SERENA

SERENA® RELEASE CONTROL

Getting Started Guide

Serena Proprietary and Confidential Information

Copyright © 2011-2016 Serena Software, Inc. All rights reserved.

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

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

License and copyright information for 3rd party software included in this release can be found on the product's news page at http://support.serena.com/ProductNews/default.aspx and may also be found as part of the software download available at http://support.serena.com.

Trademarks

Serena, Dimensions, ChangeMan, Comparex, and StarTool are registered trademarks of Serena Software, Inc. The Serena logo, PVCS, TeamTrack, License Manager and Composer 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, Suite 150, San Mateo, CA 94404.

Part number: Serena Release Control 6.1

Publication date: 2016-06-15

Table of Contents

Chapter 1: Welcome to Serena Release Control 5

About This Document 5

Chapter 2: About Serena Release Control 7

Terminology 7

Release Control Objects 8

Release Control Participants 9

A Typical Release Control Flow Using Release Trains 9

A Typical Flow Using Release Packages 10

Key Metrics 10

Chapter 3: About the Solution Process Apps 15

About the Release Train Process App 15

About the Release Package Process App 17

About the Deployment Path Application 18

About the Approvals Process App 20

About the Release Data Process App 22

About the Environment Process App 22

About the Task Templates Process App 24

About the Manual Deployment Task Process App 25

About the Sample Request Process App 26

Chapter 4: Configuring Release Trains 27

Creating Release Trains 27

Creating Approval Items 29

Approving Release Items 31

Releasing Release Trains 33

Chapter 5: Configuring and Deploying Release Packages 35

Creating Release Packages 36

Designating Release Packages as Parents 38

Adding Child Release Packages 39

Working with Child Release Packages 39

Adding Requests 40

Reloading Request Data 41

Adding Deployment Units 41

Reloading Deployment Unit Data 41

Adding Deployment Tasks 42

Creating Task Templates 43

Validating Deployment Tasks 44

Scheduling Deployment 45

Deploying Release Packages 45

Adding Release Packages to Release Trains 47

Glossary 49

Chapter 1: Welcome to Serena Release Control

Release Control provides an enforced, repeatable process for all of your organization's releases, from distributed and mainframe applications to cloud and mobile apps. Release Control brings visibility, traceability, and compliance across the business, development, and operations teams in your enterprise.

Release Control is powered by Serena® Business Manager. Highlights include:

- **Planning, tracking, and control for releases**

  Streamline the process of assembling, managing, approving, and tracking releases.

- **Proactive dashboards, timeline, calendars, and reports**

  View immediate updates on all release activities, from executive-level status to deployment task details. Shared calendars and a configurable timeline provide visibility to your entire organization.

- **Managed deployments and environments**

  Streamline the transfer of releases between teams, reduce the build-to-deployment time for each release stage, and enforce deployment processes.

- **DevOps collaboration and coordination**

  Ensure compliance with release policies.

- **Enterprise-level availability, security, and scalability**

  Bring governance and reliability to your release activities and capture subject matter expertise for subsequent sharing.

- **Extensibility and best practices examples**

  Through configurable best practice example processes and open plugin architecture you can tailor the solution to fit your organization.

About This Document

This document gives an overview of the Serena Release Control SBM solution and gives basic usage information based on the default implementation.

For a list of other documentation and resources available for Release Control, refer to the Release Control Documentation Roadmap.

Chapter 2: About Serena Release Control

Release trains and release packages help you control your release processes.

Note: Users access Release Control through the **Release Ctrl** application group in Serena Work Center that includes Release Control. The classic SBM shell is not supported for Release Control. For details on making this application group visible to users, refer to the *Serena Release Control Installation and Setup Guide*.

The following sections define the objects and describe participants, relationships, and typical process flows in the default Release Control implementation.

- Terminology [page 7]
- Release Control Objects [page 8]
- Release Control Participants [page 9]
- A Typical Release Control Flow Using Release Trains [page 9]
- A Typical Flow Using Release Packages [page 10]
- Key Metrics [page 10]

Terminology

Before you get started with Release Control, here are some terms you should know.

- **Release Packages**

  Release packages manage portions of IT or service infrastructure normally built, deployed, tested, and released together. Release packages define the set of changes to be released and drive the release processes.

- **Release Trains**

  Release trains manage a set of release packages from concept approval through scheduling and scoping, construction, verification, and deployment.

- **Approvals**

  Approvals may be required to allow a release train to proceed to the final release stage; to validate that a release train may exit a stage based on stage gate exit criteria; and to confirm that pre-defined milestones for release trains have been reached by a certain date.

- **Requests**

  Requests can be submitted that represent work items in response to a business need.

- **Deployment Tasks**

Deployment tasks include the details of what is to be deployed, what order they are to be deployed in, and what needs to be done before, during, and after they are deployed.

- **Task Templates**

  Task templates store commonly-used sets of deployment tasks. These can be applied to release packages to ease configuration and keep consistency from deployment to deployment.

- **Environments**

  Environments are logical representations of the physical environments used by your organization. Typical environments include testing, staging, and production.

- **Deployment Paths**

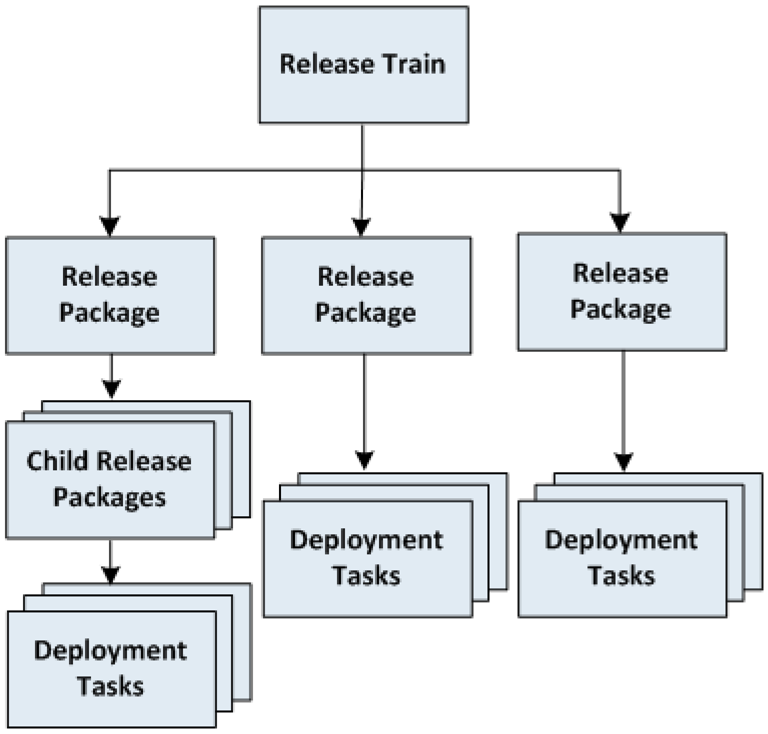
  Deployment paths are sequences of environments through which release packages progress during the release cycle.

- **Plugins**

  Plugins enable you to access objects in integrating products directly from Release Control. Several plugins are provided as part of the default installation.

Release Control Objects

Release Control includes the following objects, which together manage the overall release process. The object relationships of Release Control are shown in the following figure.



- **Release Trains**

  Release trains manage sets of release packages.

- **Release Packages**

  Release packages manage sets of change that are to be released together. You can define parent release packages that have child release packages. Only individual or parent release packages can be associated directly to release trains; child release packages cannot be directly associated.

- **Deployment Tasks**

  Deployment tasks manage the release activities.

Release Control Participants

Release Control provides roles for users in your organization who may have the following titles and responsibilities.

- **Release Control Administrators**

  Administer Release Control, performing duties such as setting up and maintaining groups and users and organization-specific release entities.

- **Release Managers**

  Ensure that a release process is followed over the entire life cycle of each release train.

- **Release Package Creators**

  Create release packages.

- **Release Engineers**

  Manage release packages, including creating deployment tasks, designing deployment paths, and accepting and deploying release packages.

- **Release Control Users**

  Any users who will access Release Control, such as developers, but who don't have one of the other distinct roles.

- **Approvers**

  Approve release train stages, review, and accept exit criteria for stage gates, and ensure milestones are met as planned and dates are adjusted if necessary.

A Typical Release Control Flow Using Release Trains

This section describes a typical successful flow using release trains. The flow you use in your implementation may be different.

