SERENA®

# Development Manager 4.0

## Getting Started Guide

Serena Proprietary and Confidential Information

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

## Trademarks

## 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.

Publication date: October 2012

# Table of Contents

# Welcome to Serena Development Manager

Thank you for choosing Development Manager to plan and control your releases.

Serena® Development Manager (Powered by Serena Business Manager) 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.

Audience and Scope

This document is intended for personnel who participate in the processes of managing releases using Serena Development Manager.

Before You Begin

See the Readme for the latest updates and corrections for this document.

## Contacting Technical Support

Serena provides technical support for all registered users of this product, including limited installation support for the first 30 days. If you need support after that time, contact Serena Support at the following URL and follow the instructions:

http://www.serena.com/support

Language-specific technical support is available during local business hours. For all other hours, technical support is provided in English.

## Platform Support

For details of supported server and client platforms, third party integrations, and Serena Integrations , see the Serena Release Plan for Serena Development Manager at:

http://support.serena.com/Roadmap/Product.aspx?sel=DEV_MANAGER

From the Products list, select Development Manager, then click on the 4.0 release. From here you can display supported platforms and integrations.

## Demonstrations

Demonstrations of Serena product features can be viewed at the following public Web site:

http://courseware.serena.com

# Documentation

The Development Manager documentation set includes the following manuals and tutorials.

| Manual or Tutorial | Description |
|---|---|
| *Serena Development Manager Installation and Configuration Guide* | Describes how to install and configure the Serena Dimensions CM suite of products. (PDF manual) |
| *Serena Development Manager Getting Started Guide* | Gives an overview of Serena Development manager (PDF manual). |
| *Serena Development Manager Connector for HP Quality Center / ALM* | Describes how to configure and use the Development Manager web services that interact with HP Quality Center. |
| *Serena Development Manager Web Services Reference* | Provides Information on the web services provided for Development Manager. |
| *Serena Development Manager Readme* | Provides information about platform requirements and known issues for Development Manager. |

Access the documentation and tutorials for the Serena Dimensions CM suite through the Serena Help server at http://help.serena.com/alm/info_center/doc_index/dvm/4.0/index.html.

# Chapter 1

# Introduction to Serena Development Manager

# Welcome to Serena ALM

Serena Development Manager is a key component of the Serena Application Lifecycle Management (ALM) product line, providing software version and issue traceability from requirements definition through to deployment. With Serena ALM, you can rest assured that all changes to your critical business and commercial systems are managed and visible.

Serena ALM orchestrates your change management process through the powerful lifecycle engine in Serena Business Manager (SBM). SBM centralizes all of your related software lifecycle workflows, orchestrating events across diverse systems using Web services. You can define and customize your requirements management, application development, and software deployment lifecycles. Customize the stages, approvals, and events based on your own organizational needs and best practices.

# Key Benefits of Development Manager

Serena Development Manager enables you to orchestrate and monitor your key software development efforts, tracking source code changes and approvals through a central workflow engine. Development Manager uses Serena Business Manager (SBM) to coordinate events across your systems using Web services, integrating application project definition, source code management, test management, and release approvals. Track and report on development progress using the included dashboard solution, providing comprehensive decision-making support to managers and directors who need the latest information at all times on key performance indicators (KPIs).

Serena Development Manager helps you to conform with CMMI standards by enabling you to establish and maintain an organizational policy for planning and performing your processes and make them visible to relevant members of the organization. It helps you to create organizational expectations for establishing and maintaining baselines, tracking and controlling changes to work products under configuration management, and establishing and maintaining integrity of those baselines.

Serena Development Manager provides the following benefits. You can choose which of these options would be helpful for your organization, and configure and customize the workflow as needed based on your work practices.

- **Development project lifecycle management.** Promote development activities through lifecycles stages such as planning, code, test, and release. You can customize the development stages according to your organizational practices, define user roles and owners for each stage, and set up notifications.
- **Source configuration management**. Development Manager out-of-the-box provides the capability of using two different Configuration Management tools to manage your source code:
  - **Dimensions CM**, which enables you to control your source code for development tasks and baselines for development packages.
  - **ChangeMan ZMF** which enables you to control your source code for development tasks (but not development packages).
- **Test management system integration**. Monitor your QA team’s progress on defects and requirements by integrating Development Manager with HP Quality Center / ALM.

- **Graphical business intelligence dashboard**. As an option, Serena provides a rich, graphical reporting tool, Serena® Dashboard that allows you to carefully monitor key performance indicators (KPIs). The reports are fully configurable, and can provide graphical information on any aspects of your projects that managers or executives need to track.

# Using the Development Manager Components

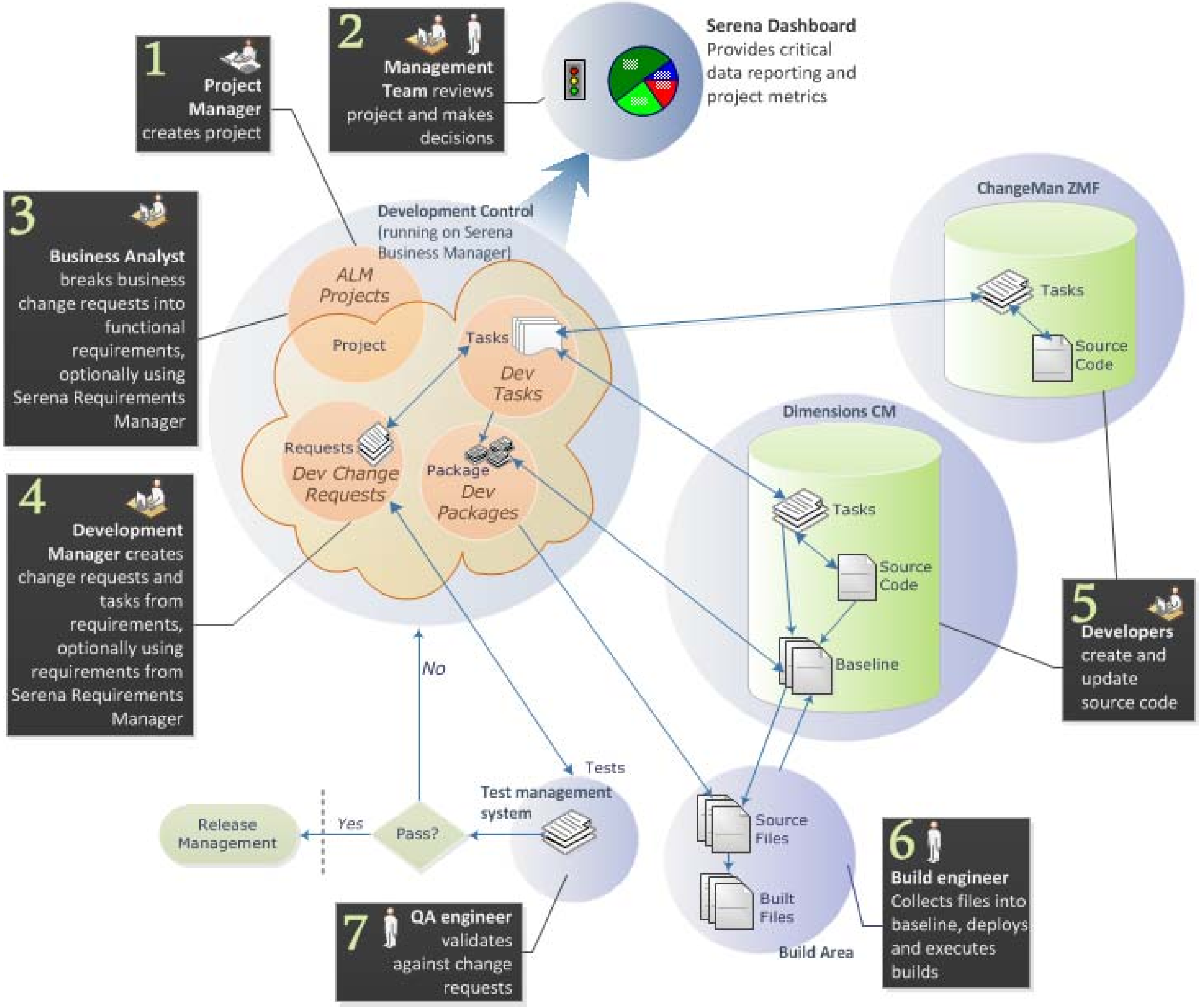
Serena Development Manager includes the following:

- ***Development Control,*** a collection of process apps implemented on the Serena Business Manager platform. A process app is a Web application, hosted on a Serena Business Manager server, that you can log into and use from any supported Web browser. With the included process apps, Development Control orchestrates your development work across all of your inter-related systems. The included process apps are:
  - **ALM Projects**: Use this app to define and track development projects. This process app sets the workflow for all projects, including states for planning, development, testing, and release. As a project progresses through the workflow, different users work with it using a combination of their own systems.
  - **Dev Change Requests**: Define and track development requests. You can relate requests tasks in the Development Tasks process app, which you can then relate to source code assets in Dimensions CM or ChangeMan ZMF. You can also relate specific types of requests, such as defects, to test cases in your test management system.
  - **Dev Tasks**: Define and track development tasks. Manage tasks by relating them to change requests. Track source code changes by relating tasks to source files in Dimensions CM or ChangeMan ZMF.
  - **Dev Packages**: Define and manage development packages. Integrate with your SCM tool to compile and build source code, and integrate with your test management system to test builds before preparing them for release.
- **Serena® Dimensions® CM**, a rich enterprise-class version and source code configuration management solution. Dimensions CM provides a wide variety of clients and usage models, with Web and desktop clients, as well as complete scripting and API support. Development Control integrates with Dimensions CM via Web services, enabling you to associate source code assets in the Dimensions CM repository with your tasks in Development Control.
- **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 Requirement Control (optional),*** a collection of process apps implemented on the Serena Business Manager platform that interfaces with Serena Dimensions RM. This enables you to manage and approve requirements. The additional process app it provides is:
  - **Req Approval**: This is used in Development Manager to associate requirements with dev change requests.

- **Serena Release Manager *(optional)*,** 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 (optional)***, a powerful, highly configurable reporting solution built on the IBI WebFocus platform. With Serena Dashboard, you can build dashboards of graphs that provide precisely the information you, your managers, and your executives need, when they need it.

## Usage Overview

The components included with Serena Development Manager work together as defined in the following diagram. This describes a standard configuration, as well as the order in which users interact with it. This only focuses on the end user scenario; configuration is addressed in the *Serena Development Manager Installation and Configuration Guide*.


1 Project Manager creates project
2 Management Team reviews project and makes decisions
Serena Dashboard
Provides critical data reporting and project metrics
3 Business Analyst breaks business change requests into functional requirements, optionally using Serena Requirements Manager
Development Control (running on Serena Business Manager)
ALM Projects
Project
Tasks
Dev Tasks
Requests
Dev Change Requests
Package
Dev Packages
ChangeMan ZMF
Tasks
Source Code
Dimensions CM
Tasks
Source Code
Baseline
4 Development Manager creates change requests and tasks from requirements, optionally using requirements from Serena Requirements Manager
5 Developers create and update source code
No
Tests
Test management system
Release Management
Yes
Pass?
Source Files
Built Files
Build Area
6 Build engineer Collects files into baseline, deploys and executes builds
7 QA engineer validates against change requests

Lets step through this diagram.

| Step | Description |
|---|---|
| 1 | At the beginning of a project, a project manager uses the ALM Projects process app in Development Control (running on Serena Business Manager) to create a new project. This is typically in response to incoming demand, such as requirements from a business analyst for a new feature, or a customer request or defect. For details on the Development Project workflow, please see "ALM Projects Workflow" on page 19. For details on creating projects, please see "Creating and Managing Projects" on page 31.<br>If you are using the additional separately licenced Serena Requirements Manager, the requirements that drive the development process will be managed, approved and assigned using this application. For details see "Using the Requirements Manager Integration" on page 57. |
| 2 | At any point during a project's lifecycle, leads, managers, executives, and others may consult Serena Dashboard to review project status and key performance indicators (KPIs). The reports displayed here may help decision makers choose the correct path forward when work must be prioritized or re-evaluated. For more on viewing and customizing Serena Dashboard views, see "Displaying Project Metrics with the Serena® Dashboard" on page 54. |
| 3 | The Business Analyst creates functional requirements based on business change requests.<br>If you are using Serena Requirements Manager, the Business Analyst submits these requirements for review and approval. When they are approved, he generates distribution tasks and assigns them to the Development Manager who uses them to create development change requests. To learn about using Requirements Manager with Development Manager, see "Using the Requirements Manager Integration" on page 57. |
| 4 | The Development Manager creates development change requests using the Dev Change Requests process app - or from the project in ALM Projects. The change requests describe the features and other work to be implemented. The change requests are related back to the project, ensuring complete traceability of work. The Development Manager and others also create tasks using the Dev Tasks process app. Tasks can be used to split the work into more manageable units, that can be assigned to individual developers. When you create a development task, the task is synchronized to Dimensions CM, and a new request of type Task is created.<br>To learn about the detailed workflow for change requests and tasks, please see "Dev Change Requests Workflow" on page 20 and "Dev Tasks Workflow" on page 21. For information on creating change requests, please see "Creating and Working on Change Requests" on page 36. For information on working with tasks, please see "Creating and Working on Tasks" on page 40. |
| 5 | Developers update the source code using their source control environment. Serena Development Manager includes Serena Dimensions CM. Tasks in Development Control are synchronized to Dimensions CM, and information on all work on files in Dimensions CM is stored in Dimensions CM tasks. This information is then synchronized back to the originating tasks (which are in turn related back to the originating change requests) in Development Control, ensuring a complete audit path of all work completed in context of a project. |

| Step | Description |
|---|---|
| 6 | As work progresses on the project, the build engineer sets up packages using the Dev Packages process app in Development Control. The build engineer relates the packages to baselines in Dimensions CM that collect all of the files associated with change requests in the project and deploy them to a build area. Using these files, the build engineer runs a build and installs it for testing purposes. To learn more about the workflow for the Development Baselines process app, see "Dev Packages Workflow" on page 24. For details on creating and working with development packages, see "Creating and Working with Development Packages" on page 44. |
| 7 | Using HP Quality Center / ALM, the QA staff tests the builds, both the nightly builds and release candidate builds. Defects may be tracked in the Dev Change Requests process app, and related to defects in Quality Center. Failed tests are returned to Development Control and the original change requests are returned to developers to fix. When a release candidate build passes testing, the build is turned over to the release engineer who will use a release management solution, such as Serena Release Manager, to deploy the build into all of the required environments.<br>For details on implementing the SBM Connector to Quality Center, see the *SBM Connector for HP Quality Center / ALM Implementation Guide*. |

# Detailed Example Use Case

We will walk through a scenario that demonstrates what Development Manager can do for you. Keep in mind that this is just an example, and that these steps can all be customized to satisfy the goals of your organization. Customization is one of the key benefits of Serena solutions built on Serena Business Manager; you can adapt the workflows, forms, integrations, and reports to meet your business needs.

