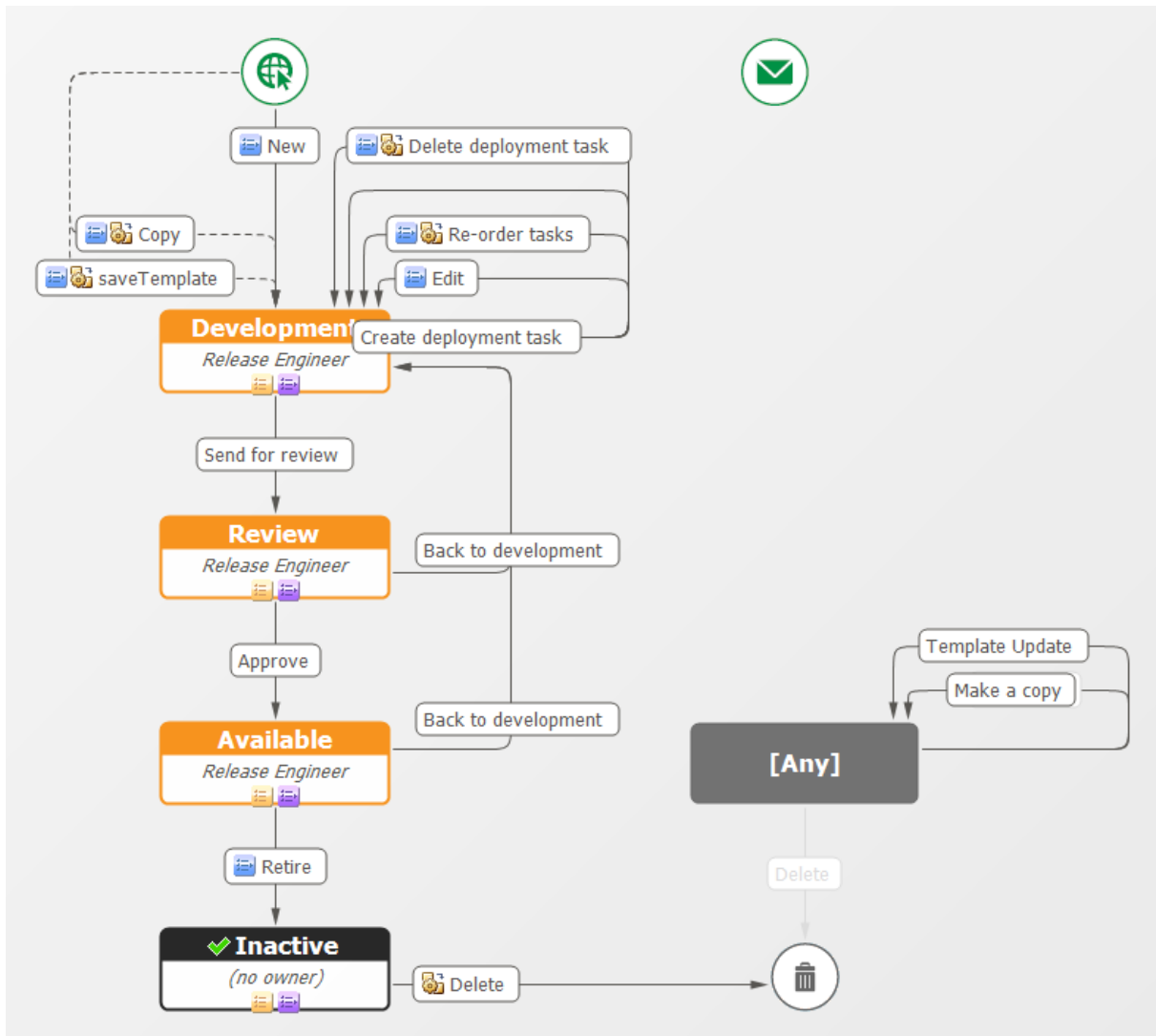
Main Vault Task Template Workflow

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.