1. Create the release train.
2. Create standalone release packages.
3. Complete construction on the release packages.
4. Deploy and validate release packages through the pre-Production environments in the deployment path.

5. Deliver production-ready release packages to the release engineering team for acceptance.

6. Optionally organize the release packages in a logical parent/child hierarchy for attachment to a release train.

   Tip: This is typically done in preparation for deploying the release train, and is therefore done after candidate release packages have been tested through test environments in a pre-production deployment path. Once it has been determined which release packages are ready to be released as part of a particular release train, the release package hierarchy can be added to the release train.

7. Add all standalone and parent release packages targeted for this release to the release train.

   Tip: Ensure the top-level release packages are associated with the production deployment path.

8. Verify and send the release train for release approval.

9. Initiate deployment so the release packages are deployed to production on their scheduled date.

After release package deployment to production is verified, the release train is closed by the release manager or a release engineer.

A Typical Flow Using Release Packages

This section describes a typical successful flow using release packages without release trains. The flow you use in your implementation may be different.

1. Create parent release packages.

2. Optionally create child release packages.

3. Complete construction on the release packages.

4. Deploy and validate release packages through the pre-Production environments in the deployment path.

5. Deliver production-ready release packages to the release engineering team for acceptance.

6. Initiate deployment so the release packages are deployed to production on their scheduled date.

After release package deployment to production is verified, the release packages are closed by the release manager or a release engineer.

Key Metrics

Release Control provides several views into your release activities.

- Release Dashboard [page 11]

- Timelines [page 11]
- Calendar Views [page 12]
- Activity Views [page 13]
- Custom Reports [page 13]

These views are available when the **Release Ctrl** application group is pinned and selected. Views are available to all Release Control users, but privileges determine which data is visible to users in each view.

All views except timelines are implemented using standard Serena Work Center functionality. For information on modifying these, refer to the Serena Work Center help.

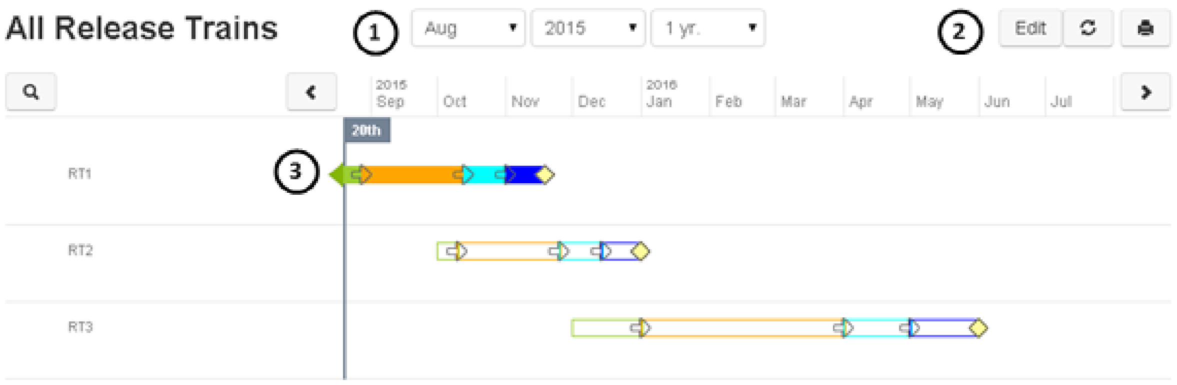
Release Dashboard

The Release Dashboard includes widgets with graphical reports that show the overall health of your release activities. By default, widgets with the following reports are provided:

- Current Overdue Milestones
- Open Exit Criteria
- Release Train Throughput
- Deployment Failure by Release Package
- Release Packages on Environments by State

Timelines

A customizable Timeline presents the overall status for release trains in a Gantt chart format. The Timeline provided in the default implementation is shown in the following figure.



1. Select a time range for the data to display.
2. Click the **Edit** button to change the settings for the Timeline. Note the following:
   - To edit the Timeline, you must have the Remote Administration privilege.
   - When you change the Timeline, it is changed for all users.

- If you change the report to something other than All Release Trains, you must ensure that the selected report has all the fields required for the Timeline. You can use the All Release Trains report as a model for a new report.

3. Move your cursor over icons or colors on the Timeline to see specific details. The Timeline in the default implementation consists of the following parts:

   - **Primary items:** Release trains that are plotted on the Timeline and listed on the left.
   - **Ranges:** The colored bars that are displayed horizontally across the Timeline. Ranges must have a start date and an end date and can optionally have actual start and end dates.
   - **Events:** Single points in time represented on the Timeline with icons. Like ranges, events can optionally have actual dates and icons configured for comparison.
   - **Overlays:** Milestones that come from a different report than the primary items and are linked with a relational field. This powerful feature enables you to plot both ranges and events from different items directly on the Timeline of the primary item.
   - **Child items:** Release packages plotted on the Timeline beneath their parent release trains. Child items can have their own ranges, events, and even overlays.

You can also create a new Timeline and add it to Work Center. For details, view the "Timeline" posts on Serena Central.

Calendar Views

Release Control provides several system Calendar views that show various release metrics in a calendar format. Each of these system views has feeds that show schedules for each application; users can choose which of the feeds to view for a browser session.

The following default feeds are included with the system Calendar views:

- **Release Calendar**
  - Release Trains - Shows end dates for all trains, along with train status.
  - Milestones - Shows due dates for all milestones, along with the approval state.
  - Release Packages - Shows last deployment date for all release packages, along with deployment result.
- **Release Trains**
  - Due Dates of Milestones
  - Scoping Complete Dates
  - Construction Complete Dates
  - QA Complete Dates
  - End Dates

- **Release Packages**
  - Submit Dates
  - Deployment Dates
- **Environments**
  - Scheduled events waiting for approval
  - Rejected scheduled events
  - Approved scheduled events
- **Approvals**
  - Due Dates

Activity Views

Release Control provides system Activity views that give users easy access to release items based on feeds.

Three types of Activity views are provided:

- **Release Activity**

  Shows all release trains, approvals, environments, task templates, and release packages.

- **My [Activity**]

  Depending on the selection in the navigation pane, shows all items owned by the current user. For example, "My Approvals" shows all approvals owned by the current user.

- **All [Activity**]

  Depending on the selection in the navigation pane, shows all items. For example, "All Trains" shows all release trains in the system.

Administrators can edit and remove the default feeds from each system Activity view. They can also add custom feeds. These changes are visible to all Release Control users.

Users can add Activity feeds to the system views, but they can remove only the feeds they have added.

Custom Reports

Release Control includes the power of SBM reports, enabling you to define search, display, and sort information for release activities. Reports offer real-time data based on the report criteria, giving users current information when they need it. Several types of report formats are available, including lists, tables, and charts.

Reports can be added to Dashboard views. For details, refer to the Serena Work Center help.

Chapter 3: About the Solution Process Apps

The Release Control process apps each have their own process, while at the same time interacting as part of the larger release control process. The following topics give an overview of each process app and explain how each fits into the overall release control processes.

- About the Release Train Process App [page 15]
- About the Release Package Process App [page 17]
- About the Deployment Path Application [page 18]
- About the Approvals Process App [page 20]
- About the Release Data Process App [page 22]
- About the Environment Process App [page 22]
- About the Task Templates Process App [page 24]
- About the Manual Deployment Task Process App [page 25]
- About the Sample Request Process App [page 26]

About the Release Train Process App

Release trains manage one or more release packages that have a common goal. That goal could be a date, such as a quarterly train; a set of features, such as performance or security improvements; or some other business requirement.

Most release trains are planned around a specific time frame, such as a month or fiscal quarter, but some solve a specific problem, such as improved security or database performance. Each release train item provides easy access to critical data points, such as schedules and milestones, release train status, gate exit criteria, approvals, and release packages.

Participants

The primary participants in Release Trains are as follows:

- Release Managers
- Release Engineers
- Approvers

Stages

Release trains follow a process organized into these stages, based on the swimlanes in the Release Train workflow:

- Plan [page 16]
- Do [page 16]
- Check [page 16]
- Deploy [page 16]

Note:

The exit criteria points, called gates, do not match the swimlanes, which are a variation on the PDCA (plan-do-check-act). PDCA is an iterative four-step management method used in business for the control and continuous improvement of processes and products. The swimlanes in the Release Train are PDCD (plan-do-check-deploy).

Following are the stages of the release and the specific actions that can be done in those stages.

Plan

The Plan stage is the primary planning stage for a release train.