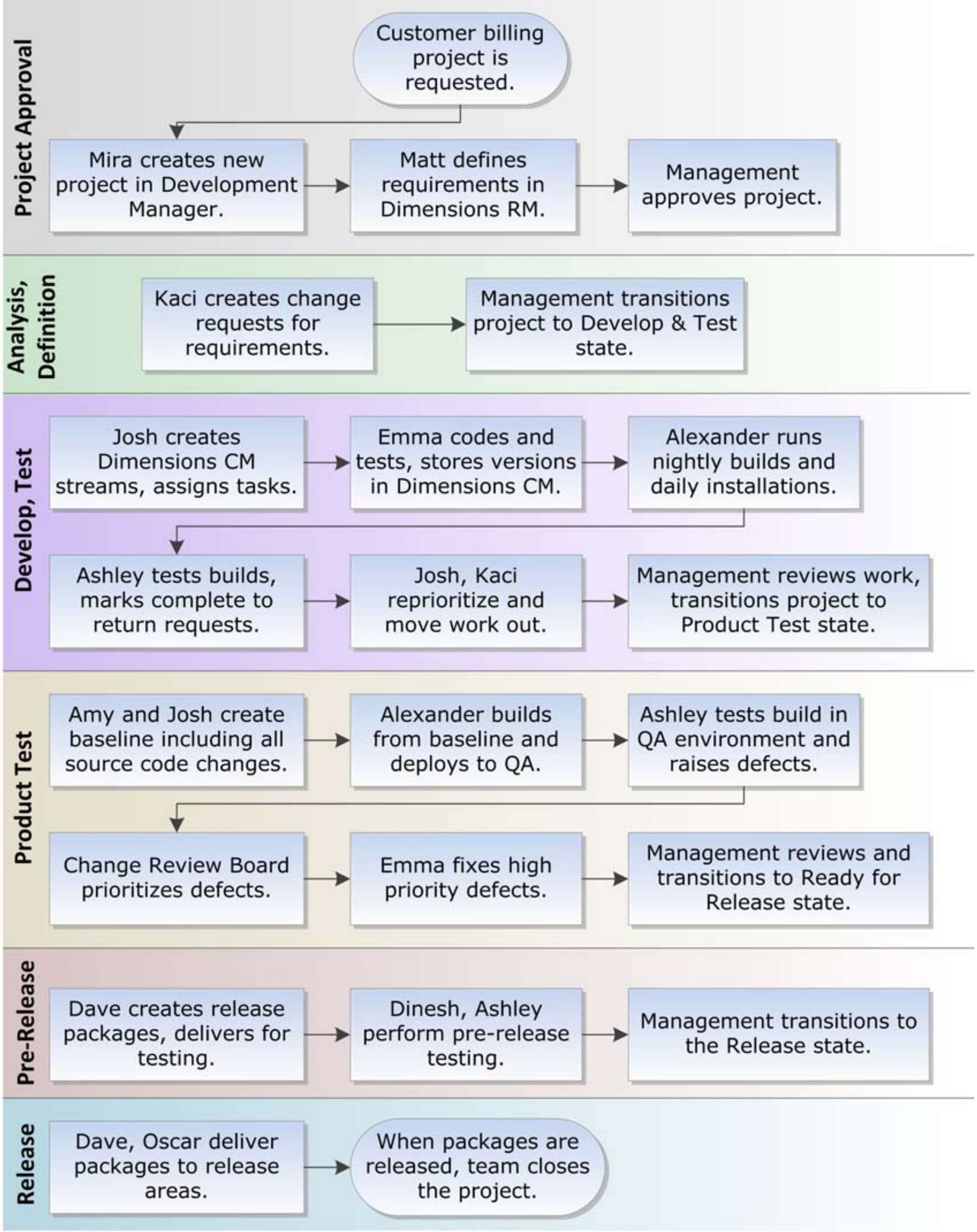
## User Roles

Serena Development Manager is designed to meet the needs of key participants in a standard development effort. Roles in your organization may vary. This example scenario includes the following users:

- Mira: Project Manager
- Matt: Business Analyst
- Kaci: Product Owner
- Josh: Development Manager
- Emma: Development Engineer (member of Development Team)
- Amy: Build Manager (member of Build Team)
- Alexander: Build Engineer (member of Build Team)
- Dinesh: QA Manager (member of QA Team)
- Ashley: QA Engineer (member of QA Team)
- Dave: Release Manager (member of Release Team)
- Oscar: Release Engineer (member of Release Team)

## Process Example

The following flowchart presents an example scenario using the personas described above. Details about each step follow the diagram.


Project Approval
Customer billing project is requested.
Mira creates new project in Development Manager.
Matt defines requirements in Dimensions RM.
Management approves project.
Analysis, Definition
Kaci creates change requests for requirements.
Management transitions project to Develop & Test state.
Develop, Test
Josh creates Dimensions CM streams, assigns tasks.
Emma codes and tests, stores versions in Dimensions CM.
Alexander runs nightly builds and daily installations.
Ashley tests builds, marks complete to return requests.
Josh, Kaci reprioritize and move work out.
Management reviews work, transitions project to Product Test state.
Product Test
Amy and Josh create baseline including all source code changes.
Alexander builds from baseline and deploys to QA.
Ashley tests build in QA environment and raises defects.
Change Review Board prioritizes defects.
Emma fixes high priority defects.
Management reviews and transitions to Ready for Release state.
Pre-Release
Dave creates release packages, delivers for testing.
Dinesh, Ashley perform pre-release testing.
Management transitions to the Release state.
Release
Dave, Oscar deliver packages to release areas.
When packages are released, team closes the project.

| Project Phase | Description | Tool |
|---|---|---|
| **Project Approval** | Matt (*Business Analyst*) asks Mira (*Project Manager*) to create a new project for the next release of the customer billing system. Mira creates the project in the ALM Projects process app in SBM. This project will be used to coordinate development activities. | ALM Projects |
| | Matt is notified that the project has been created, and uses his requirements management tool to define detailed requirements. He adds requirement links to the project. | External requirements management tool |
| | The management team (Mira, Matt, Kaci, Josh, Dinesh, Amy, and Dave) reviews the requirements and works through any open questions. When the requirements are approved, they promote the project to the next phase, *Analysis and Definition*. | ALM Projects |
| **Analysis and Definition** | Kaci (*Product Owner*) reviews the detailed requirements and creates software change requests for each of them. Her goal is to break the requirements into small, clearly defined units of work. | ALM Projects |
| | The management team reviews the requirements, stories, and testing requirements. When the requirements and change requests are approved, they promote the project to the next state, *Develop and Test*. | ALM Projects |
| **Develop and Test** | Josh (*Development Manager*) reviews the requests and works with Kaci to prioritize the work. Josh also creates the development streams in Dimensions CM where the source code will be managed. | ALM Projects |
| | Josh, Kaci, Emma (*Software Engineer*), and Ashley (*QA Engineer*) break the requests into development tasks. As they are created, the tasks are synchronized to Dimensions CM requests. Once the requests and tasks are ready to work on, Josh moves them into development. Ashley and Emma can now begin work. | ALM Projects |
| | Emma develops and tests code changes for each of her tasks. She works with the tasks in Dimensions CM via the Dimensions CM integration to Microsoft Visual Studio. She stores new versions of source files in Dimensions CM. Meanwhile, Ashley defines test cases that can be used to test each story as it is completed. | Dimensions CM/ ChangeMan ZMF |
| | Alexander (Build Engineer) manages nightly builds and daily installation on the QA test environments. | External build tool |
| | When she completes all of the tasks for a change request, Emma assigns that request to Ashley for testing. | Dimensions CM/ ChangeMan ZMF |
| | Ashley runs the test cases for the request that Emma finished coding. Ashley uses the installation that Alexander runs from the nightly build. Ashley returns any request that fails testing back to Emma to fix. | HP Quality Center / ALM |
| | Once she has finished testing a request, Ashley marks it as complete. This delegates the request back to Josh. | HP Quality Center / ALM |
| | Josh and Kaci continually review the list of requests in progress, and move work out of the project as needed based on remaining time and business priorities. | ALM Projects |
| | Once the requests have all been developed and tested, the management team reviews the requirements, requests, and test results. When they approve the completed work, they move the project to the next phase, *Product Test*. | ALM Projects |

| **Project Phase** | **Description** | **Tool** |
|---|---|---|
| **Product Test** | Amy and Josh create a package in the Dev Packages process app, and then relate that package to a baseline in Dimensions CM. The baseline assembles all of the source code changes associated with completed requests. Amy delegates the package to Alexander who then builds the source associated with the Dimensions CM baseline. He collects the build outputs back into Dimensions CM, and deploys them to the QA testing environments. | ALM Dev Packages, Dimensions CM |
| | Josh assigns the turnover package to Dinesh (*QA Manager*) to coordinate testing. | ALM Projects |
| | Dinesh prioritizes testing requirements for the project. | HP Quality Center / ALM |
| | Ashley runs test cases against the QA environments and raises defects as she finds issues. | HP Quality Center / ALM |
| | The change review board determines which defects need to be fixed. Defects that need to be fixed are delegated to Emma. Deferred defects are are moved out of the project to be fixed at a later date. | ALM Projects |
| | As Emma fixes defects, new packages that include the fixes are forwarded to QA. Ashley re-tests the defects. | HP Quality Center / ALM |
| | Once all of the requirements and defects have been tested, the management team reviews the requirements, stories, package(s), and test results. Once they approve the completed work, they promote the project to the next phase, *Ready for Release*. | ALM Projects |
| **Ready for Release** | Using Serena Release Manager, Dave (*Release Manager*) integrates the packages into the release cycle. He works with Oscar (Release Engineer) to get the package delivered to the appropriate pre-release areas for testing. | Dev Packages |
| | Dinesh coordinates the pre-release testing and Ashley raises defects as necessary. | HP Quality Center / ALM |
| | Once the packages have been successfully tested, the management team does a final review of the requirements, completed requests, packages, and test results. Once they approve the release, they promote the project to the next phase, *Release*. | ALM Projects |
| **Released (inactive)** | Using Serena Release Manager, Dave works with Oscar to deliver the package to the release team. | Release Control |
| | Once the packages have been successfully released and the management team approves the final release, they close the project. | ALM Projects |

Chapter 2

# Understanding the Out-of-the-Box Workflows

# About the Out-of-the-Box Development Workflows

Serena Development Manager includes rich out-of-the-box workflows as part of Development Control, a collection of process apps that run on Serena Business Manager. Development Control integrates development processes with versioning capabilities provided by Serena Dimensions CM, and a set of out-of-the-box reports provided by Serena Dashboard. Development Control satisfies use cases specific to the development states of a project, including analysis and design, development, testing, and turnover for release preparation.

Out-of-the-box, Development Control provides workflows for the following:

- Development projects, implemented by the ALM Projects process app.
- Development change requests, implemented by the Dev Change Requests process app.
- Development tasks, implemented by the Dev Tasks process app.
- Development baselines and packages, implemented by the Dev Packages process app.

# Object Types and Relationships

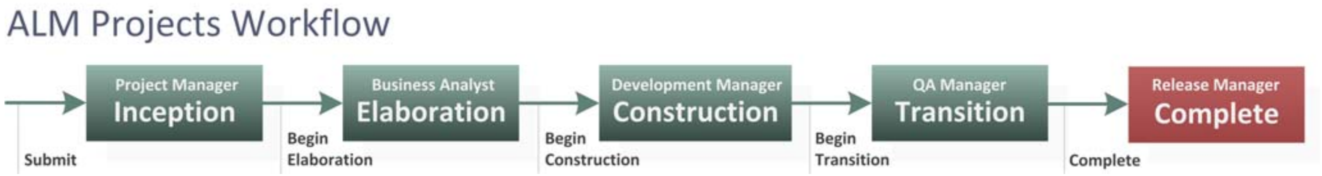
The SBM process apps included with Development Control enable you to manage projects, change requests, tasks, and release packages. Each of these object types can be related to specific items in external systems.

- **Projects**: You manage projects using the ALM Projects process app. You can relate projects as follows:
  - By relating a project to Dimensions CM streams and projects, you enable the association of source code and other managed assets related to the project.
  - By relating a project in ALM Projects to change requests in the Dev Change Requests process app, you ensure that all work to be completed in context of the project is associated with the project. You also provide scope for the creation of Dimensions CM tasks and items that are needed for this project.
  - By relating a project to packages in the Dev Packages process app, you ensure that all completed work included in releases are associated with the project.
- **Change Requests**: You manage change requests using the Dev Change Requests process app. You can relate change requests as follows:
  - Relate change requests to tasks in the Dev Tasks process app. You can then manage all development work from the related tasks.
  - If you use HP Quality Center / ALM to manage defects, associate requirements and defects to defect type change requests in Development Control.
  - If you use Serena Requirements Manager, associate requirements to dev change requests. You can associate multiple requirements to a single dev change request or multiple dev change requests to a single requirement.

- **Tasks**: You manage tasks using the Dev Tasks process app. You can relate tasks as follows:
    - Relate tasks to change requests in the Dev Change Requests process app to maintain a history of which tasks were associated with the requests.
    - Relate tasks to requests in Dimensions CM in order to track specific code changes. All updates to files stored in Dimensions CM are tracked in Dimensions CM tasks; this information is then synchronized to tasks (and their associated requests) in the Dev Tasks process app.
- **Development packages**: You manage development packages using the Dev Packages process app. Packages define the scope of a release by collecting all included change requests and tasks into one place. You can relate packages as follows:
    - Relate packages to baselines in Dimensions CM. Dimensions CM baselines collect all source code (or compiled deliverables) associated with a particular project or release. Those baselines can be deployed, using Dimensions deployment features or using Serena Release Manager.
    - Packages are also related to specific change requests. The change requests define the scope of the code included in the baseline.