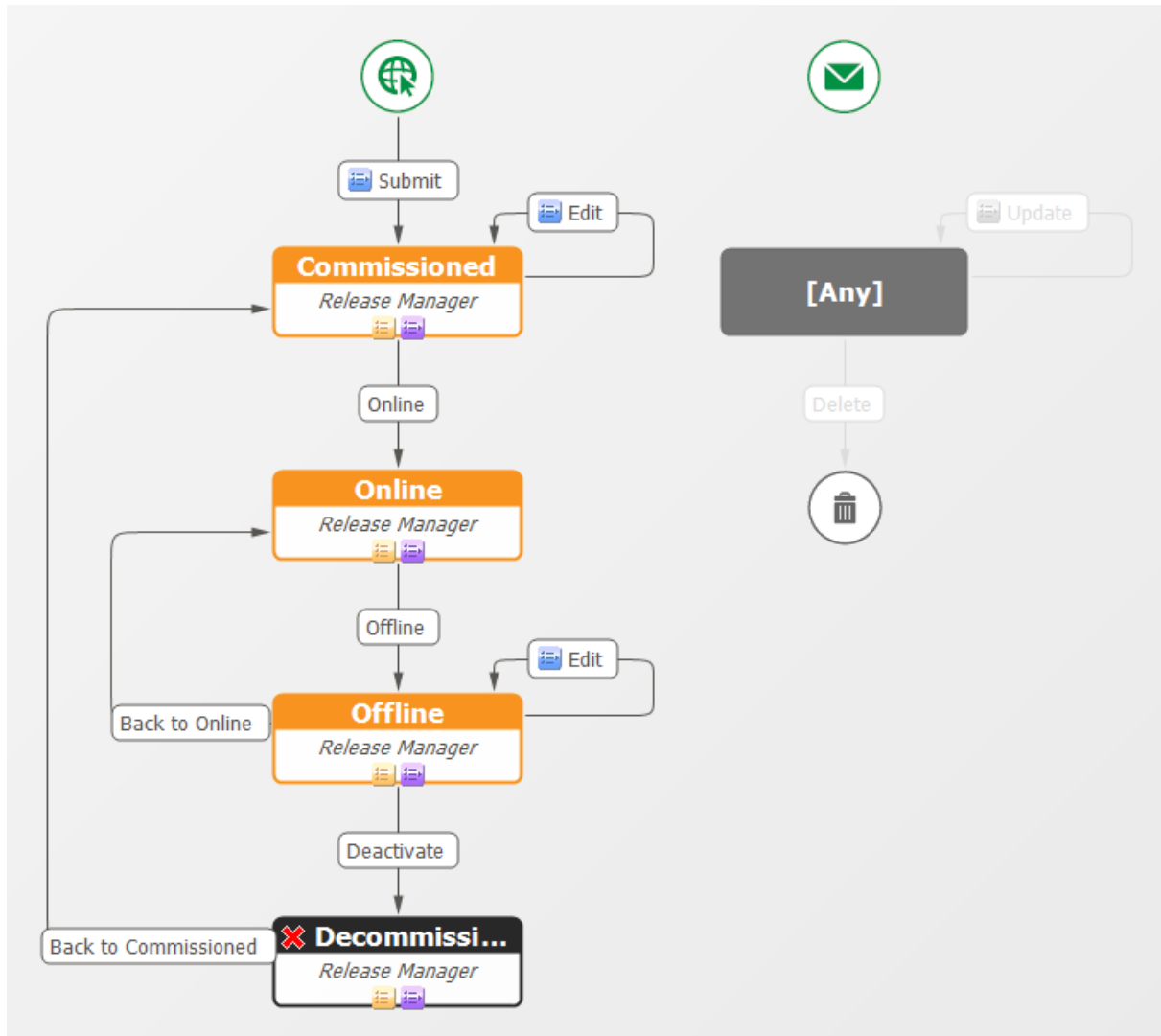
Deployment Process Template Workflow

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



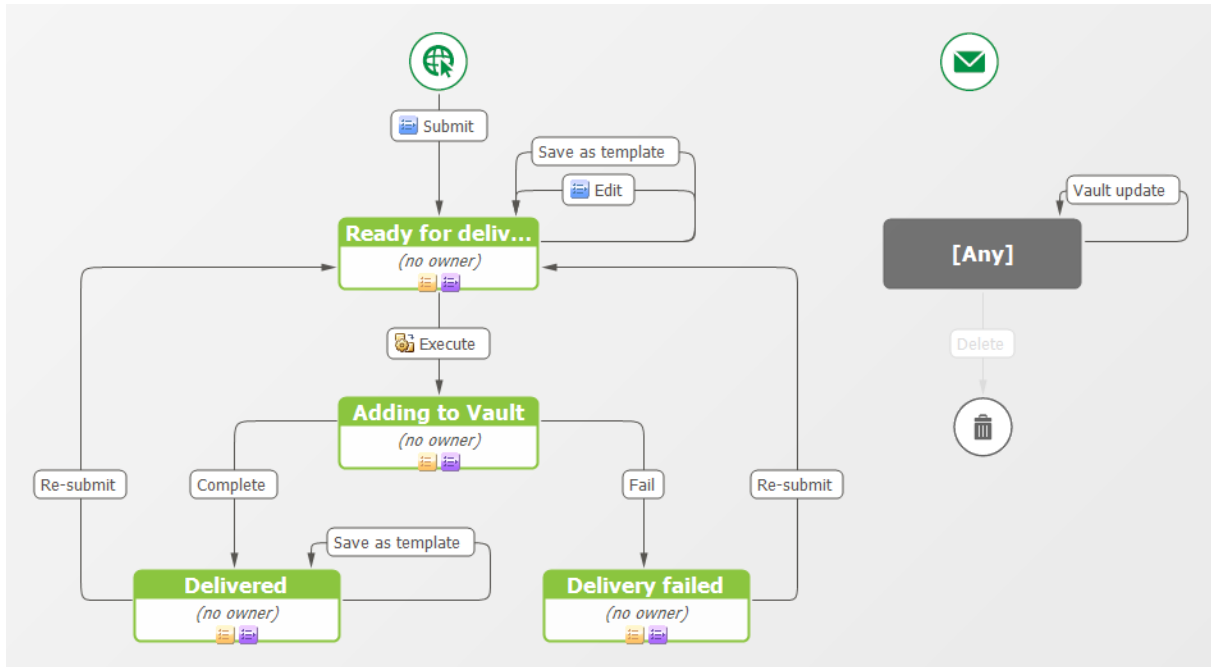
Environment Workflow

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



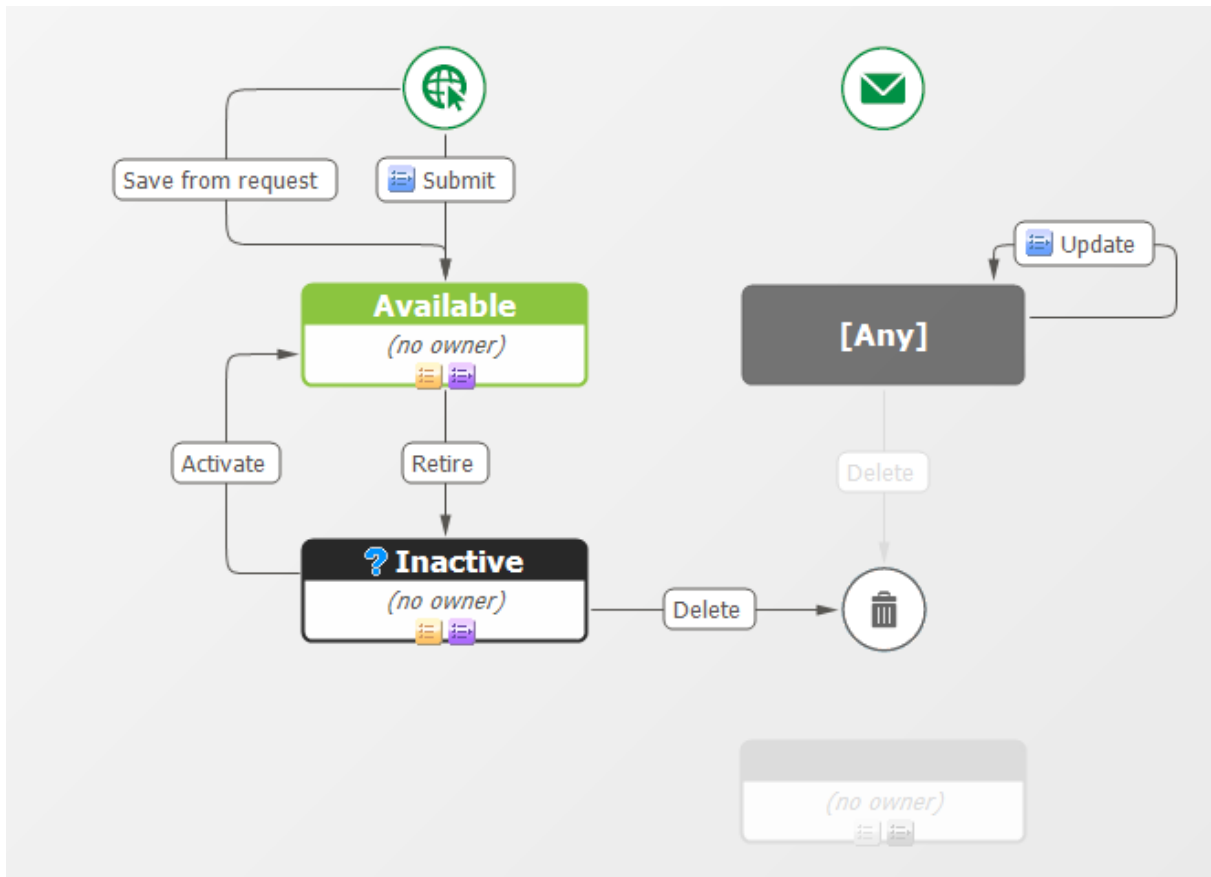
Vault Request Workflow

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



Vault Template Workflow

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



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 \[page 171\]](#)

[Configuration Files on the Agile Server \[page 174\]](#)

[Configuration Files on the Dimensions CM Server \[page 174\]](#)

[Configuration Files on the Release Automation \(Nolio\) Server \[page 175\]](#)

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 ALM 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 41] .