- Once planning activities are completed, the release manager sends the train to the Do stage.
- Before the release train can proceed, approvers must approve to validate that any planning exit criteria is met.

Do

Release trains are sent to development during the Do stage.

- In this stage, all release packages are typically under development. Most of the work during this stage is managed through the Release Packages process app.
- Before the release train can proceed, approvers must approve to validate that any development exit criteria is met.
- After approval, the train progresses to the verification stage.

Check

In the Check stage, the functionality delivered by the release packages associated with the release train is verified.

- During verification, the release manager verifies that the QA exit criteria has been met, and then sends the release train for release approval.
- After all levels of release approval have been received, the train is deployed to production.

Deploy

In the Deploy stage, the release packages associated with the release train are deployed to final or production environments.

- During this final stage, the release manager or a release engineer determines that the train is ready for deployment and deploys the train.
- Once deployment successfully completes, the train is closed by the release manager or release engineer.

About the Release Package Process App

The Release Package process app includes two applications: Release Package and Deployment Path. Each has a single workflow.

- The Release Package application is described in this topic.
- The Deployment Path application is described in About the Deployment Path Application [page 18].

Participants:

- Release Managers
- Package Creators
- Release Engineers

Release Packages follow a process organized into these stages:

- Define [page 17]
- Deliver [page 17]
- Verification [page 18]

Define

The first stage of the Release Package is the definition stage.

While the release package is in the Define stage, release engineers can:

- Select release type, which limits the deployment paths available for the release package.
- Add deployment tasks. These can be newly-created tasks specific to the release package or tasks copied from task templates and other release packages. Deployment tasks can be added, reordered, or deleted as needed as the release package is deployed through the deployment path.
- View and associate requests and deployment units.

Once accepted, the release package moves to the *Deliver* stage.

Deliver

In the Deliver stage, the release packages are deployed.

- Accepted release packages move to the *Ready to Deploy* state.
- Once the deployment is started, deployment tasks initiate the related deployment processes and deploy any deployment units.

- Once all deployment tasks are complete, the release package moves to the *Deployed* state. If any deployment tasks fail, the release package is sent to the *Deployment Error* state. Failed release packages can be retried or returned to development for more work.
- Successful deployments move to the *Verification* stage.

Verification

In the Verification stage, the functionality implemented by the release packages is verified.

- Deployed release packages move to the *Environment QA* state to be verified.
- If issues are found during the verification, the release package can be transitioned as follows:
  - If functional changes are needed:

    Sent back to the *Define* stage for fixes to be made to the deployment units
  - If changes are needed to deployment tasks or the environment configuration:

    Redeployed to the same environment for further verification after those changes are made
- If redeploy is allowed for the release package, and the environment is not locked, you can fix any problems and then redeploy, using different deployment units as needed. For example, this would typically be when QA is testing software and opening issues for Development to fix in the pre-release testing cycle. This would continue until you want the release package to proceed to the final environments, such as pre-production staging and production.

  Note: Redeploy is different from retry in that redeploy is used after the release package has been successfully deployed, but the functionality of the release package has been tested and failed. Retry is used if the release package did not successfully deploy.
- If no issues are found during the verification, and if there are additional environments pending in the deployment path, the release package moves to the next environment. Otherwise, it moves to the *Completed* state. In *Completed* status, a release package can still be associated with a release train. Once the cycle of the release package is completely done, the release engineer or release manager can move it to *Closed*.

About the Deployment Path Application

The Deployment Path application is included in the Release Package process app.

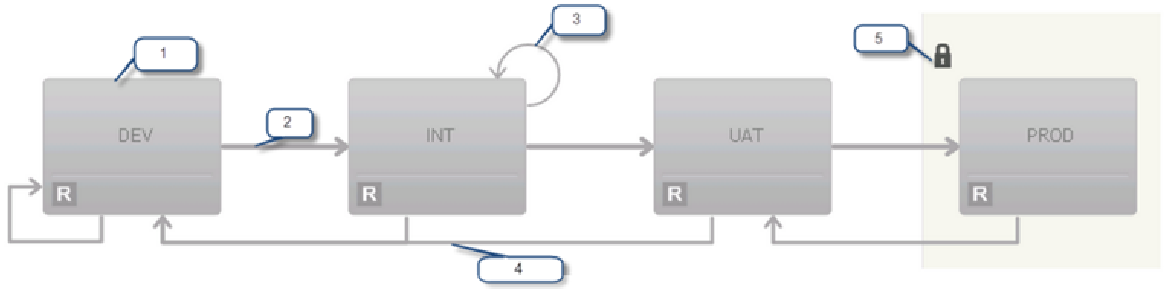
Deployment paths are a sequence of environments through which a release package must progress, such as development, integration testing, and production. This allows you to ensure that each release package has gone through the required testing stages before it moves to production. You can define whether each environment is required or optional on this path. You can also define when the release package is locked to prevent further changes, ensuring that what was previously tested is what is going to be deployed to your production environment.

Deployment paths can be associated with specific release types, which enable you to limit the selection of paths available for a release package. For example, you may have both "development" and "production" release types for your deployment paths. The "development' paths would be available to developers as they test their items on the development and QA servers. The "production" paths would be available to release engineers and would allow the release packages to be deployed to staging and production environments.

The deployment path is designed using the graphical interface to add, modify, and delete the environments in the paths. For example, a simplified environment sequence would be the following:

- DEV (Development)
- INT (Integration Testing)
- UAT (User Acceptance Testing)
- PROD (Production)

An example deployment path is shown in the following figure.



1. Each environment is represented by a box. An R in the bottom left of the box means the environment is required.
2. A successful deployment moves a release package to the next environment in the state.
3. If an environment allows redeploy, a returning line appears at the top of the diagram.
4. Failure selections are shown on the bottom of the diagram.
5. A lock icon indicates which environments are locked.

Deployment paths follow a process organized into these stages:

- Create [page 20]
- Active [page 20]
- Inactive

Participants:

- Release Engineers

Typically, deployment paths are created when you implement Release Control for your organization. You generally have a limited set of deployment paths that should be available to use for your release packages, and those should be defined and ready to be selected by release engineers who use Release Control to deploy release packages.

Create

While a deployment path is in the Create stage, release engineers can:

- set release type

  Note: Multiple deployment paths may have the same release type. For example, you may have a Major release type that includes a full set of environments for one team, and another that includes a full set of environments for another. Although they are both Major release types, they may follow slightly different deployment paths.

- select and add eligible environments to the path in a sequence
- set options for the environment on the path, such as:
  - **Required:** Indicates whether deployment to an environment is required or optional
  - **On Fail:** Specifies which environment to go to in case of a failure
  - **Set Lock:** Indicates whether an environment is locked, so that no changes are allowed to the release package when it is deployed to that environment unless it is reverted to a different environment; this can be used to implement a code freeze.
  - **Redeploy:** Indicates that the release package may be deployed again into the same environment

Active

While a deployment path is in the Active stage, package creators, administrators, or release engineers:

- can choose the deployment path when creating a release package
- cannot make changes to the deployment path; to modify a deployment path, it must first be returned to the Create stage

About the Approvals Process App

Approvals help enforce processes and bring accountability to your release management activities. Approvals may be required to allow a release train to proceed to final deployment stages; to validate that a release train may exit a stage based on gate exit criteria; and to confirm that pre-defined milestones for a release have been reached by a certain date.

Release managers determine approval needs during the Plan stage of a release train. Release approvals are added and managed from release trains; the Approvals process app stores approvals and determines the approval process.