# ALM Projects Workflow

ALM Projects provides a high level view of the overall project workflow and status.



**The states include the following:**

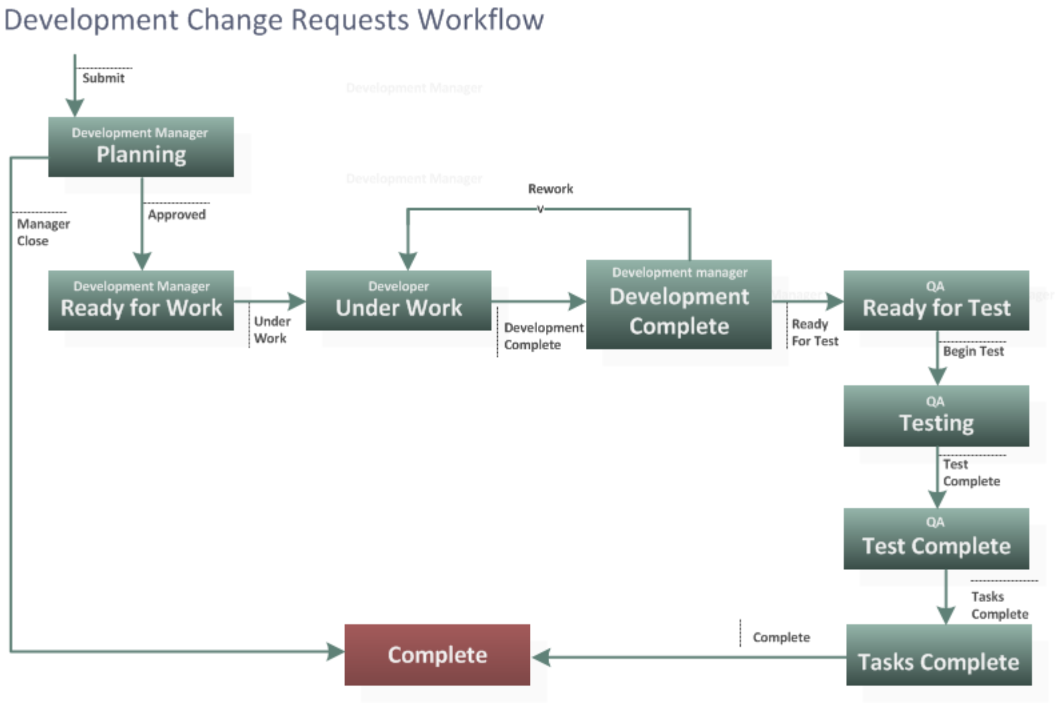
1. **Inception**: During this state, the Project Manager creates the project.
2. **Elaboration**: During this state, the Business Analyst defines change requests for the project based on the approved requirements.
3. **Construction**: At this point, the Development Manager or Lead prioritizes the change requests and creates tasks related to the requests. The tasks are assigned to developers to implement. Developers work on the tasks, storing new versions of files in their configuration management system (such as Dimensions CM). Regular builds are compiled and installed for testing, and requests are assigned to QA to test. Once all of the requests have been implemented, the Development Manager can transition the project to the next state.
4. **Transition**: During this state, QA perform robust testing of the completed product or features, recording defects as they find them. QA may use a test management system such as HP Quality Center to track test case execution. QA submits defects to the Change Request process app, and the defects are assigned to developers to fix. QA validates fixes and closes defects as they are resolved. When all critical defects are complete, QA can transition the project into the Complete state.

5 **Complete**: During this state, the finished product is prepared for release. The final builds are collected and packaged for deployment, perhaps using Serena Release Manager. When the product is ready to deploy or release, the project can move to the final state.

# Dev Change Requests Workflow

Change requests follow the workflow illustrated below. This workflow is implemented in the Dev Change Requests process app. This illustration is exported directly from SBM. Note that most work on change requests is expect to happen during the **Develop & Test** state in a project.

## SBM Workflow for Change Requests



1 **Planning**: During this state, the Product Owner and Development Manager define the content of change requests based on the requirements for the project. Once the request is approved, it can be moved into the **Ready for Work** state.

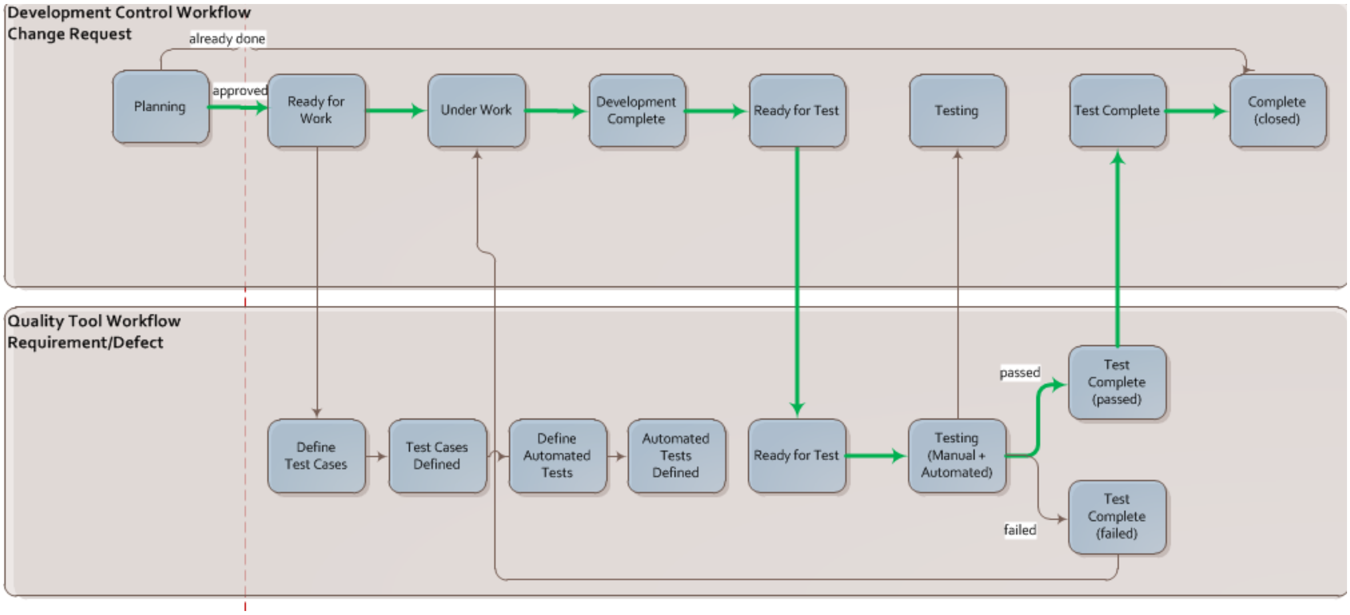
2 **Under Work**: Developers can now work on tasks associated with the request. As work is complete the developer checks updated source code files into Dimensions CM. Once this work associated with the request (and its related tasks) is ready to be compiled and tested, the request is transitioned by the developer to the next state.

3 **Development Complete**: At this state, the build engineer starts a build using the code that was created or updated during the **Under Work** state. When the build successfully completes and the request is ready to test, the build engineer transitions the request to the next state.

4 **Ready for Test**: The QA staff can now test the code that was developed and compiled to satisfy this change request. The QA manager or responsible QA engineer transitions the request to the **Testing** state. For dev change requests of type defect, testing can then be managed in HP Quality Center. If a test fails in the test case management system, the request is returned to the **Under Work** state. When testing is complete, the request can be transitioned to the **Test Complete** state, and then to the **Complete** state.

## Integrating Change Requests with HP Quality Center / ALM

The following diagram illustrates the out-of-the-box integration points between Dev Change Requests and HP Quality Center / ALM.

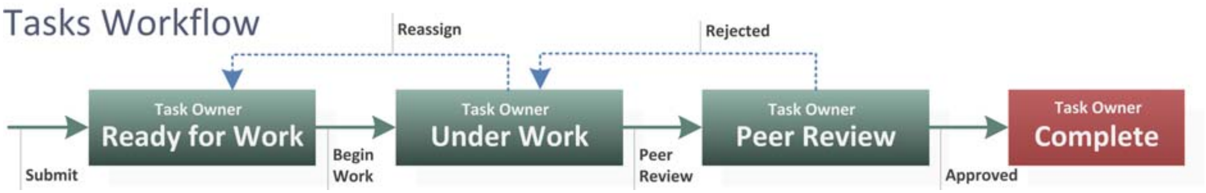


The test states in the change request workflow are mapped to workflow states in the Quality Center / ALM workflow. When the change request enters the *Ready for Test* state, it is associated with test cases in Quality Center.

# Dev Tasks Workflow

Manage specific developer tasks using the Tasks process app. You start and complete work on a task during the **Under Work** state of the change request that owns it. Development Managers create tasks to break work associated with a change request down into more manageable units that can be assigned to individual developers.

Tasks follow the workflow illustrated below. This illustration is exported directly from SBM. If you are using Dimensions CM or ChangeMan ZMF, these states map to parallel states for change requests in Dimensions CM or change packages in ChangeMan ZMF.



The states include the following:

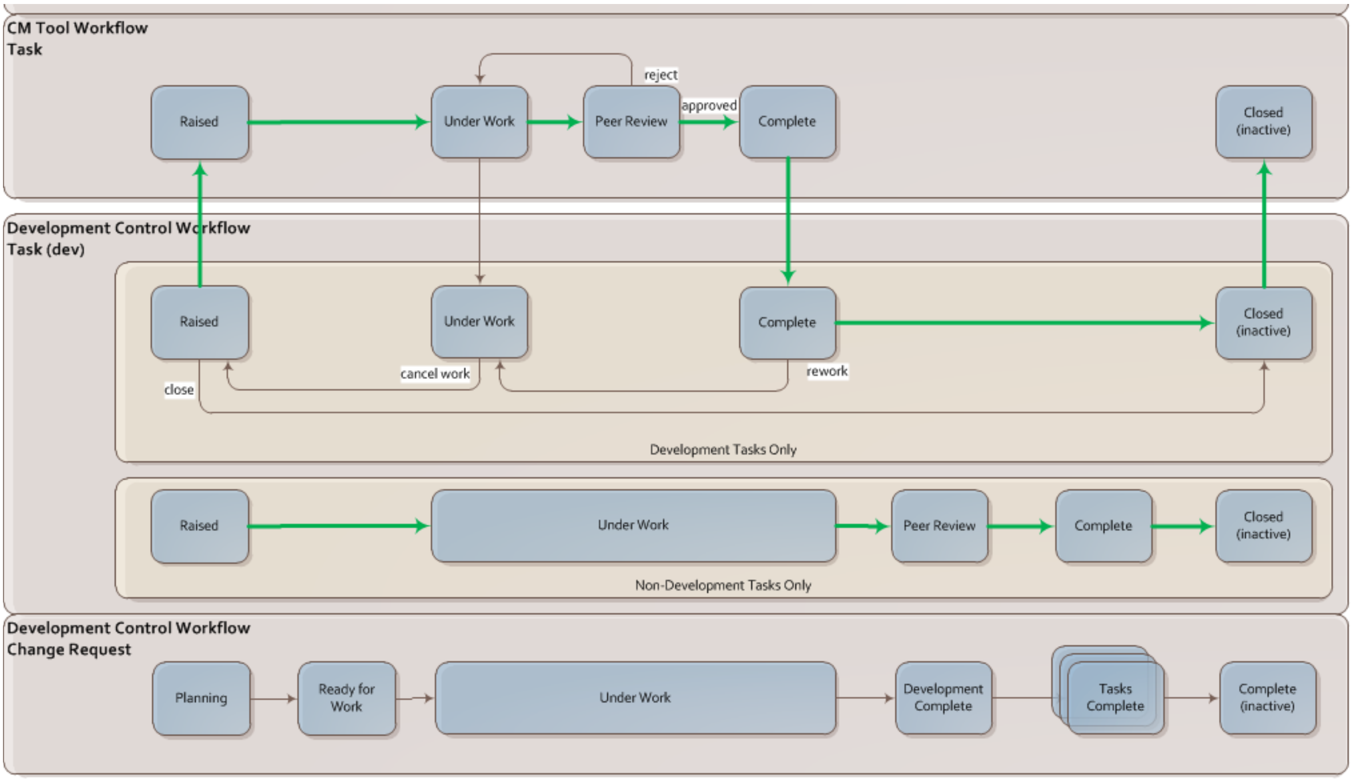
1 **Raised**: The task is submitted. If you are using Dimensions CM, a related change request is also created in Dimensions CM or ChangeMan ZMF. To start work, the Development Manager assigns an owner and transitions to the **Under Work** state. The related item in Dimensions CM or ChangeMan ZMF is also transitioned to the parallel **Under Work** state in Dimensions CM or **Development** in ChangeMan ZMF.

2 **Under Work**: As work is complete the developer checks updated source code files into Dimensions CM or ChangeMan ZMF. Information about the new and updated files is stored in the application, and that data is synchronized back to the task in Dev Tasks. All information about affected files is therefore captured in context of the change request or change package that the task belongs to.

3 **Peer Review**: A software development best practice is to include review of changes by another developer before turning the new or updated code over for testing. Once the peer has approved the changes, the the Dimensions CM task can be actioned to the **Complete** state. At this point the corresponding dev task in Development Control is also transitioned to the **Complete** state.