bcr.properties	Provides provider definitions for Business Change Requests (BCRs). See Configuring Access to Business Change Requests [page 125] .
bcr-connection.properties	Provides connection information for Business Change Requests (BCRs). See BCR Provider Connection Information [page 42] .
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 126] and Configuring Access to Deployment Units [page 129] .
dm-client-connection.properties	Provides connection information for Dimensions CM. See Dimensions CM Client Connection Information [page 44] .

dvmzmf_packages.properties	Provides filtering information for accessing ChangeMan ZMF information from Development Manager. See Development Manager Dev Task Provider Details [page 133] .
itsm.properties	Provides provider definitions for ITSM (for RFCs). See Configuring Access to Requests for Change [page 123] .
itsm-connection.properties	Provides connection information for ITSM (for RFCs). See RFC Provider Connection Information [page 43] .
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 147] .
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 117] .
nolio-client-connection.properties	Provides connection information for Serena Release Automation, powered by Nolio. See Serena Release Automation (Nolio) Client Connection Information [page 45] .
nolio-client-queries.properties	Provides filtering information for Serena Release Automation, powered by Nolio. See Configuring Release Automation (Nolio) Communication in Release Manager [page 117] .
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 122] .
ra-client.properties	Provides client-specific information for Serena Release Automation. See Configuring Release Automation Communication in Release Manager [page 112] .
ra-client-connection.properties	Provides connection information for Serena Release Automation. See Serena Release Automation Client Connection Information [page 44] .