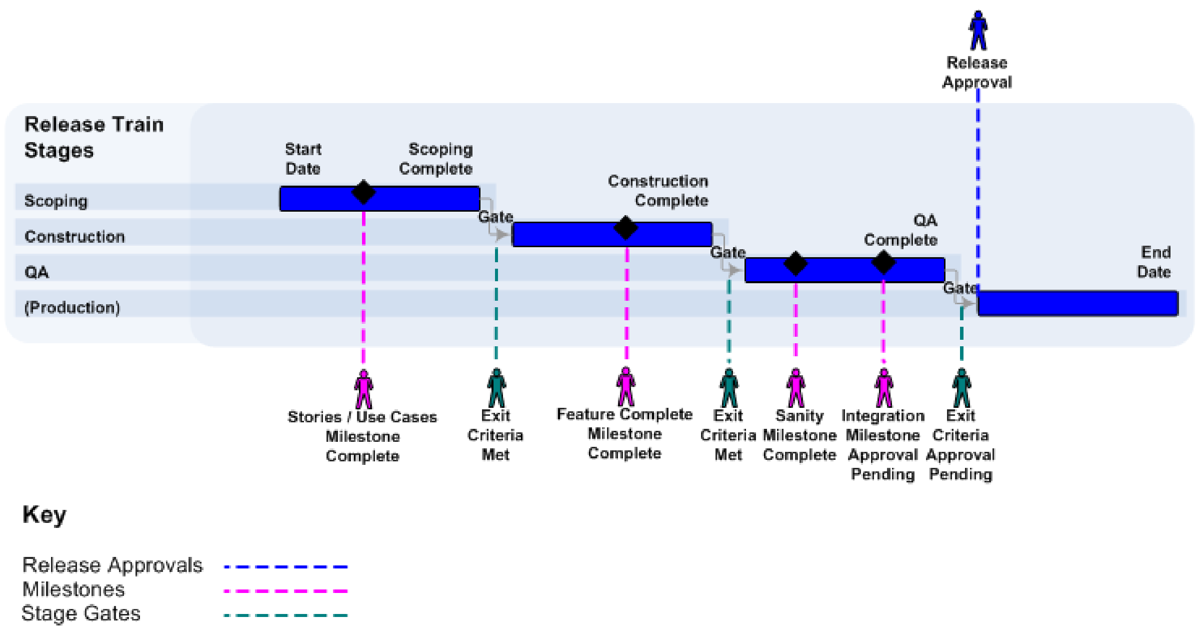
The following approval types are provided:

- Release Approval [page 21]
- Gate Approvals [page 21]
- Milestone Approvals [page 22]

E-mail notifications for each type are configured so that recipients can approve or reject directly from the e-mail they receive.

The approval types, including example milestones, are shown in the following figure.

Release, Milestone, and Gate Approvals



Release Approval

Release Approval is intended for approval boards that need to approve a release train to be deployed to the final environment, often referred to as Production.

Note: Release Approval can be done by individual users, but not groups.

Gate Approvals

Release managers can use gate exit criteria to require individuals to approve a train to exit a certain stage. Exit criteria may be added for each gate, Scoping, Construction, and QA. For example, the QA gate may have exit criteria for successful test automation runs. If exit criteria is added for a gate, approval is required before the train can continue past that gate.

- The gate completion date is automatically used as the due date for associated exit criteria. If the schedule for a gate is changed, the due dates for any exit criteria for that gate can be automatically updated as well.

- The release manager is responsible for creating exit criteria for each gate. Exit criteria owners are responsible for ensuring exit criteria are completed.
- Use the **Timeline** view for release trains to monitor gates and exit criteria.

Note: Gate exit criteria are added and managed from release trains, while the Approvals process app stores these items and determines their process.

Milestone Approvals

Milestones are date-based activities against which release train progress can be measured. For example, you may create a milestone for completing drafts of technical documentation.

- You can set completion dates and owners for each milestone.
- Milestone owners are responsible for completing or rejecting milestones.
- Use the **Timeline** view for release trains to monitor milestone activities.

About the Release Data Process App

The release data process app serves the following purposes:

- Stores data common to multiple process apps, such as applications, application versions, and release types
- Stores the custom column data
- Maintains an upgrade path for previous releases of Release Control that used the Application Release process app

The only participant in Release Data is the Release Manager, and the only stages are Active and Closed.

About the Environment Process App

Environments represent a set of resources to which release packages may be deployed. Environments manage and give visibility into the scheduling of maintenance and other events in environments, flagging any scheduling conflicts. Release packages are scheduled into environments in deployment paths.

The Environment process app includes two applications: Environment and Environment Schedule. The Environment Schedule application includes the Environment Schedules and Path Elements workflows. The applications and their workflows are described in the following sections:

Environment [page 22]

Environment Schedule [page 23]

Path Elements [page 24]

Environment

The primary application in the Environment process app is Environment.

Participants:

- Release Engineer
- Release Manager

Release engineers typically own environments.

Release engineers and release managers can do the following for environments:

- Create (commission) environments.
- Schedule maintenance or other events and decommission environments as needed. This blocks out periods of time when release packages cannot be scheduled for the environments.
- Decommission environments that are no longer needed.

Process Phases:

Environments follow a process organized into these stages:

- **Provision**

  Environment items are created. Each environment must be owned by a release engineer.

- **Active**

  Environments are placed online, signaling that scheduling can take place for release packages, maintenance windows, and other environment events, such as events for release trains.

- **Inactive**

  The environment owner can take environments offline for maintenance and return them back online when maintenance is complete.

Environment Schedule

The Environment Schedule application implements the environment scheduling process.

When an event is scheduled for an environment, the scheduled event can be required to go through an approval process. If you schedule the environment in a time period that overlaps, you'll get an error. The error is highlighted on the schedule and notifications go to those subscribed.

The Environment Schedule application manages the following:

- **Maintenance Events:** You can schedule a maintenance event during a period of time the environment is being initialized, upgraded, or otherwise maintained. This blocks the environment from being scheduled for a release package or other events.
- **Other Events:** You can schedule other types of events in addition to maintenance events; the other event has no limitations on it, in that it can be used for any type of event that you need to schedule, such as environment provisioning.
- **Conflicting Events**: You cannot schedule more than one maintenance event into overlapping timeframes; if you try, you'll get a message in the form with the schedule conflict highlighted.

- **Schedule Approvals**: You can have approvals for the environment schedule either through the Pending Requests tab or emails. You can approve, reject, reschedule, update, or delete.
- **Rescheduling**: When you reschedule, you can either set the schedule then, or mark it to set the schedule later. You have to start and finish maintenance using the corresponding options.
- **Event Status**: The status of the scheduled events are shown on the schedule calendar.
- **Scheduled Release Packages**: When you have scheduled a release package into an environment, those will show on the schedule calendar as well.

Process Phases:

Environment schedules follow a process organized into these stages:

- **Schedule**

  Environment schedule items are created. The environment schedule is owned by the submitter or a release engineer.

  If approvals are required, the environment schedule item must be approved by a release engineer or release manager.

- **Active**

  The environment schedule is started.

  When the environment item leaves the deployment state in the workflow, the environment schedule is completed.

- **Inactive**

  The environment schedule is complete.

Path Elements

The Path Elements workflow is implemented in the Environment Schedule application.

The individual environment items in a deployment path are stored in the Path Elements project. The path element items act as templates and are not associated with a specific release package. When a deployment path is used in a release package, duplicates are made of these path element items and associated with the specific release package. The new items are created in the Environment Schedules project.

About the Task Templates Process App

Task templates are templates for a collection of ordered deployment tasks. Task templates can be created from release packages and deployment tasks. Deployment tasks from approved task templates can be copied to release packages.

Note: Deployment units cannot be associated with task templates. These are associated only with release packages, which are for specific deployments.

Participants:

- Release engineer

- Release manager

Task templates follow a process organized into these stages:

- **Create**

  Release engineers typically create a task template and add deployment tasks. Tasks can be also copied from other task templates or release packages.

  Once tasks are added, the task template is sent to a release manager for review.

- **Active**

  Once a task template is accepted, tasks can no longer be added or modified.

  The task template's tasks can be copied to release packages, however, and reordered and modified as needed. New deployment tasks can also be added to release packages.

  Note: To modify an accepted task template, return it to the construction state.

- **Inactive**

  Inactive task templates are no longer available to release packages.

About the Manual Deployment Task Process App

The Manual Deployment Task Process App is a sample process app. You can use this as a model for a process app into which you want to submit or transition manual deployment task items or as part of a proof of concept. If you want to use this process app, you can use it as-is or customize it as needed to fit your needs. It is not necessary to use this process app at all if you have your own SBM process apps that you prefer to use for manual deployment task items.

Note: The example SBM plugin execution provider type is configured to interact with this process app in the default implementation of Release Control.

Participants:

- Submitter
- Task Owner

The sample manual deployment tasks follow a process organized into these stages:

- **Pending**
- **In Progress**

About the Sample Request Process App

The Sample Request Process App is a sample process app. You can use this as a model for a process app into which you want to submit request items and subsequently associate them with a release package. If you want to use this process app, you can use it as-is or customize it as needed to fit your needs. It is not necessary to use this process app at all if you have your own SBM process app that you prefer to use for this purpose.

Note: The example SBM plugin request provider type is configured to interact with this process app in the default implementation of Release Control.