## Task Workflow Using Dimensions CM

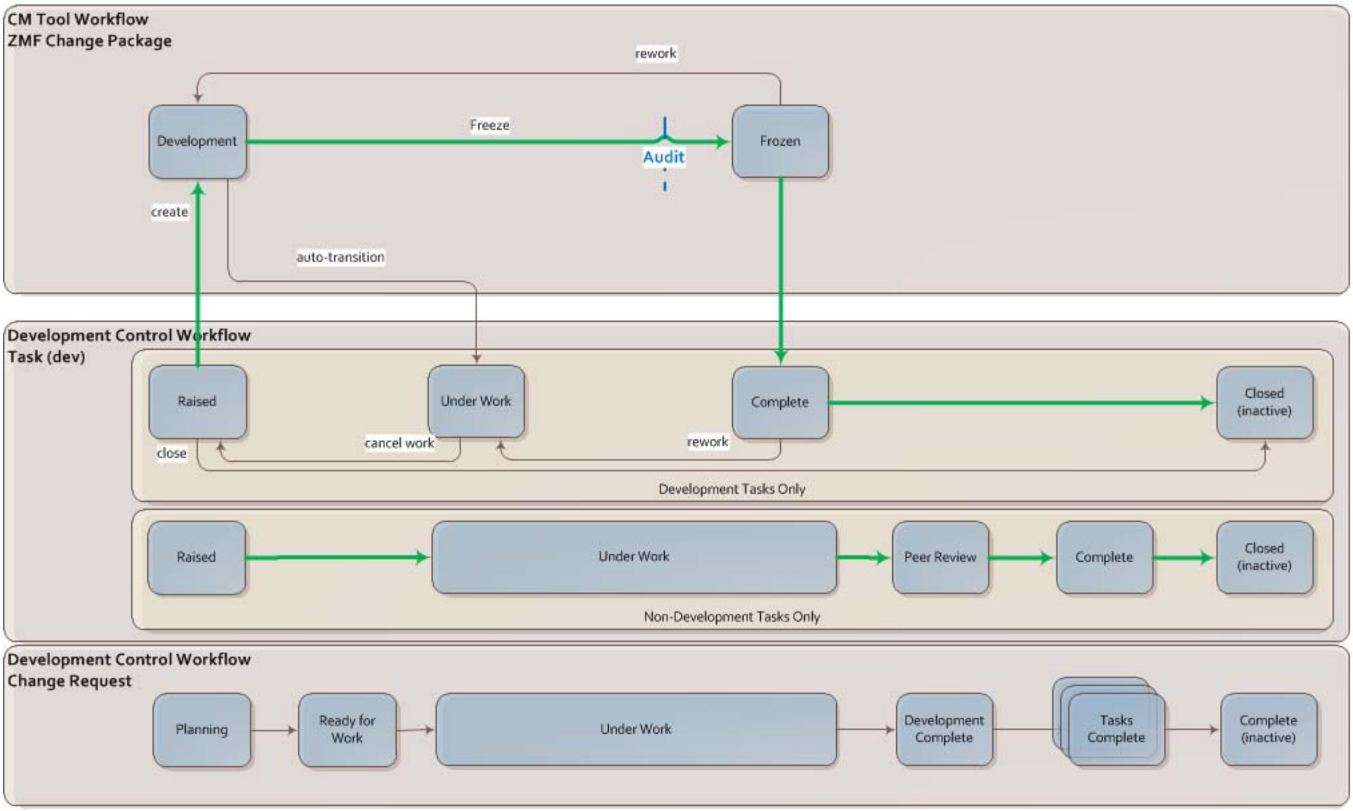
The following illustration clarifies the relationship between tasks in Dev Tasks and related tasks in Dimensions CM. Note that this is just one example of how tasks in Development Control might be linked to tasks in Dimensions CM for purposes of tracking work.



In this example, the associated task in Dimensions CM follows a parallel workflow. The request may be a specific type that is designed to synchronize with tasks in the Dev Tasks process app. In this example, the task in Dev Tasks is transitioned to the completed state once the related task in Dimensions CM is completed, requiring all work completed in Dimensions CM to be peer reviewed and approved before the task can be closed. Transitioning the request in Dimensions CM to the **Under Work** and **Complete** states in turn automatically transitions the related task in Dev Tasks.

## Task Workflow Using ChangeMan ZMF

The following illustration shows the relationship between tasks in Dev Tasks and the corresponding change package in ChangeMan ZMF. In this case, the change package is under development until work is complet, when it is frozen. In order to be frozen it must have first been audited. Freezing the change package in ChangeMan ZMF automatically transitions the task in Dev tasks to **Complete**.

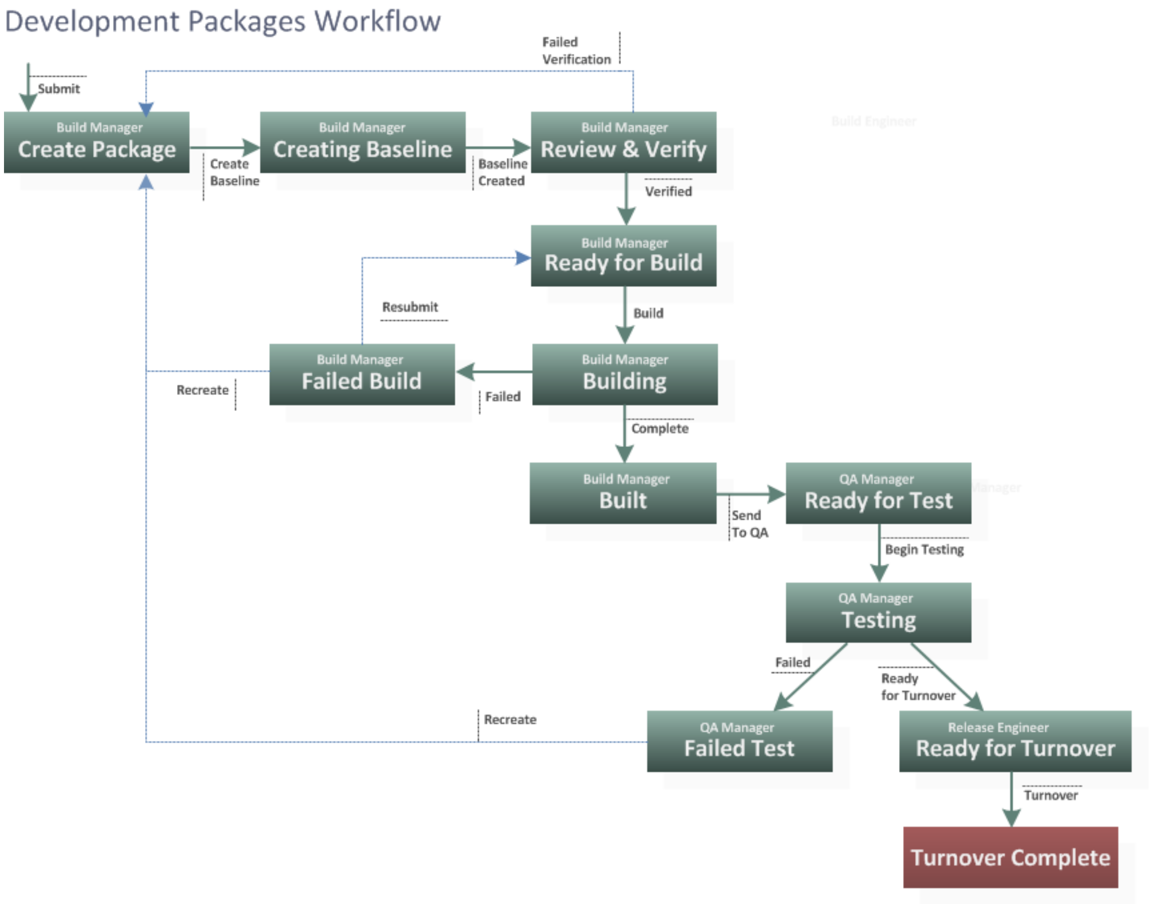


# Dev Packages Workflow

Use the Dev Packages process app to manage the creation and validation of release packages and baselines, including testing builds and approving for final turnover. Most of the work managed within the Dev Packages process app takes place during the **Product Test** project state.

The Dev Packages workflow provides integration points into Dimensions CM, as well as Quality Center. Use Dev Packages to fully automate the build and validation process. Start by building from baselined sources and finish by passing the final build off to your release management system, such as Serena Release Manager.

# Dev Packages Workflow Diagram



The workflow states include the following:

1 **Create Package**: The release engineer creates a package that defines the scope of a release. The package includes links to change requests that define the content of the package.

2 **Creating Baseline** and **Review and Verify**: A baseline is created in Dimensions CM that includes all of the files that are to be included in the release. The baseline is related to the package in Dev Packages, and the baseline follows a parallel workflow. Once the baseline and package are approved for testing, the package is transitioned to the next state.

3 **Ready for Build** and **Building**: The build manager compiles the build from the package that the release engineer turned over. If the build fails, the release engineer reviews and updates the package as needed. The baseline may need to be recreated and the build run again. If the build succeeds, the build manager transitions the package to the **Built** state, and then deploys the compiled deliverables for QA to test.

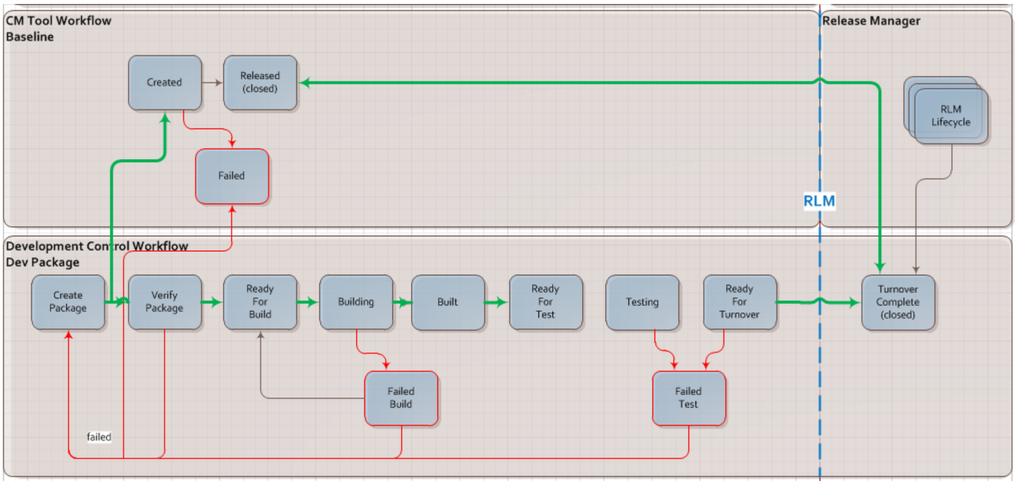
4 **Ready for Test** and **Testing**: QA staff test the deployed build. Information about the packages is synchronized to requirements in the test management tool. Test cases for validating those requirements are associated with the requirements. QA run their tests and update with pass / fail information, which is then synchronized back to the packages in Dev Packages. If a test fails, then either the build, baseline, or source code will be need to be updated to correct the defect. The team must review the

results and decide who will fix any issues that the testing reveals. If the source code is updated, the build manager must re-generate the baseline, and re-run the build and deploy it. When testing is complete and the build is approved for turnover for release, QA transitions the package to the **Ready for Turnover** state.

5 **Ready for Turnover**: At this state, the release management process takes over. For example, you may use Serena Release Manager to coordinate deployment.

6 **Turnover Complete**: If you are using the integration between Serena Release Manager and Development Manager, a turnover in the form of a Dimensions CM baseline is published to Serena Release Vault. This can be input to a release and deployment process in Serena Release Manager.

The following diagram illustrates the integration points between the Dev Packages process app and Dimensions CM. This is based on the out-of-the-box configuration. In this example, a baseline in Dimensions CM follows a separate workflow. As the package progresses through its workflow states, information and state are synchronized to the related baseline in Dimensions CM.



# Supporting Your Development Process

Serena Development Manager is a powerful, highly customizable lifecycle management and reporting application. To make the best use of it, spend some time evaluating your own development processes. It is important to decide what processes and issues Serena Development Manager can help you solve, to set expectations appropriately and begin to plan roll-out and adoption. Consider creating a diagram of your development lifecycle if you do not already have one, as this can be an valuable resource when mapping your workflows in Serena Business Manager.

## What Types of Development Projects Will You Manage?

In Serena Development Manager, you can define project types that correspond to specific development management needs in your organization. For example, you can define different project types for different applications, or for different departments. Consider the following when deciding what types of project you will support:

- Will Serena Development Manager support projects for different departments? Do the departments have their own lifecycles?
- What applications does your organization use, and which of these will you integrate with Serena Development Manager? For example, does your development team use Serena Dimensions CM for source code management, or another system? What about Requirements and Test Case Management?

### Best Practices for Development Project Management

- Use the Dev Change Request process app to manage defect reports alongside other change requests. If you also use HP Quality Center for defect management, you can synchronize your defects between the two systems.

## Is Your Development Process Agile?

Determine what type of development practice your project types follow. Do they follow iterative Agile methodologies? Is their process closer to traditional "waterfall" methodologies, with heavier upfront analysis and planning?

### Best Practices for Agile Projects:

- Capture user stories with change requests using the Dev Change Requests process app. You can simply word change requests as you would word a user story.

## What Are Your Workflow States?

Identify the states in your development workflow. You can replicate your existing states, or refine your current practices when configuring the lifecycle in Serena Business Manager. Questions you might hear include:

- Who submits new requests for different types of work (such as features, defects, etc.) to your development team, and in what form?
- What is involved with design and analysis of new features, before developers actually begin work? Who is responsible?
- How is testing managed? Is it part of the development effort, completed afterwards, or both? Who manages the test states?
- What steps are required in order to certify new work for turnover for release? Consider such things as stabilization, defect burndown sprints, and deploying to staging. Who manages these steps?

### Best Practices for Defining Your Workflow

- You can customize the workflows in Serena Development Manager as needed. Study and work with the out-of-the-box workflows, itemize the changes that you need to make, and update them using Serena Business Manager Composer.
- Integrate testing into your development lifecycle. Out-of-the-box, the Serena Development Manager workflow includes a separate state for testing that occurs after development is complete. However, we strongly advocate for testing in context of development. Developers should test their own work, as well as peer review each other’s work.

## How Do You Manage Test Cases?

Development Manager provides a Web Services integration to HP Quality Center, enabling you to synchronize information from your development requests in Serena Business Manager to defects in Quality Center. Please see the *Serena Business Manager Connector for Quality Center Implementation Guide* for implementation details.

## What Reports Does Your Organization Need?

Serena provides as an option, Serena Dashboard, a powerful dashboard reporting solution that allows you to display rich graphical data on key performance indicators (KPIs). This enables your stakeholders, including team members, managers, and executives, to quickly review project status and make critical project decisions.

However you can define additional reports as needed. Consider what reports you need to define. Who needs to review project information? Which systems are involved? For more information on the out-of-the-box metrics included with Serena Dashboard, see "Displaying Project Metrics with the Serena® Dashboard" on page 54. To learn more about configuration Dashboard views, please see the *Serena Development Manager Installation and Configuration Guide*.