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> <p>rlm.config.service.authdUsers=<list of user IDs delimited by commas></p> <p>This defines the list of users who can access the Serena ALM Configurator through SSO. See Configuring Access to the Configurator [page 40].</p>
rlm-sbm.properties	<p>Sets values for the Serena ALM Web services used for Serena Release Manager. Use this file for setting maximum associations for Release Manager objects.</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	<p>Provides filtering information for accessing ChangeMan ZMF information from Release Manager. See Configuring Access to Deployment Units [page 129].</p>
sbm.properties	<p>Sets values for the Serena ALM Web services.</p>
sbm_incidents.properties	<p>Provides provider definitions for SBM Incidents, typically associated with BCRs. See Designating the Details for Each BCR Provider [page 125].</p>
sbm_issues.properties	<p>Provides provider definitions for SBM Issues, typically associated with DCRs. See Designating the Details for Each DCR Provider [page 126].</p>
sbm-client-connection.properties	<p>Provides connection information for SBM. See SBM Application Connection Information [page 41].</p>
zmf-client.properties	<p>Provides client-specific information for ChangeMan ZMF. See Configuring ZMF Communication in Release Manager [page 106].</p>
zmf-client-connection.properties	<p>Provides connection information for ChangeMan ZMF. See ZMF Client Connection Information [page 46].</p>

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 106].</p>

Files in the Agile Classes Folder

Configuration files for the Agile functionality included with Development Manager are located under the Serena Common Tomcat home folder in the `alm#agile\WEB-INF\classes` folder. For example:

```
..\Program Files\Common\tomcat\6.0\webapps\alm#agile\WEB-INF\classes
```

A partial list of the Agile option configuration files and purposes are given in the following table.

File Name	Purpose
<code>config.properties</code>	Use this file to specify the Agile option general configuration properties. See General Agile Configuration [page 72] .
<code>dvm-project.properties.xml</code>	Use this file to map the Agile option project and group objects to SBM projects and groups. See Configuring the Project Context Properties [page 78] and Creating and Assigning the Development Manager Users [page 67] .
<code>import_mssql.sql</code>	Use this file to run the Agile database upgrade script for Microsoft SQL Server databases. See Upgrading the Agile Database [page 25] .
<code>import_oracle.sql</code>	Use this file to run the Agile database upgrade script for Oracle databases. See Upgrading the Agile Database [page 25] .
<code>itemhierarchy.properties.xml</code>	Use this file to specify the Agile item type names and hierarchy. The type names used here must match those in the <code>story.properties.xml</code> and <code>task.properties.xml</code> files. See Configuring the Item Hierarchy Properties [page 78] .
<code>sbm-client-connection.properties</code>	Use this file to specify the Agile option SBM application and administrative server connection information. See Agile Connection Information [page 73] .
<code>story.properties.xml</code>	Use this file to map the Agile properties to your custom epic and story process app properties. See Configuring Epic and Story Properties [page 75] .
<code>task.properties.xml</code>	Use this file to map the Agile properties to your custom task process app properties. See Configuring_Task_Properties.htm [page 76] .

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
<code>dm.cfg</code>	<p><code>..\Program Files\Dimensions 12.2\CM</code></p> <p>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.</p>

File Name	Location and Purpose
ALF_EVENTS_CONFIG.XML	<p>..\Program Files..\Dimensions\12.2\CM\dfs</p> <p>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 83]</p>

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	<p>..\Program Files\Serena\Serena Release Automation\conf</p> <p>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 115].</p>