Participants:

- Request Owner

The sample requests follow a process organized into these stages:

- **New**
- **Active**
- **Inactive**

Chapter 4: Configuring Release Trains

After you have created the basic release elements and configured users, users can begin using Release Control. However, administrators and release managers should set up an initial set of objects as they will be used for your organization's release management before making the solution live to all users. These should be used to verify the processes and train users on your implementation of Release Control. Thereafter, release participants will continue to create and configure release objects as needed to support your ongoing releases.

For details on configuring release trains, see the following topics.

- Creating Release Trains [page 27]
- Creating Approval Items [page 29]
- Approving Release Items [page 31]
- Releasing Release Trains [page 33]

Creating Release Trains

You should create a release train when you are ready to request approval for or begin scoping and scheduling the deployment of one of more release packages. You can create a release train before or after you create the release packages that will be associated with it.

Prerequisites: Ensure you have set up the minimum objects required before you try to create a release train.

To create a release train:

1. In Work Center, click **+New**.
2. Search for the project that will store your release train, and then select the project to open a Submit form. (By default, the Release Trains project is used to store release trains.)
3. Fill out the required fields and any optional fields and then click **Submit**.
4. Configure and manage the release train using the tabs that appear in the form.

Required Fields

The required fields when you create a release train are as follows:

- **Title**
- **Purpose**
- **Release Manager**
- **Dates:** Specify the dates for the release train stages, including Start Date, Scoping Complete Date, Construction Complete Date, QA Complete Date, and End Date. You

can also select Quarter Release or Six Month Release to auto-populate the fields using today's date as the start date, or calculate them using a specified start or end date.

Optional Fields

Optional fields include:

- **Release Engineers**
- **Release Approval Required:** Specify whether approval is required for the release train to enter production after QA is complete. Release approval is required by default.
- **Message Log**

Tabs

The tabs available from a release train once it is created are as follows:

- **Overview:** View reports on release packages associated with this release train. Reports display active release packages by environment and by state.

  From this tab, you can do the following:

  - Create and link release packages. You can see the list of release packages, including any parent and child release packages.
  - In the list of release packages, click the eye icon beside a release package to see a quick view of the details without opening the item.
  - In the list of release packages, click the release package link to open the item.
- **Schedule/Gates:** View or update the release train stage schedule. Specify exit criteria for each of the stage gates.
- **Details/Approvals:** Specify details of the release approval item, including the approval level and approvers. Specify the milestone dates that major aspects of the release should be complete.
- **Notes/Attachments:** View notes and attachments that have been added to the item through the Serena Work Center **More** option and related email correspondence.
- **History**: View the history of the release train item.

Copying Release Trains

To ease the release train setup process, you can create a release train and copy it to create new release trains, modifying the information as needed for the new trains. If you plan to use more than one type of release train, consider setting up a standard release train of each type that you plan to use so that release managers can then copy them and complete their configuration more quickly.

Information that you may want to set up ahead of time so that it can be copied to various release trains includes:

- Release manager and release engineers

- Gate exit criteria
- Release approval levels and approvers
- Milestones

During a release train copy, release managers select new start and end dates. Dates for stage gates and milestones automatically adjust relative to the new train stage dates. Dates can be adjusted manually as needed.

To copy a release train, select the **Make a copy** link located on the **More** drop-down list on an existing release train.

Note: The release train copy does not copy the associated objects, such as release packages. These must be added to the newly copied trains.

Creating Approval Items

You can create Approval items that will subsequently be approved as part of the release life cycle. These help to enforce the release processes you have defined.

For details on each type of approval, see the following topics:

Creating Milestone Items [page 29]

Creating Gate Items [page 30]

Creating Release Approval Items [page 31]

Creating Milestone Items

From the **Details/Approvals** tab of a release train, you can define milestone items based on activities that you want to approve at various points throughout the release. For example, when Agile stories or use cases are complete, when all features of the planned development are complete, and when certain test phases are complete, such as sanity and integration testing. Milestone dates are date based.

From the **Details/Approvals** tab of a release train or application version, you can:

- Define milestone items based on activities that you want to approve at various points throughout the release
- View milestone items' approvers, due dates, and states
- Complete, reject, update, or delete milestone items

To create milestone items:

1. On the **Details/Approvals** tab, click **Add Milestone**.
2. Enter a descriptive title or accept the automatically-generated name.
3. Enter a description.
4. Specify a due date for the milestone.
5. Select an approver.
6. Click **Submit**.

Release train and application version milestone items are submitted into and enforced through the Approvals process app workflow.

Note: You can create multiple milestone items for a release train.

Tabs

The tabs available from a milestone once it is created are as follows:

- **Train Details:** View information about the release train.
- **Notes/Attachments:** View and add notes and attachments for the item.
- **History**: View the history of the milestone item.

Creating Gate Items

From the **Schedule/Gates** tab of a release train, you can define exit criteria items that must be verified and approved before the release train can leave each stage, Scoping, Construction, and QA.

From the **Schedule/Gates** tab of a release train, you can:

- Add Scoping, Construction, and QA Exit Criteria items
- View Exit Criteria items' approvers, due dates, and states
- Complete, reject, update, or delete Gate Exit Criteria items

To create Gate Exit Criteria items:

1. On the **Schedule/Gates** tab, click **Add <stage> Exit Criteria**, where <stage> is Scoping, Construction, or QA.
2. Fill out the form as follows:
    - Enter a descriptive title or accept the automatically-generated name.
    - Enter a description.
    - Select the environment to which the exit criteria item applies. For example, DEV, INT, UAT, or PROD.
    - Select an approver.
    - Click **Submit**.

      Release train gate exit criteria items are submitted into and enforced through the Approvals process app workflow.

Note: You can create multiple exit criteria items for each of the gates.

Tabs

The tabs available from an exit criteria item once it is created are as follows:

- **Train Details:** View information about the release train.

- **Notes/Attachments:** View and add notes and attachments for the item.
- **History**: View the history of the exit criteria item.

Creating Release Approval Items

Release approvals may be required before the train is deployed into production.

From the **Details/Approvals** tab of a release train, you can:

- Add Release Approval items
- View Release Approval items' states and approval types
- Mark a Release Approval items as ready for approval, or update or delete Release Approval items

To create Release Approval items:

1. On the **Details/Approvals** tab, click **Create Release Approval**.
2. In **Approval Type**, select the level of approval, Level 1 up to Level 4.
3. Depending on the level you chose in **Approval Type**, select the approvers for each required level of approval.
4. If you want to allow individual users with the same role as the selected user to approve if needed, select *Yes* for **Allow Override Approvals**.
5. Click **Submit**.

Note: You can create only one Release Approval item for each release train.

Tabs

The tabs available from an approval item once it is created are as follows:

- **Approvers:** View the list of approvers for this Release Approval.
- **Train Details:** View information about the release train.
- **Notes/Attachments:** View and add notes and attachments for the item.
- **History**: View the history of the Release Approval item.

Release train approvals items are submitted into and enforced through the Approvals process app workflow.

Approving Release Items

Typically, you should approve items from the release train. Some users may only be approvers, and may only have the Approvals application in their Release Control **Release Ctrl** application group. Those users must approve using the Approvals application.

Information on how to approve is given in the following sections:

Approving Release Approval Items [page 32]

Approving Milestone and Stage Gate Items [page 33]

Approving Release Approval Items

For Release Approvals, the action **Ready for Approval** must be done before you see the options to complete or reject the item. This is typically done by the Release Manager. After the item is ready for approval, an approver for each level of approval must approve or otherwise action the item.

Ready for Approval

To make the Release Approval item ready for approval:

- From a release train, from the **Details/Approvals** tab, click **Ready for Approval** in the **Actions** list.
- From a release train, from the **Details/Approvals** tab, open the item you want to action. Click **More** to select from other options, and then click **Ready for Approval**.
- From the Approvals application, from the list of items, open the item you want to action. Click **More** to select from other options, and then click **Ready for Approval**.

Approval

To approve Release Approval items, or otherwise take action on them, you can do one of the following:

- From a release train, from the **Details/Approvals** tab, click **Approve**, **Override**, **Reject**, **Update**, or **Delete** in the **Actions** list.
- From a release train, from the **Details/Approvals** tab, open the item you want to action and click **Approve**, **Reject**, or **Update**. Click **More** to select from other options.
- From the Approvals application, from the list of items, open the item you want to action and click **Approve**, **Reject**, or **Update**. Click **More** to select from other options.