Chapter 3

# Getting Started Using Development Manager

# Introduction

After installing and configuring the Serena ALM components you have purchased, you can log in right away to the out-of-the-box configurations of Serena ALM. You can then start learning your way around the default process apps. A process app is an application running in Serena Business Manager with a Web-based user interface and workflow logic that coordinates the features and integrations across all included platforms. Process apps can be customized as needed to meet the needs of your organization, however right out-of-the-box you can step through the default process apps to see a mature example of a Development Manager implementation. Each process app is represented by a tab in the Serena ALM UI. You will only see the process apps for which you have the required role to access.

# Logging In to Serena ALM

To work with the Development Control, log in to Serena Development Manager using one of the default user accounts provided as part of your Development Manager Installation.

**To log into Development Control:**

1. In a supported browser application, open the URL to your Serena Business Mashups server. By default, this is:

   http://*server*:*port*/tmtrack/tmtrack.dll?shell=alm

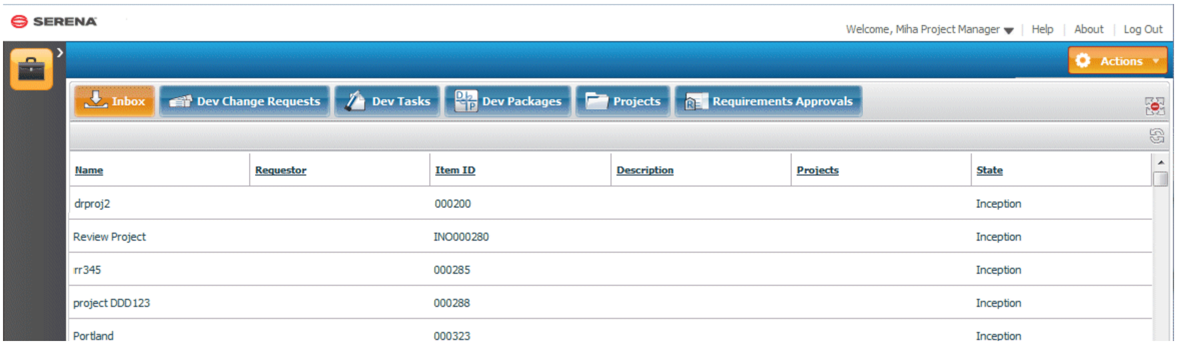
   For example, if the server is called dev and the port number is 8085, and you are using the Serena-supplied custom shell:

   ```
   http://dev:8085/tmtrack/tmtrack.dll?shell=alm
   ```

   Don’t forget the question mark at the end of the `.dll`.

2. Enter your login credentials.

You will be presented with your home page that you can customize to your needs.



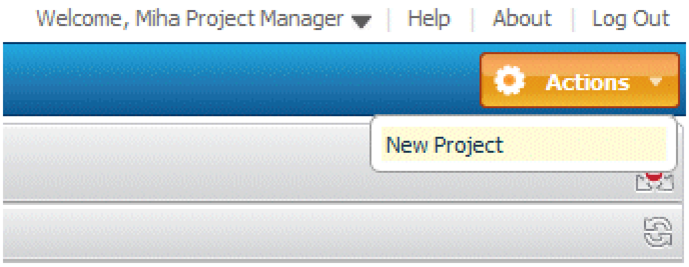
| Name | Requestor | Item ID | Description | Projects | State |
|---|---|---|---|---|---|
| drproj2 | | 000200 | | | Inception |
| Review Project | | INO000280 | | | Inception |
| rr345 | | 000285 | | | Inception |
| project DDD123 | | 000288 | | | Inception |
| Portland | | 000323 | | | Inception |

# Creating and Managing Projects

Use the **Projects** tab to view and manage your development projects. As new requests and requirements come in, create projects to manage development work. Once you create an ALM project, you can associate change requests and packages with it.

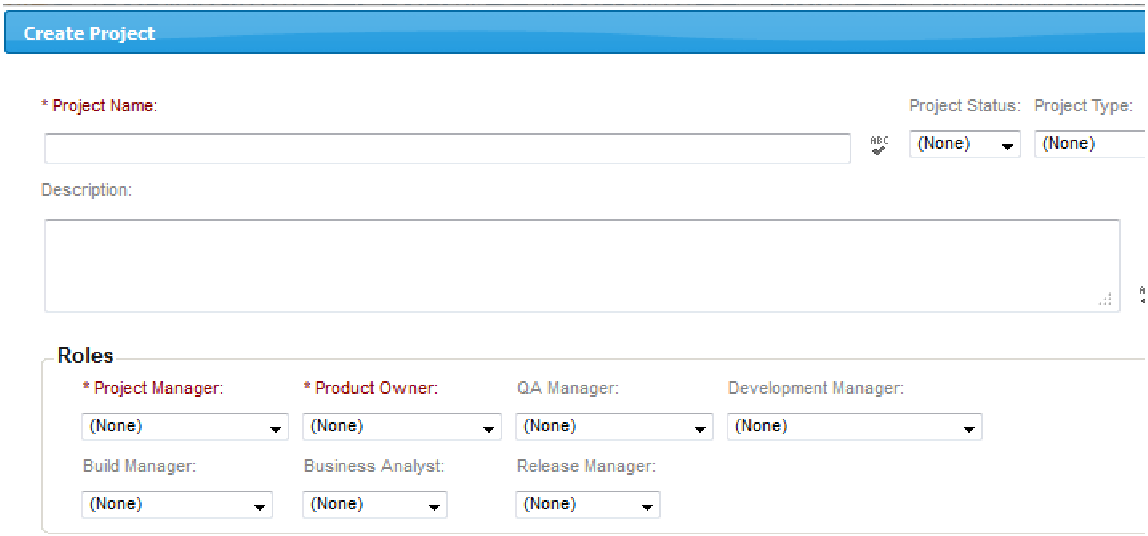
## Creating a Project

From the **Actions** list at the top right of the window, select *New Project*.



Note that you will need to be logged in as a user that has the privileges to create a project, such as Mira, the Project Manager.

A Create Project form appears:



Create Project

* Project Name:

Project Status: (None)

Project Type: (None)

Description:

Roles

* Project Manager: (None)

* Product Owner: (None)

QA Manager: (None)

Development Manager: (None)

Build Manager: (None)

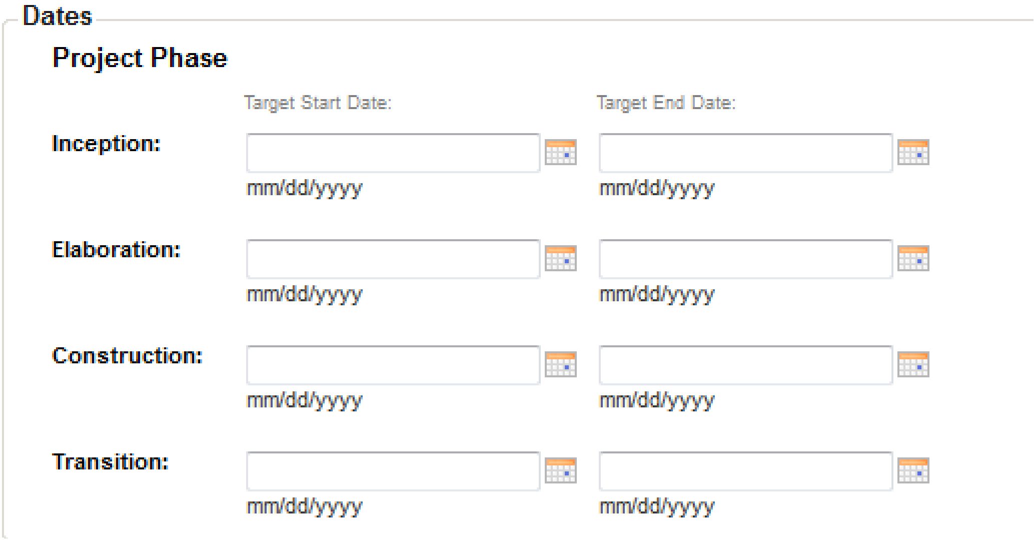
Business Analyst: (None)

Release Manager: (None)

When you create a new project, you set a number of core properties for the project, including:

- The type of project, from the **Item Type** list. Out-of-the-box, you can choose **Innovation** or **Operational**.
- The title and description.
- Key users involved with the project, such as the Project Manager, Product Owner, Business Analyst, and Development Manager.

- Milestone dates for the project. This includes target start and end dates for each stage in the project, including Inception, Elaboration, Construction, and Transition. You can set actual start and end dates later on as the actual dates are passed



Dates

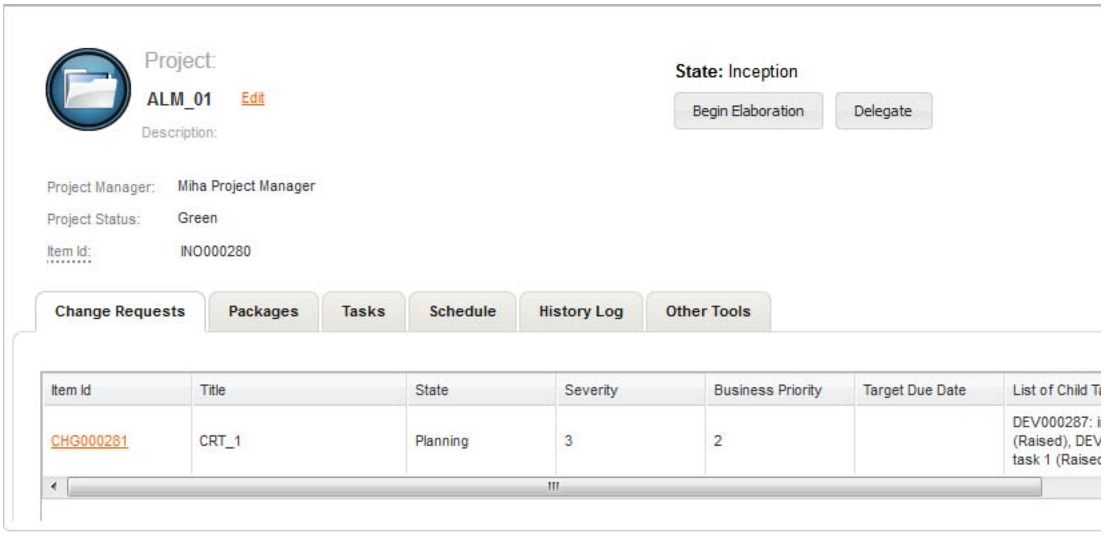
Project Phase

| | Target Start Date: | Target End Date: |
|---|---|---|
| Inception: | mm/dd/yyyy | mm/dd/yyyy |
| Elaboration: | mm/dd/yyyy | mm/dd/yyyy |
| Construction: | mm/dd/yyyy | mm/dd/yyyy |
| Transition: | mm/dd/yyyy | mm/dd/yyyy |

- Configuration Management Settings. You can select ChangeMan ZMF, Dimensions CM, or both as a Configuration Management provider in which to carry out your development work.

## Example Project in ALM Projects

Here we see an example project in ALM Projects. Clicking the Change Requests tab shows that this has a change request associated with it.



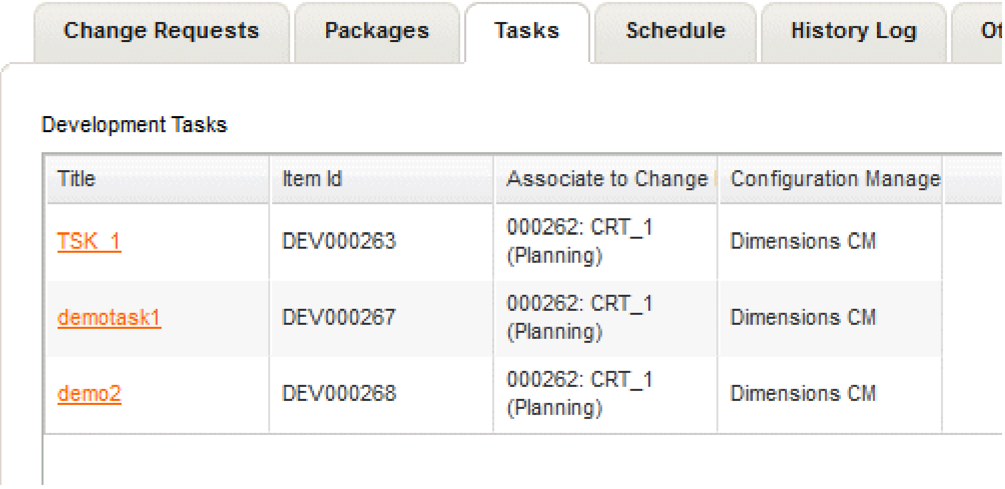
Project:
ALM_01
Description:

State: Inception

Project Manager: Miha Project Manager
Project Status: Green
Item Id: INO000280

| Item Id | Title | State | Severity | Business Priority | Target Due Date | List of Child T |
|---|---|---|---|---|---|---|
| CHG000281 | CRT_1 | Planning | 3 | 2 | | DEV000287: i (Raised), DEV task 1 (Raisec |

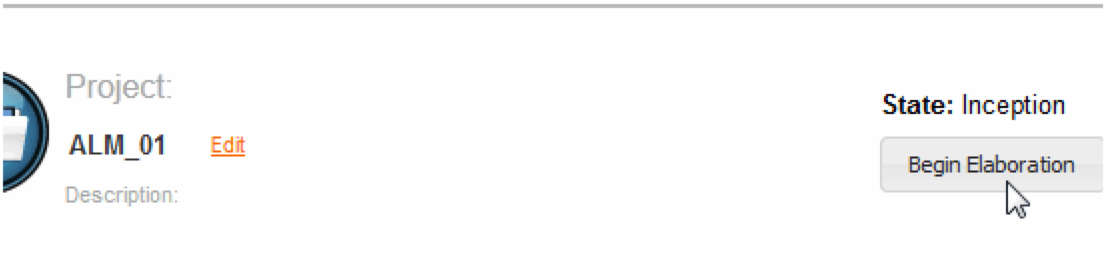
Clicking the Tasks tab shows that it has three tasks.