Overrides

When you choose to override an approval, the following options are available:

- **Types of override**:
  - Approve for another user
  - Add/Remove an approver
  - Move to next level of approval
  - Reject approval
- **Select users to approve for**
- **Override Reasons**

Approving Milestone and Stage Gate Items

To confirm milestones or stage gate exit criteria are complete, or otherwise take action on them, you can do one of the following:

- From a release train, from the appropriate tab, **Details/Approvals** or **Scehdule/ Gates**, click **Complete**, **Reject**, **Update**, or **Delete** in the **Actions** list.
- From a release train, from the appropriate tab, **Details/Approvals** or **Schedule/ Gates**, open the item you want to action and click **Complete** or **Reject**. Click **More** to select from other options.
- From an application version, from the **Details/Approvals** tab, open the item you want to action and click **Complete** or **Reject**. Click **More** to select from other options.
- From the Approvals application, from the list of items, open the item you want to action and click **Complete** or **Reject**. Click **More** to select from other options.

Releasing Release Trains

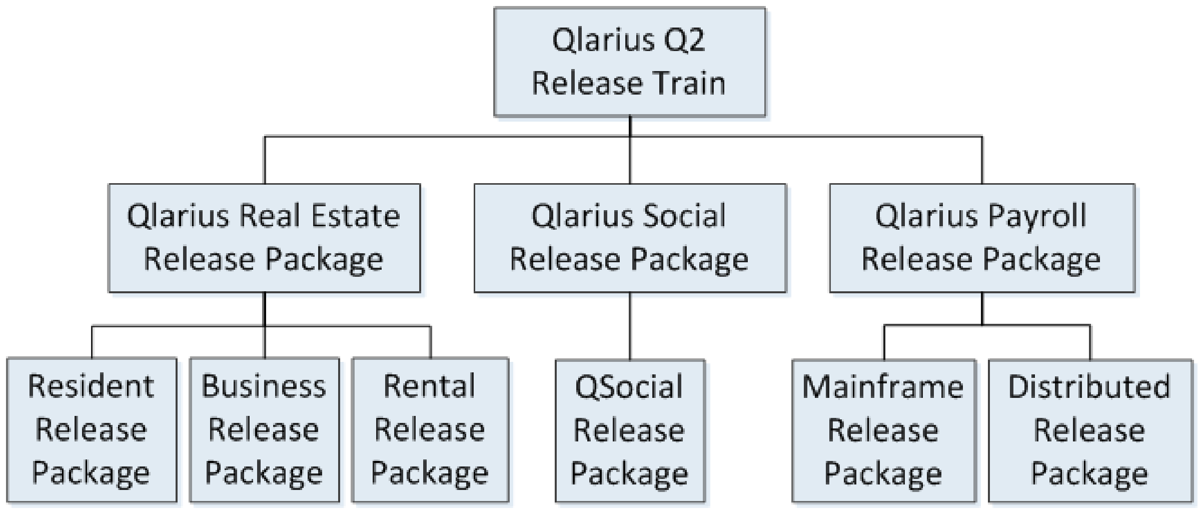
When you are ready to deploy all release packages that are part of a release train, you send the release train for release approval. Once approval is complete, you deploy each of its associated release packages on the scheduled deployment date.

See A Typical Release Control Flow Using Release Trains [page 9] and Deploying Release Packages [page 45].

Chapter 5: Configuring and Deploying Release Packages

Release packages contain all the information needed to deploy deployment units to and execute processes in environments. Release packages are typically created from and associated with release trains.

An example of release packages for a whole release train are shown in the following figure.

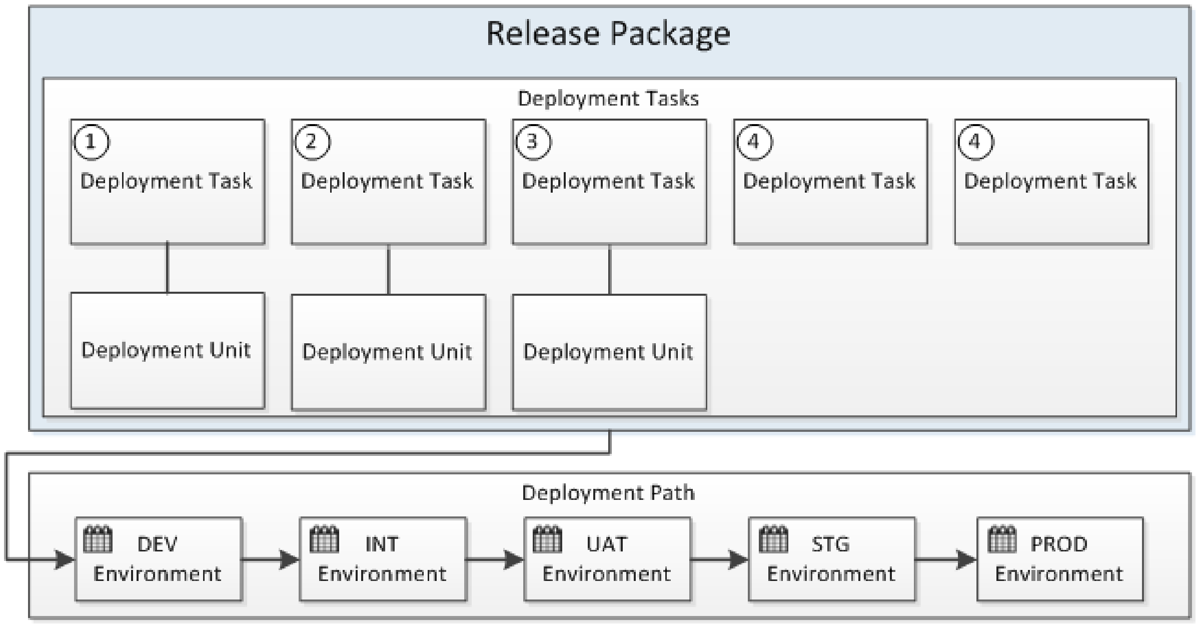


Qlarius Q2 Release Train
Qlarius Real Estate Release Package
Qlarius Social Release Package
Qlarius Payroll Release Package
Resident Release Package
Business Release Package
Rental Release Package
QSocial Release Package
Mainframe Release Package
Distributed Release Package

In this example, the release packages are grouped according to the makeup of various development teams for an application release. Release packages may represent a functional part of a release or components targeted to different platforms. A single release package can be used for the entire release. The categorization and makeup of release packages will depend on your organization's needs.

The details of what to deploy are defined in deployment tasks and the information on where and when to deploy it are defined in the deployment paths, which are a sequence of environments to which to deploy.

The following example shows the composition of a release package.



When a release package is deployed, it is deployed to each environment in the deployment path. The numbers on the deployment tasks are the order in which they are executed. More than one deployment task can have the same order number, such as order number 4 in this example, so that they are executed at the same time.

For information on creating and configuring release packages and scheduling and initiating release package deployment, see the following topics.

- Creating Release Packages [page 36]
- Adding Requests [page 40]
- Adding Deployment Units [page 41]
- Adding Deployment Tasks [page 42]
- Creating Task Templates [page 43]
- Validating Deployment Tasks [page 44]
- Scheduling Deployment [page 45]
- Deploying Release Packages [page 45]
- Adding Release Packages to Release Trains [page 47]

Creating Release Packages

You should create a release package when you are ready to begin configuring the deployment details for a release. You can create release packages by submitting to the release packages project. If using release trains, you can create a release package directly from a release train.

By default release packages are standalone, but they can be designated as parent or child packages. See Designating Release Packages as Parents [page 38] and Adding Child Release Packages [page 39].

Prerequisites:

- You must have at least one active deployment path.

To create a release package:

1. Depending on whether you are using release trains, open the release packages project submit form in one of the following ways:
   - In Work Center, select a release train, and in its **Overview** tab, click **Create Release Package**.
   - In Work Center, click **+New** and select your release packages project.

     (By default, the Release Packages project is used to store release packages.)
2. Fill out the required fields and any desired optional fields.

   Tip:

   - Enter the version of the release package in the **Title** or **Description** so that you can later report on and compare release package versions through standard SBM reports.
   - In the **Overview** section, in the **Use Task Template** field, select a task template from which to copy deployment tasks.
   - In the **Options** section, you can select custom column items for add and display of deployment units and requests.
   - In the **Options** section, you can select whether or not to bypass the *Deployed* and *Deployment Complete* states. By default, the release package process flow bypasses the *Deployed* state. If you want to have users manually transition to either of these states, you can set that option to *No*.
3. Click **Submit**.

   This places the release package in the *Acceptance* state.

   Note: You can instead click **Save as Draft**, which places the release package in the *Construction* state, where it must be configured before delivering it to the release engineering team for acceptance.
4. Configure and manage the release package using the tabs that appear in the form.

Required Fields

The required fields when you create a release package are as follows:

- **Title**
- **Description**
- **Release Type:** Select the release type, such as Emergency, Major, or Minor. The release type filters the deployment paths available to this release package.

- **Deployment Path:** Specify the deployment path for this release package. You must select the release type first, because deployment paths are filtered by release type. Only active deployment paths appear for selection.
- **Release Manager**

Tabs

The tabs available from a release package once it is created are as follows:

- **Deployment Path:** View the deployment path for this release package, including detailed deployment history for each environment in the path.

  Deployment paths are typically created and maintained by Administrators. For information on creating and activating deployment paths, refer to the *Serena Release Control Installation and Setup Guide*.
- **Deployment Units:** View deployment units for this release package. Refer to Adding Deployment Units [page 41].
- **Deployment Tasks:** View and add deployment tasks for this release package. Refer to Adding Deployment Tasks [page 42].
- **Requests:** Select requests to associate with this release package.
- **Notes/Attachments:** View notes and attachments that have been added to the item through the Serena Work Center **More** option and related email correspondence.
- **History**: View the history of the release package item.

Designating Release Packages as Parents

You can configure release packages to be parent release packages.

Parent release packages:

- Organize a set of related release packages that must move through a deployment path together
- Have one or more child release packages associated with them
- Can be associated directly with release trains

Prerequisites:

- At least one release package to be designated as a child and one release package to be designated as a parent must exist

To designate a release package as standalone:

1. Open the parent release package that you want to designate as a parent.
2. Under **More**, select **Link Package**.
3. In **Link Details**, in **Make the current release package**, select **Parent**.
4. Find and select the release package for which you want to make this release package a parent.

5. Click **OK**.

Tip: After linking or unlinking a child release package, click the **Reload Item** button to refresh the form.

Note: If you unlink all child release packages from a parent release package, it automatically becomes a standalone release package.

Related Topics

Adding Child Release Packages [page 39]

Working with Child Release Packages [page 39]

Designating Release Packages as Parents [page 38]

Creating Release Packages [page 36]

Adding Child Release Packages

You can configure a release packages to be a child of one other release package.

Prerequisites:

- At least one release package to be designated as a child and one release package to be designated as a parent must exist

To add a child release package:

1. Open the release package that you want to designate as a child.
2. Under **More**, select **Link Package**.
3. In **Link Details**, in **Make the current release package**, select **Child**.
4. Find and select the release package for which you want to make this release package a child.
5. Click **OK**.

Tip: After linking or unlinking a child release package, click the **Reload Item** button to refresh the form.

Related Topics

Working with Child Release Packages [page 39]

Designating Release Packages as Parents [page 38]

Creating Release Packages [page 36]

Working with Child Release Packages

After you have linked child release packages to their parents, you can work with the child packages.

A child release package:

- Cannot have any child release packages

- Has its own deployment task collection
- Follows the parent's workflow; deployment must be done at the parent level
- Follows the parent's deployment path
- When added to the parent, causes the parent to be reset to the beginning of the deployment path
- Retains a history of previous deployment paths it had before being added to a parent
- Retains a history of new deployment paths started after having been added to a parent

After a child is added, the child release packages have a **Parent Package Details** tab instead of a **Deployment Path** tab, to indicate that the deployment path is now following the parent release package path.

The parent release package has an additional tab called **Child Packages** that lists the child packages. In this list, you can do the following:

- Click the child release package link to go to the item.
- Click the eye icon to see a quick view of the child release package information in a pop-up.
- Through the **Actions** menu beside each child release package, you can select the following options:
  - Edit Deployment Tasks
  - Edit Deployment Units
  - Unlink Child Package
  - Edit Requests
  - Reload Deployment Unit Data
  - Reload Request Data

Related Topics

Designating Release Packages as Parents [page 38]

Creating Release Packages [page 36]

Adding Requests

You can associate existing requests with release packages to give visibility to what business requests or requirements are addressed in a release package.

Prerequisites: Before you can add a request, you must have a connection to the integrating product.

You must also have a supporting plugin configured. Refer the plugin documentation on the Documentation Center.

To add a request:

1. In a release package, select the **Requests** tab, and then click **Edit Requests**.
2. Click **+**, and then select the plugin request provider.
3. Select from the available requests.
4. Click **Add**.

Reloading Request Data

Request data is stored in Release Control when you associate requests with a release package. To ensure you have the latest data, you can update the requests with the latest data from the request provider.

You should reload the data if any of the following have occurred since you added the requests:

- If the requests have been updated in the integrating product
- If custom column fields have been added or removed
- If a plugin has been upgraded

Prerequisites: Before you can get request data, requests must be added and you must have a connection to the integrating product.

To reload the request data:

- In a release package, select the **Requests** tab, and then click **Reload Request Data**.

Adding Deployment Units

You can associate existing deployment units with release packages to give visibility to what is deployed in a release package.

Prerequisites: Before you can add a deployment unit, you must have a connection to the integrating product.

You must also have a supporting plugin configured. Refer the plugin documentation on the Documentation Center.

To add a deployment unit:

1. In a release package, select the **Deployment Units** tab, and then click **Edit Deployment Units**.
2. Click **+**, and then select the plugin deployment unit provider.
3. Select from the available deployment units.
4. Click **Add**.

Reloading Deployment Unit Data

Deployment unit information is stored in Release Control when you associate deployment units with a release package. To ensure you have the latest information, you can update the deployment units with the latest information from the deployment unit provider.

You should reload the data if any of the following have occurred since you added the deployment units:

- The deployment units have been updated in the integrating product
- Custom column fields have been added or removed
- A plugin has been upgraded

Prerequisites: Before you can get deployment unit information, you must have a connection to the integrating product.

To reload the deployment unit data:

- In a release package, select the **Deployment Units** tab, and then click **Reload Deployment Unit Data**.

Adding Deployment Tasks

If you are integrating with a product that provides the ability to execute processes, such as SBM, Serena Deployment Automation, Serena Dimensions CM, or Serena ChangeMan ZMF, and have the corresponding plugin configured properly, you can add deployment tasks. Depending on your selections, the execution of the deployment task may deploy the associated deployment units to the target environment, it may execute an operation on the environment, or it may submit or transition a request in another system to signal that a given task must be completed.

Prerequisites: Before you can add a deployment task, you must have a connection to the integrating product.

You must also have a supporting plugin configured. Refer the plugin documentation on the Documentation Center.

Note: Deployment tasks are implemented using the plugin execution provider type.

To add a deployment task:

1. In a release package, select the **Deployment Tasks** tab, and then click **Edit Deployment Tasks**.

   Note: For the **Edit Deployment Tasks** button to appear, you must be logged in as the Release Manager.

2. On the form that opens, click **+**, and then select the plugin execution provider.
3. Give the task a title and description.
4. Fill out the required fields and any optional fields and then click **Save**.

Tip: You can also copy tasks from a task template or from another release package.

Updating Deployment Tasks

Once you have created deployment tasks, you can do the following in the Edit Deployment Tasks list:

- **Reorder tasks:** Drag and drop to reorder. Move your cursor over the task you want to move, and when you see the move icon, drag and drop it where you want it.
- **Add deployment units to tasks:** Edit, delete, or link to a deployment unit by clicking the corresponding icons beside the task title. Move your cursor over the title area to see the icons.
- **Skip tasks:** Click the active environment name beside the deployment task you want to skip. It will be grayed out to indicate that the deployment task will not be run in that environment. Using this technique, you can inactivate the tasks for some environments and not others.

Creating Task Templates

If you have a standard set of deployment tasks that are performed for similar release packages, environments, or applications, you can create templates for these tasks. As release engineers create release packages, they can copy tasks from task templates, and then reorder, remove, or add new tasks as needed.

To create a task template:

1. Open the task templates project submit form in one of the following ways:
   - In Work Center, select a release package, click the **More** button, and select **Create Task Template**.
   - In Work Center, click **+New** and select your task templates project.