Development Tasks

| Title | Item Id | Associate to Change | Configuration Manage |
|---|---|---|---|
| TSK_1 | DEV000263 | 000262: CRT_1 (Planning) | Dimensions CM |
| demotask1 | DEV000267 | 000262: CRT_1 (Planning) | Dimensions CM |
| demo2 | DEV000268 | 000262: CRT_1 (Planning) | Dimensions CM |

You can transition a project to the next state in the lifecycle by pressing the appropriate transition button, such as **Begin Elaboration** or **Begin Construction**. Note that you need the required role, for example Project Manager, to transition the project, otherwise the required button will not be displayed.



Project:

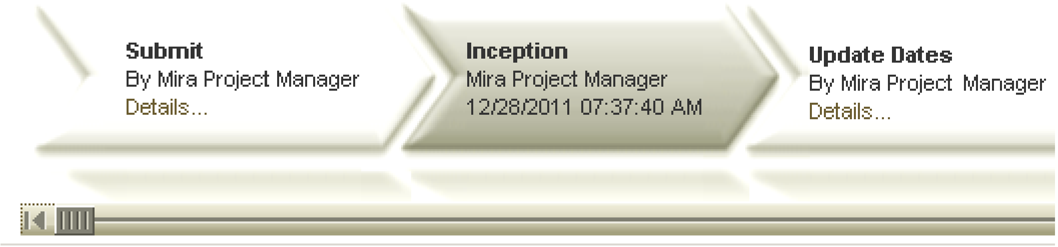
ALM_01

Description:

State: Inception

You can update details for the project by clicking the **Edit** link.

You can display transition history for a project or other Development Manager item by selecting its **History Log** tab.

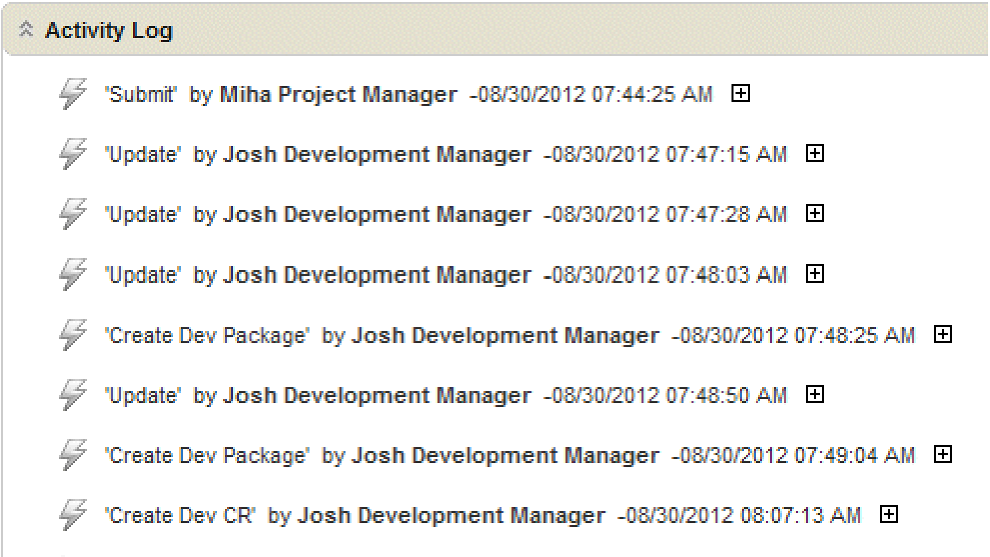


**Submit**
By Mira Project Manager

**Inception**
Mira Project Manager
12/28/2011 07:37:40 AM

**Update Dates**
By Mira Project Manager

Below the state change history you can view the history of activities by expanding the Activity Log.



Activity Log

'Submit' by **Miha Project Manager** -08/30/2012 07:44:25 AM ⊞

'Update' by **Josh Development Manager** -08/30/2012 07:47:15 AM ⊞

'Update' by **Josh Development Manager** -08/30/2012 07:47:28 AM ⊞

'Update' by **Josh Development Manager** -08/30/2012 07:48:03 AM ⊞

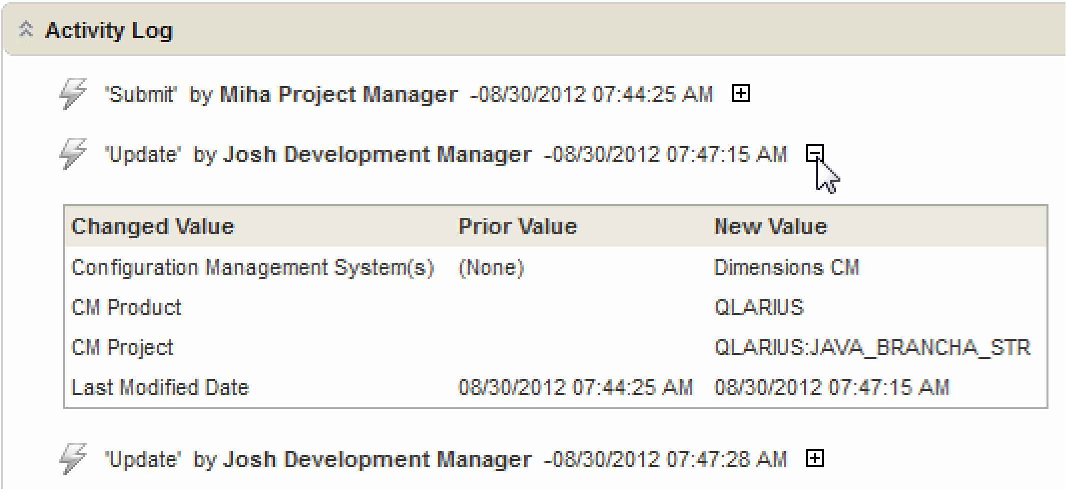
'Create Dev Package' by **Josh Development Manager** -08/30/2012 07:48:25 AM ⊞

'Update' by **Josh Development Manager** -08/30/2012 07:48:50 AM ⊞

'Create Dev Package' by **Josh Development Manager** -08/30/2012 07:49:04 AM ⊞

'Create Dev CR' by **Josh Development Manager** -08/30/2012 08:07:13 AM ⊞

You can expand an entry to reveal more details by clicking the + icon.



Activity Log

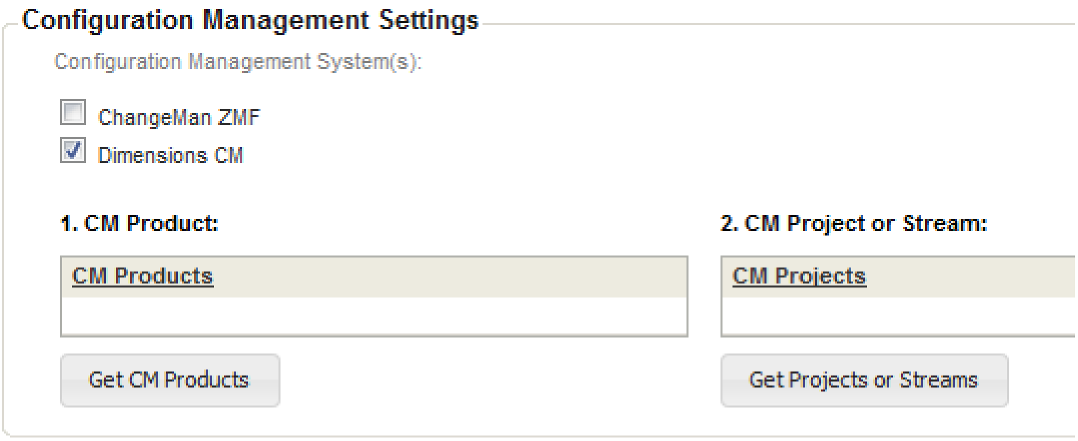
'Submit' by **Miha Project Manager** -08/30/2012 07:44:25 AM ⊞

'Update' by **Josh Development Manager** -08/30/2012 07:47:15 AM ⊟

| Changed Value | Prior Value | New Value |
|---|---|---|
| Configuration Management System(s) | (None) | Dimensions CM |
| CM Product | | QLARIUS |
| CM Project | | QLARIUS:JAVA_BRANCHA_STR |
| Last Modified Date | 08/30/2012 07:44:25 AM | 08/30/2012 07:47:15 AM |

'Update' by **Josh Development Manager** -08/30/2012 07:47:28 AM ⊞

- If you have selected Dimensions CM as the Configuration Management tool, the Dimensions CM products and projects or streams to associate with the project. Click **Get CM Products** and **Get CM Projects/Streams** to list available products and projects / streams.



Configuration Management Settings

Configuration Management System(s):

☐ ChangeMan ZMF

☑ Dimensions CM

1. CM Product:

CM Products

2. CM Project or Stream:

CM Projects

## Displaying and Creating Related Change Requests

On the **Change Requests** tab, you can list and display details on the change requests associated with the project, as well as create new change requests.

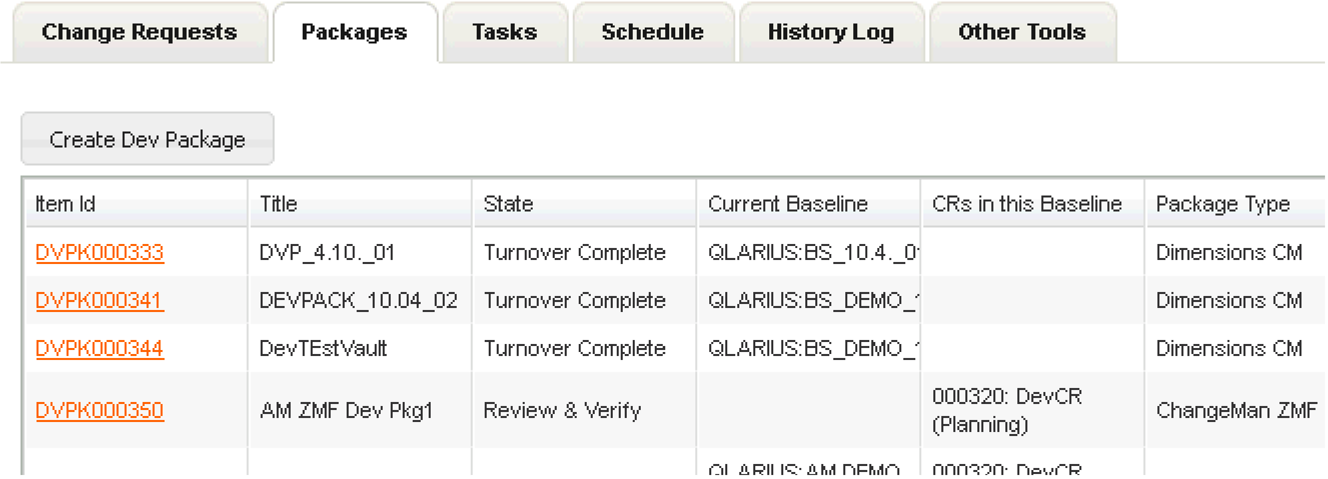
To create new change requests, click the **Create Dev CR** button.

You can also create dev change requests from the **Action** button, but in this case you will need to select the ALM project to which it belongs when you create it.

See "Creating New Change Requests" on page 37 for details on creating change requests.

## Creating and Relating Development Packages

Under **Dev Packages**, you can list and display details on the development packages associated with the project.



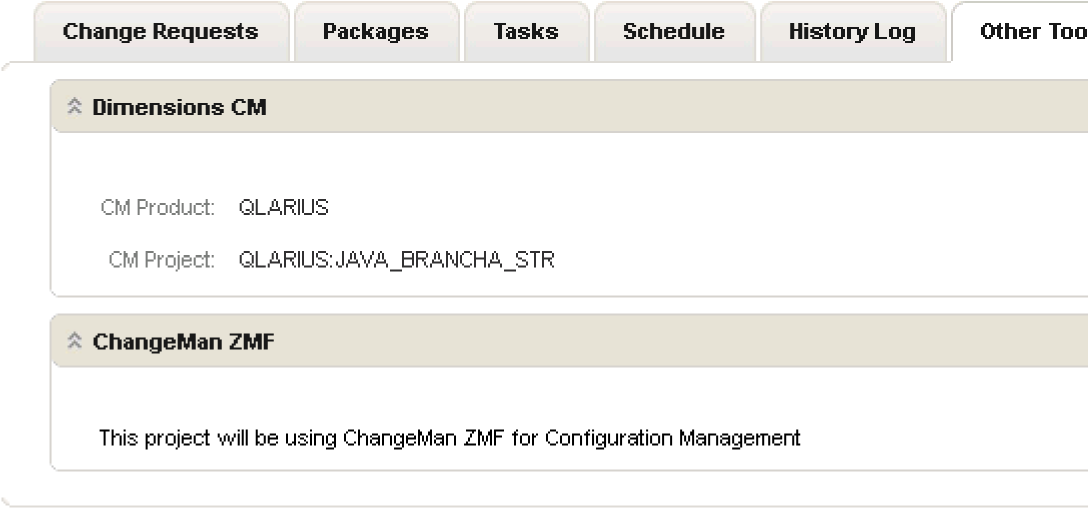
| Item Id | Title | State | Current Baseline | CRs in this Baseline | Package Type |
|---|---|---|---|---|---|
| DVPK000333 | DVP_4.10._01 | Turnover Complete | QLARIUS:BS_10.4._0 | | Dimensions CM |
| DVPK000341 | DEVPACK_10.04_02 | Turnover Complete | QLARIUS:BS_DEMO_ | | Dimensions CM |
| DVPK000344 | DevTEstVault | Turnover Complete | QLARIUS:BS_DEMO_ | | Dimensions CM |
| DVPK000350 | AM ZMF Dev Pkg1 | Review & Verify | | 000320: DevCR (Planning) | ChangeMan ZMF |
| | | | QLARIUS:AM DEMO | 000320: DevCR | |

Click the item id of any dev package to display more information about it. Click the **Create Dev Package** button to create a new package.

See "Creating New Packages" on page 45 for details.

## Configuration Management Tools

You can view details of the Configuration Management tools being used for this project by clicking the **Other Tools** tab. You can see whether Dimensions CM or ChangeMan ZMF are being used to track change requests, and for Dimensions CM which product and project/ stream is being used.