     (By default, the Task templates project is used to store task templates.)
2. Provide a name and description for the task template, and then optionally select values for the following fields:
   - **Release Type**

     Select the release type to which the task template applies.
   - **Applications**

     Select specific applications to which the task template applies.

     Note: Users can copy deployment tasks from any approved task template, but when you specify a release type or application, they can filter tasks to those related to a specific release type or application.
3. Select the release engineer and other release team members who will be responsible for the task template, and then click **Submit**.
4. Click the **Edit Deployment Tasks** button.
5. Use the form to add new deployment tasks, copy tasks from another task template or release package, delete tasks, or change the sequence of tasks. Click **Close** when you are done.
6. Click **Complete** to send the task template to a release manager for review.
7. Once approved, the task template is available for release packages.

Note: Deployment tasks associated with task templates won't have deployment units associated, since these are specific to an environment. After the task template is copied to a release package, deployment units must be associated with the release package.

Tip: You can also create a task template from a release package using **More > Create Task Template**.

Validating Deployment Tasks

Task are automatically validated when you start deployment. You can also validate them before you deploy to ensure your configuration is correct before you begin the deployment process.

Validation checks for the following:

- All environments have at least one deployment task
- None of the locked environments have an invalid task

- All required fields are filled in for the tasks
- All fields with values have valid values, in case something was changed in an application that makes the values no longer valid

To validate a deployment task:

1. In a release package, select the **Deployment Tasks** tab, and then click **Validate Deployment Tasks**.

   Note: For the **Validate Deployment Tasks** button to appear, you must be logged in as the Release Manager and have existing deployment tasks.

   The validation checks are completed for all deployment tasks associated with the release package.

Scheduling Deployment

After you associate a deployment path with a release package, you can then schedule the release package into the instance of the path that is associated with that release package.

The instance of the deployment path is specific to the release package and cannot be copied to another release package. New release packages start with the original deployment path, and the specific schedule must be added for each release package.

To schedule a release package deployment:

1. In a release package, select the **Deployment Path** tab.
2. In the table below the deployment path diagram, for each environment, click the schedule icon and select the date range and schedule owners.

Note: The schedules are informational only, and are not enforced. Once someone clicks **Start Deploy**, the deployment starts, regardless of the scheduled date.

Deploying Release Packages

Deployment initiates the deployment tasks that you have configured and scheduled.

Prerequisites: You must have completed the release package configuration, including adding the deployment tasks.

It is recommended that you validate the deployment tasks before beginning the deployment. See Validating Deployment Tasks [page 44].

Starting the Deployment

To run a release package deployment:

1. Select the release package you want to deploy.

   Note: Standalone and parent release packages can be deployed directly. Child packages are deployed when their parent release package is deployed.

2. Move the release package through its lifecycle until the **Start Deploy** option is available.

3. Click **Start Deploy** and enter any comments needed in the **Message Log** field.

   The deployment is initiated and the deployment tasks associated with the release packages begin. The deployment executes each of the deployment tasks in the order specified.

   CAUTION:

   After you click **Start Deploy**, validation automatically begins. If the deployment tasks are valid, deployment starts immediately. However, if any task is invalid, deployment will not start The release package will remain in the Ready to Deploy state and the following message will appear:

   "The current or locked environment has invalid tasks. Refer to the System Log for more information."

4. Check the deployment status.

5. After the deployment has been verified, click **Deployment Verified** and enter any comments needed in the **Message Log** field.

6. After testing is complete, click **Complete Testing** or **Failed Testing** and fill out the resulting form as needed.

Note: Even if you have a schedule and the start date has not yet arrived, when you click **Start Deploy**, the deployment starts; the schedule is not enforced by the software, and is for informational purposes.

Important: You cannot add environments or change the options once the release package is started on the deployment path. To add environments and change the options, you must change the deployment path and start again.

Checking Deployment Status

To check the status of the deployment:

1. Select the release package's **Deployment Tasks** tab.

2. Filter the list of deployment tasks by **Task Executions**.

3. Refresh the status as needed using the refresh button.

4. The status of the deployment task may differ depending on the plugin implementation. Typically:

   - Deployment tasks requiring manual action stay in the Active status until the open item is completed.

   - Deployment tasks that initiate external processes show the status as the process for the task is executed and information is retrieved from the product to which the plugin integrates.

If the deployment fails, investigate and fix whatever caused the problem, such as the configuration in the integrating product, a selection or sequencing problem in the deployment tasks, or configuration in the target environment itself. If the environment allows redeployment, you can redeploy the release package to the same environment. If it doesn't allow redeployment, you must return the release package to construction and progress the release package back through the process from there.

Deploying to the Next Environment

Use the provided options to transition the release package through the deployment process.

- When the deployment is marked complete for an environment, a list appears with the available environments for the next deployment.
- You will see the upcoming optional and required environments listed, and you can select the one to which you want to deploy, skipping the optional ones if desired.
- Start the deployment in the next environment and continue until you have completed the deployment to the last environment in the deployment path.
- When the deployment is completed in the last environment of the deployment path, a message appears telling you that the release package has been promoted to all environments and the release package is now complete.

Viewing Deployment Executions for the Path

View information about deployment executions in the **Deployment Path** tab.

Select deployment executions to see information for each deployment run.

- The path shows only the latest deployment into an environment, so when there are retries and redeploys, the prior ones are not shown.
- Click the information icon in an environment box to see the list of deployment units deployed to that environment. This information is also only for the latest deployment into that environment.

If you do not see the complete report below the deployment path and the environments to which you have deployed are not shown in color, refer to the "Troubleshooting the Deployment Log" section in the *Serena Release Control Installation and Setup Guide*.

Adding Release Packages to Release Trains

You can add standalone and parent release packages to release trains.

A Typical Release Control Flow Using Release Trains [page 9] describes one approach for when to add standalone release packages and when to add parent release packages to a release train at different points the release lifecycle. Your use of release trains may be different.

Prerequisites:

- At least one standalone or parent release package and one release train must exist.

To add a release package to a release train:

1. Open the release train to which you want to add a release package.
2. In the **Overview** tab, select **Link Release Package**.
3. Find and select the release package that you want to add to this release train.

4. Click **OK**.

   The **Overview** tab now shows the release package in a list along with any child release packages for parents.

Note: You can link a release package only from the release train. However, you can unlink it from either the release package or the release train.

Glossary

Applications

Release Control applications are logical representations of real-world applications that are being released as part of release trains or release packages. Application examples include a software product, a set of related components, and shared services. You may assign an application version to a release package to track that information.

Approvals

Approvals may be required to allow a release train to proceed to the final deployment stage; to validate that a release train may exit a stage based on stage gate exit criteria; and to confirm that pre-defined milestones for release trains have been reached by a certain date.

Deployment Paths

Sequences of environments through which release packages progress during the release cycle.

Deployment Tasks

Include the details of what is to be deployed, what order they are to be deployed in, and what needs to be done before, during, and after they are deployed.

Deployment Units

A packaged set of deployment-ready files and metadata that is associated with a deployment task. Examples include Deployment Automation component versions, Dimensions CM baselines, and ChangeMan ZMF change packages.

Environments

Logical representations of the physical environments used by your organization. Typical environments include testing, staging, and production.

Gates

Points in the workflow where exit criteria may be validated and approved as met before the release train can move to the next stage. In the default implementation, gates are defined at the end of each of the first three release train stages, Plan, Do, and Check, and are monitored through Planning, Development, and QA Exit Criteria.

Milestones

Date-based activities against which progress can be measured. Milestones are typically interim deliverables within a life cycle stage.

Release Approvals

Built-in approval items that may be required for the release train to enter the Scoping, QA, and final stages. The default approval names are Train, Development, and Executive. Up to four levels of approval may be defined for each of these.

Release Data

A process app whose main purpose is to manage the release data for Release Control, such as release type and application version.

Release Elements

Entities or objects that support your release processes, such as applications, environments, release types, and deployment paths.

Release Packages

Contain all the information needed to deploy deployment units to or run processes in environments.

Release Trains

Calendar-based management of a set of release packages across various life cycle stages.

Release Type

A user-defined category for release trains and release packages. Release type values are stored in the Release Data process app's auxiliary data, and you can easily add values that are appropriate for your implementation. The default values are Major, Minor, and Emergency.

Requests

Represent work items in response to a business need.

Stages

Logical divisions of the process app workflows that represent a roll-up of the states within the swimlanes of the workflow.

Task Templates

Templates of ordered deployment tasks.

Timelines

Present the overall status for release trains in a Gantt chart format.