Dimensions CM

CM Product: QLARIUS

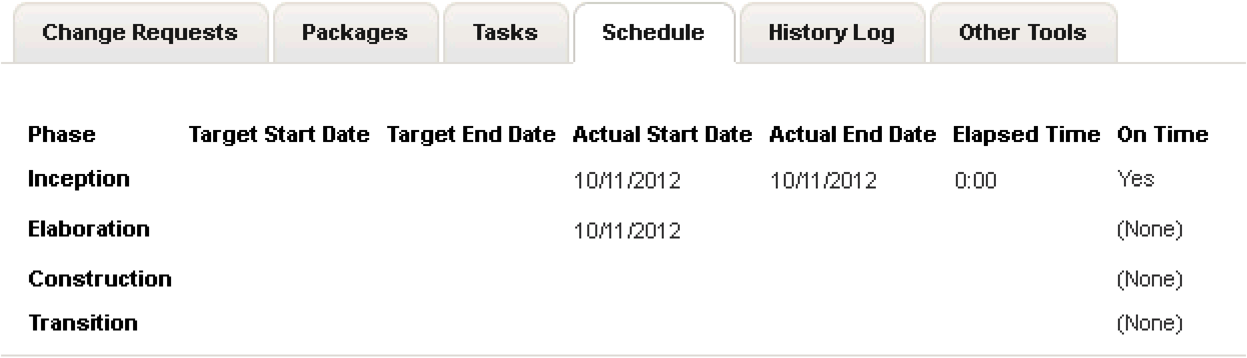
CM Project: QLARIUS:JAVA_BRANCHA_STR

ChangeMan ZMF

This project will be using ChangeMan ZMF for Configuration Management

## Project Schedule

You can view the scheduled and actual dates for the project on the Schedule Tab.

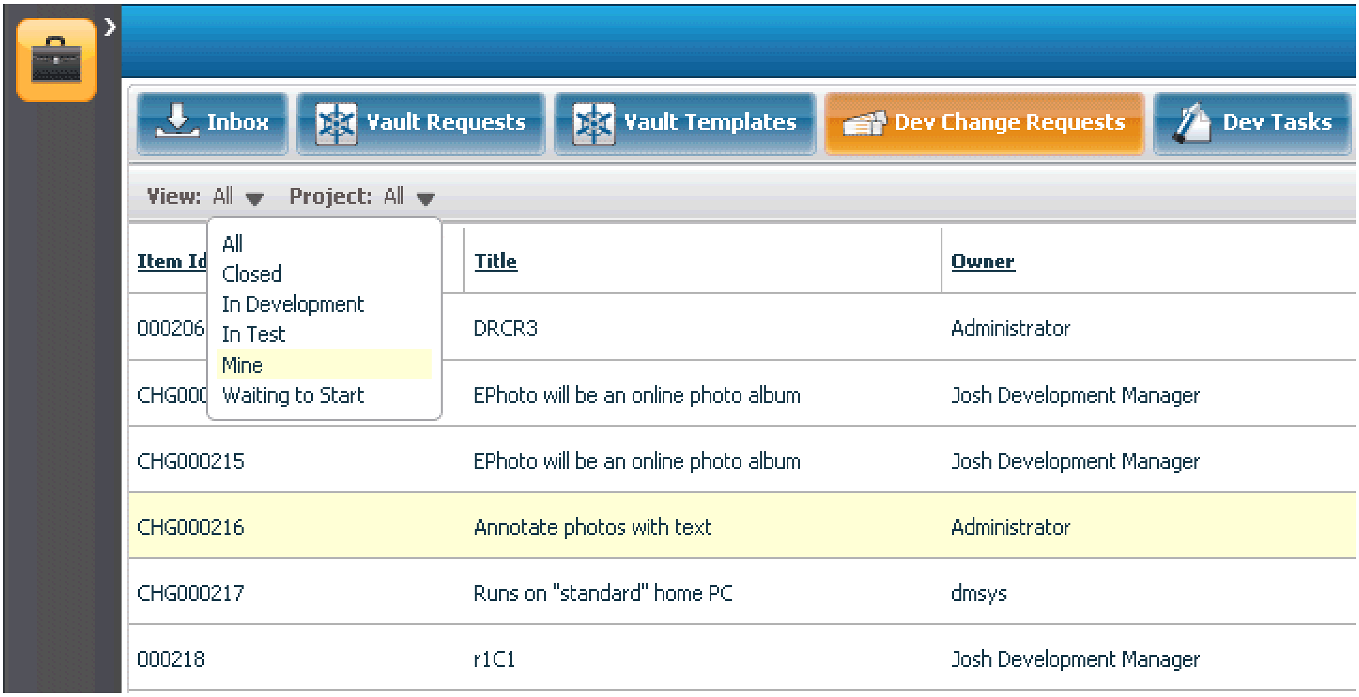


| Phase | Target Start Date | Target End Date | Actual Start Date | Actual End Date | Elapsed Time | On Time |
|---|---|---|---|---|---|---|
| Inception | | | 10/11/2012 | 10/11/2012 | 0:00 | Yes |
| Elaboration | | | 10/11/2012 | | | (None) |
| Construction | | | | | | (None) |
| Transition | | | | | | (None) |

# Creating and Working on Change Requests

From the Dev Change Request tab, you can list dev change requests and select a specific change request to edit it.create.

You can filter the list according to various criteria.



View: All    Project: All

| Item Id | Title | Owner |
|---|---|---|
| 000206 | DRCR3 | Administrator |
| CHG000 | EPhoto will be an online photo album | Josh Development Manager |
| CHG000215 | EPhoto will be an online photo album | Josh Development Manager |
| CHG000216 | Annotate photos with text | Administrator |
| CHG000217 | Runs on "standard" home PC | dmsys |
| 000218 | r1C1 | Josh Development Manager |

By clicking the arrow for the **View** option, you can select:

- *Mine* to view change requests owned by you.
- Select a state to view only dev change requests in a specific lifecycle state.

By clicking the arrow for the Project option, you can select dev change requests that belong to a specific project.

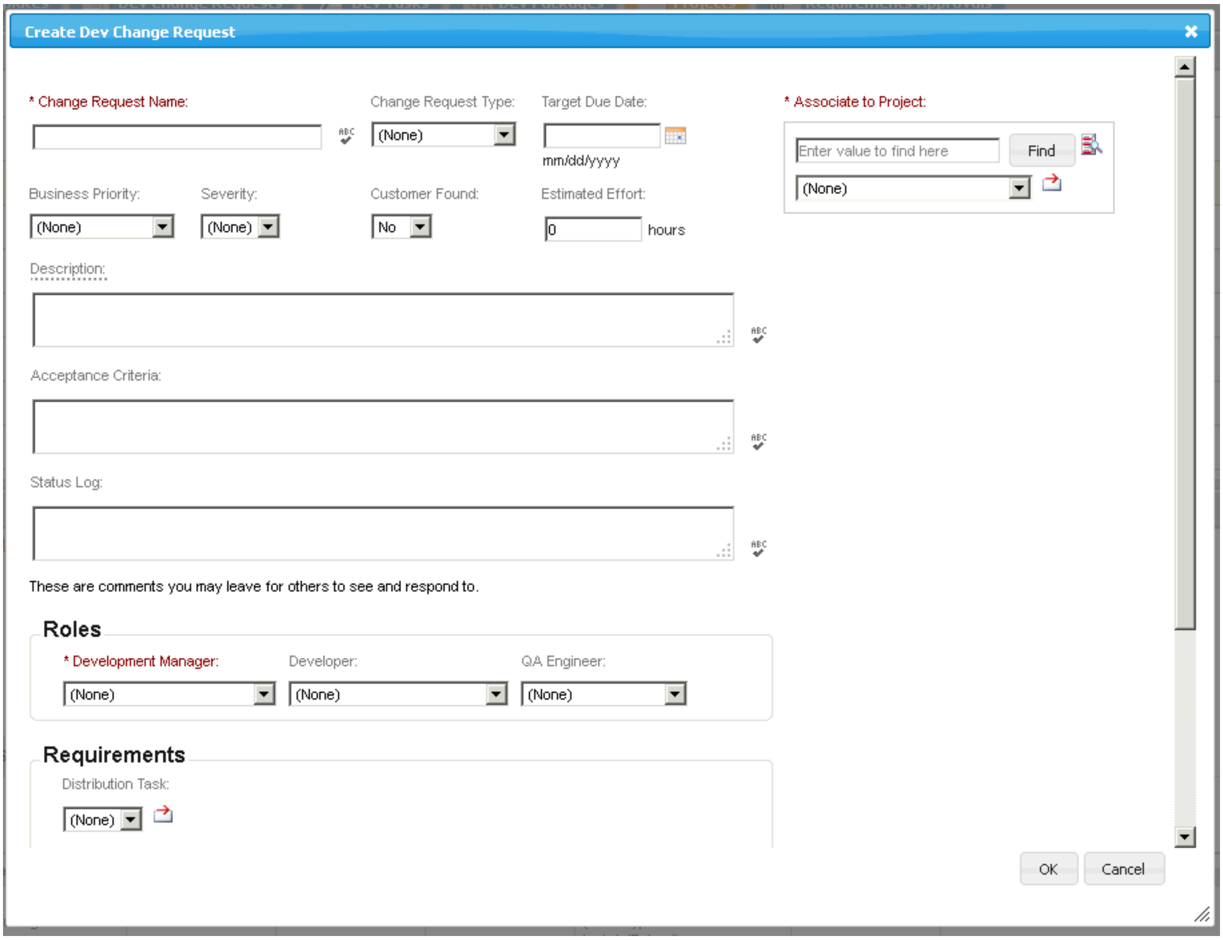
To open a change request, click on the corresponding row

## Creating New Change Requests

You can submit a new change request either:

- By clicking the **Actions** button and selecting New Change Request.
- By selecting an ALM project from the Projects tab, selecting the **Change Requests tab**, and clicking the **Create Dev CR** button. In this case you will not see the **Associate to Project** fields as the change request will automatically be relate to the selected project.

When you submit a new change request, a form Create Dev Change Request form appears.



Create Dev Change Request

* Change Request Name:

Change Request Type: (None)

Target Due Date: mm/dd/yyyy

* Associate to Project: Enter value to find here Find (None)

Business Priority: (None)

Severity: (None)

Customer Found: No

Estimated Effort: 0 hours

Description:

Acceptance Criteria:

Status Log:

These are comments you may leave for others to see and respond to.

Roles

* Development Manager: (None)

Developer: (None)

QA Engineer: (None)

Requirements

Distribution Task: (None)

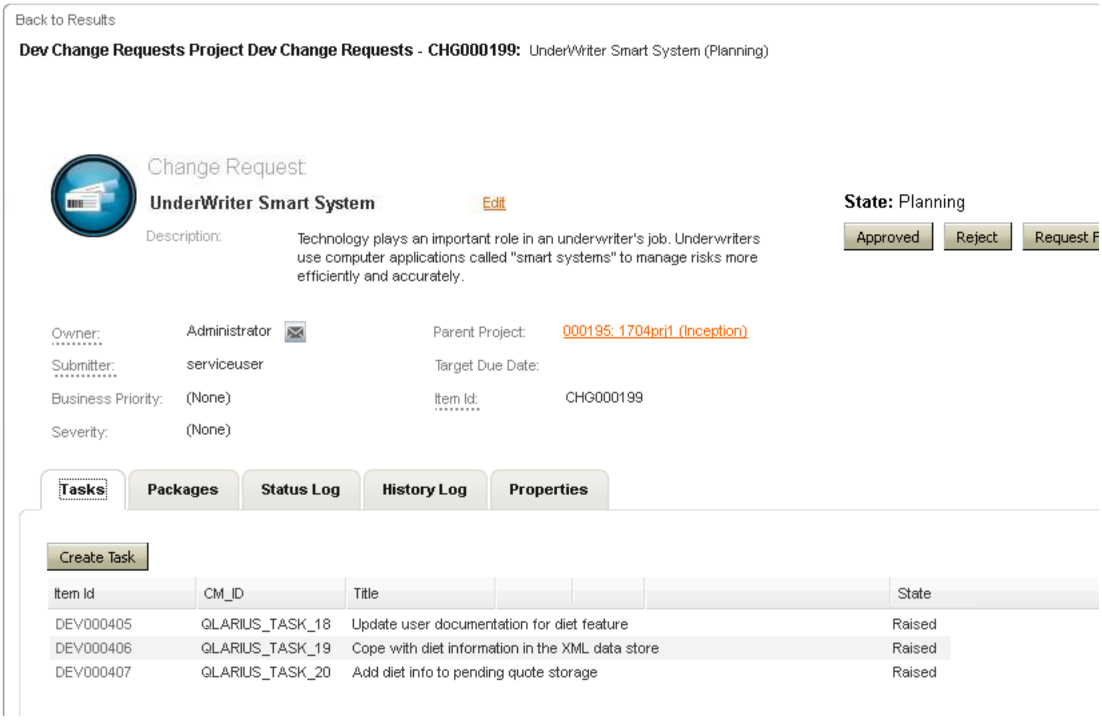
If you clicked the **Create Dev CR** button from a selected project, you will not see the **Associate to Project** fields as the change request will automatically be related to the selected project, otherwise you will need to select a project

You set a number of core properties for the request. You can set most of these later on; only the fields with red labels are required. You can set options such as:

- From the **Item Type** list, the type of request such as **Defect**.
- The title and description.
- Business priority, severity, and acceptance criteria for resolving the request.
- Specific users in the various user roles, such as the **Development Manager** or **Build Engineer**.

## Working on Change Requests

Open a change request to update and work on it. A change request with some associated tasks might look something like this.



Back to Results

**Dev Change Requests Project Dev Change Requests - CHG000199:** UnderWriter Smart System (Planning)

Change Request

UnderWriter Smart System

Description: Technology plays an important role in an underwriter's job. Underwriters use computer applications called "smart systems" to manage risks more efficiently and accurately.

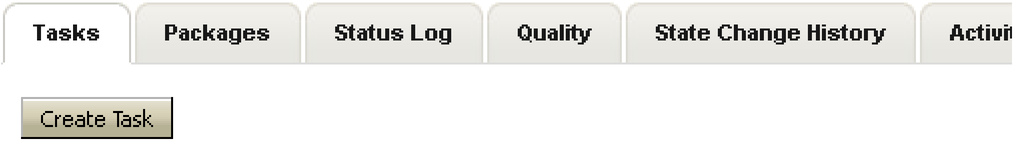
State: Planning

| | | | |
|---|---|---|---|
| Owner: | Administrator | Parent Project: | 000195: 1704prj1 (Inception) |
| Submitter: | serviceuser | Target Due Date: | |
| Business Priority: | (None) | Item Id: | CHG000199 |
| Severity: | (None) | | |

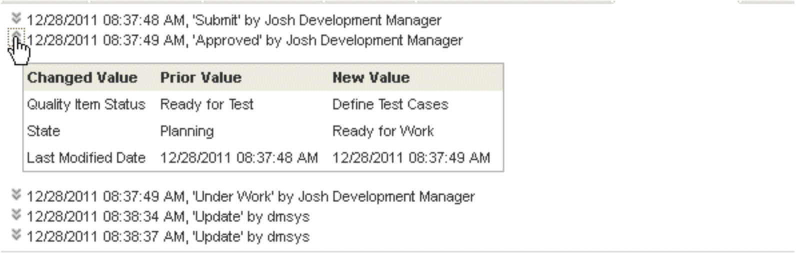
| Item Id | CM_ID | Title | State |
|---|---|---|---|
| DEV000405 | QLARIUS_TASK_18 | Update user documentation for diet feature | Raised |
| DEV000406 | QLARIUS_TASK_19 | Cope with diet information in the XML data store | Raised |
| DEV000407 | QLARIUS_TASK_20 | Add diet info to pending quote storage | Raised |

From a change request, you can:

- Review information about the change request, including its change history, Quality Center relationships, associated tasks, associated change requests, and associated development packages by selecting the relevant tabs.
- Update the title, description, assignment, and other properties by clicking the **Edit** link.
- Transition the request to a different state, for example click the **Under Work** button to transition it to the Under Work state.
- Click the **Create Task** button on the **Tasks** tab to create a task that will then be related to this request, and that you can then assign to the specific resource who will do the work.



- View the Activity Log on the **History Log** tab to see the actions that have been performed on the change request. Click the arrows at the left of a row to expand the details.



12/28/2011 08:37:48 AM, 'Submit' by Josh Development Manager
12/28/2011 08:37:49 AM, 'Approved' by Josh Development Manager

| Changed Value | Prior Value | New Value |
|---|---|---|
| Quality Item Status | Ready for Test | Define Test Cases |
| State | Planning | Ready for Work |
| Last Modified Date | 12/28/2011 08:37:48 AM | 12/28/2011 08:37:49 AM |

12/28/2011 08:37:49 AM, 'Under Work' by Josh Development Manager
12/28/2011 08:38:34 AM, 'Update' by dmsys
12/28/2011 08:38:37 AM, 'Update' by dmsys

# Creating and Working on Tasks

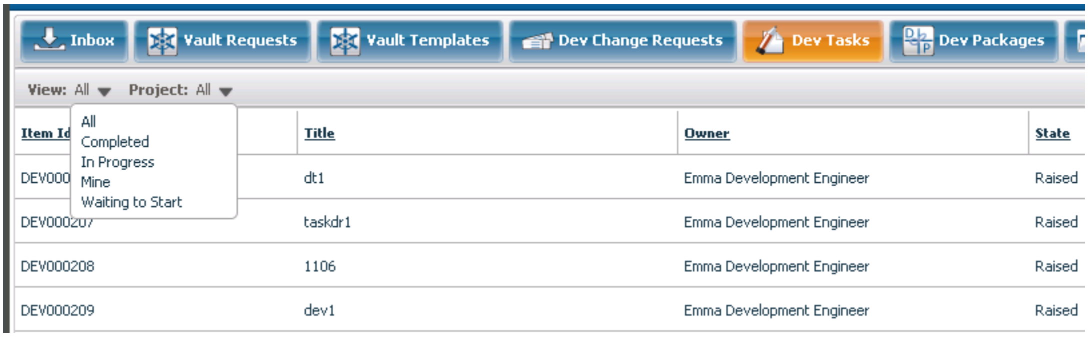
From the Dev Tasks tab, you can list, update, and report on development tasks. You can relate tasks to projects in the **Projects** tab, and relate tasks to Dimensions CM tasks.

By clicking the arrow for the **View** option, you can select:

- *Mine* to view tasks owned by you.
- Select a state to view only tasks in a specific lifecycle state.

By clicking the arrow for the **Project** option, you can select dev change requests that belong to a specific project.

To open a change request, click on the corresponding row



| Item Id | Title | Owner | State |
|---|---|---|---|
| DEV000 | dt1 | Emma Development Engineer | Raised |
| DEV000207 | taskdr1 | Emma Development Engineer | Raised |
| DEV000208 | 1106 | Emma Development Engineer | Raised |
| DEV000209 | dev1 | Emma Development Engineer | Raised |

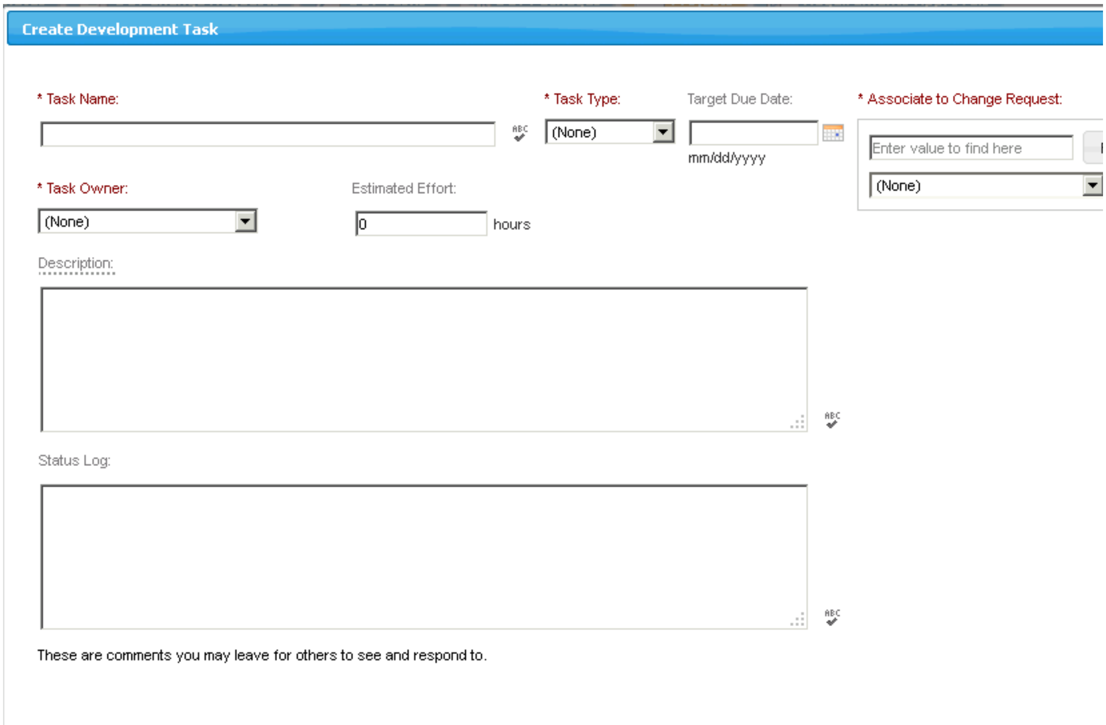
## Creating New Tasks

You can submit a new dev task either:

- By clicking the **Actions** button and selecting *New Task*.
- By selecting a Change Request from the Change Requests tab, selecting the **Tasks tab**, and clicking the **Create Task** button. In this case you will not see the **Associate to Project** fields as the change request will automatically be related to the selected project.

When you create new development tasks, the tasks are synchronized either to new requests in Dimensions CM of type *Task*, or new or existing change packages in ChangeMan ZMF. Development engineers will relate these tasks to the code changes they make in whichever Configuration management tool they are using.

When you submit a new task, a form like the following appears.



Create Development Task

* Task Name:

* Task Type: (None)

Target Due Date: mm/dd/yyyy

* Associate to Change Request: Enter value to find here (None)

* Task Owner: (None)

Estimated Effort: 0 hours

Description:

Status Log:

These are comments you may leave for others to see and respond to.

When you create a new task, you set core properties for the task, including:

- From the **Item Type** field, the type of task, such as **Design** or **Development**.
- The title and description.
- The owner and estimated effort.

If the task type is Development, you will need to select a **Configuration Management System**. The selection available here depends on what was selected for **Configuration Management Settings** for the owning project, and by default, can be Dimensions CM or ChangeMan ZMF.

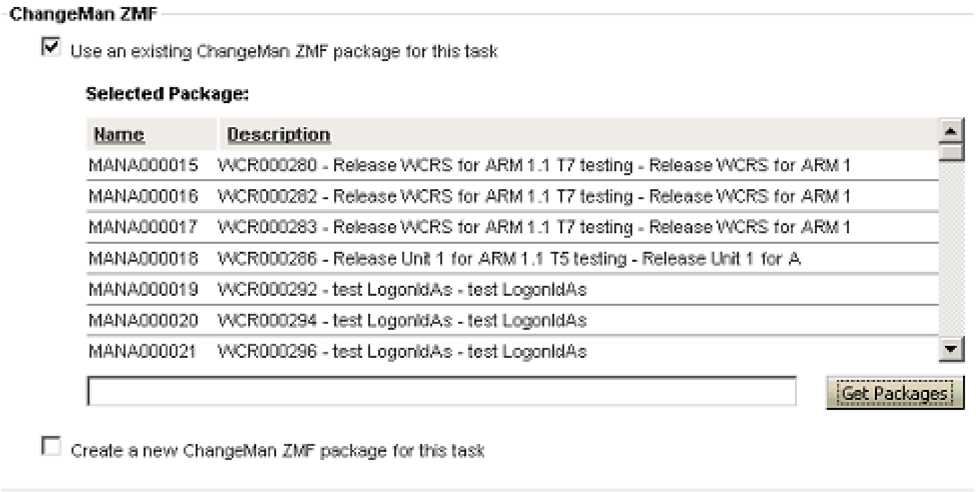
### Creating a Dimensions CM Task

If you select *Dimensions CM* for the **Configuration Management System**, a new Dimensions CM task request will be created in the associated Dimensions CM project or stream when you save the task.

### Creating a ChangeMan ZMF Task

When you have selected *ChangeMan ZMF* for the **Configuration Management System**, you will have the option of either selecting an existing ZMF change package or creating a new one. If you are selecting an existing one, click the **Get Packages** button, and select one from the list. Note that there is by default a one-to-one relationship between

ChangeMan ZMF packages and Dev Tasks, so the list will only show packages that are not associated with Development Control.



ChangeMan ZMF

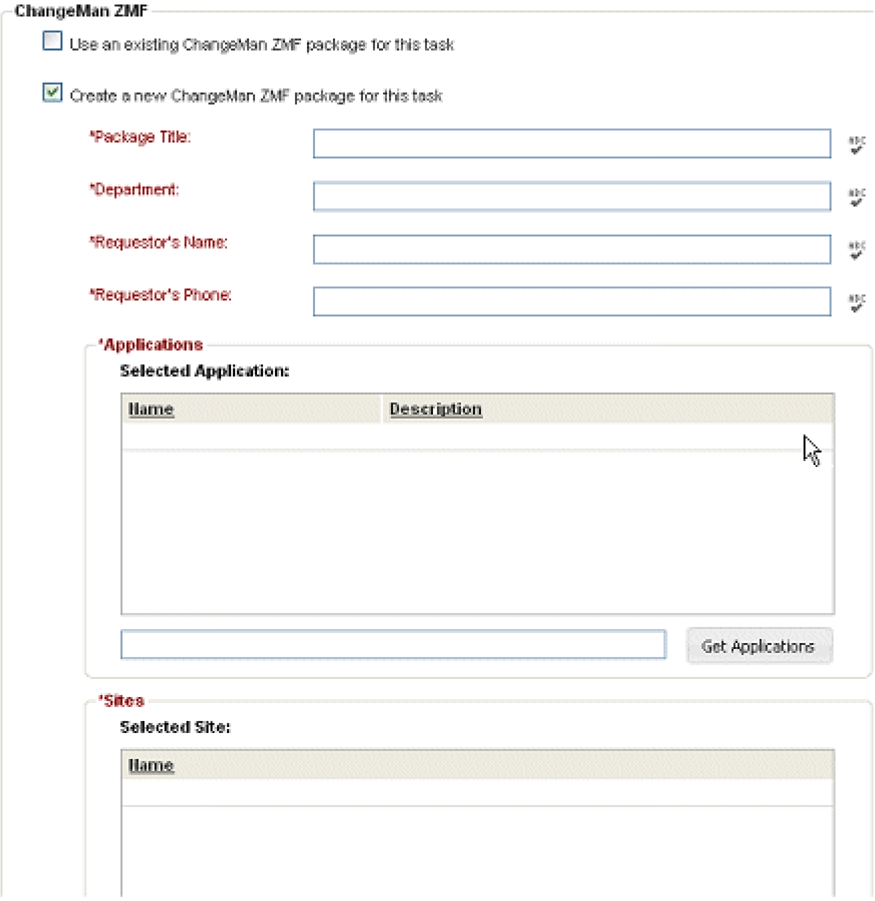
☑ Use an existing ChangeMan ZMF package for this task

Selected Package:

| Name | Description |
|---|---|
| MANA000015 | WCR000280 - Release WCRS for ARM 1.1 T7 testing - Release WCRS for ARM 1 |
| MANA000016 | WCR000282 - Release WCRS for ARM 1.1 T7 testing - Release WCRS for ARM 1 |
| MANA000017 | WCR000283 - Release WCRS for ARM 1.1 T7 testing - Release WCRS for ARM 1 |
| MANA000018 | WCR000286 - Release Unit 1 for ARM 1.1 T5 testing - Release Unit 1 for A |
| MANA000019 | WCR000292 - test LogonIdAs - test LogonIdAs |
| MANA000020 | WCR000294 - test LogonIdAs - test LogonIdAs |
| MANA000021 | WCR000296 - test LogonIdAs - test LogonIdAs |

□ Create a new ChangeMan ZMF package for this task

If you select **Create a new ChangeMan ZMF package for this task**, a new one will be created when you save the task. You will need to enter various details for the ZMF change package.



ChangeMan ZMF

□ Use an existing ChangeMan ZMF package for this task

☑ Create a new ChangeMan ZMF package for this task

*Package Title:

*Department:

*Requestor's Name:

*Requestor's Phone:

*Applications

Selected Application:

| Name | Description |
|---|---|

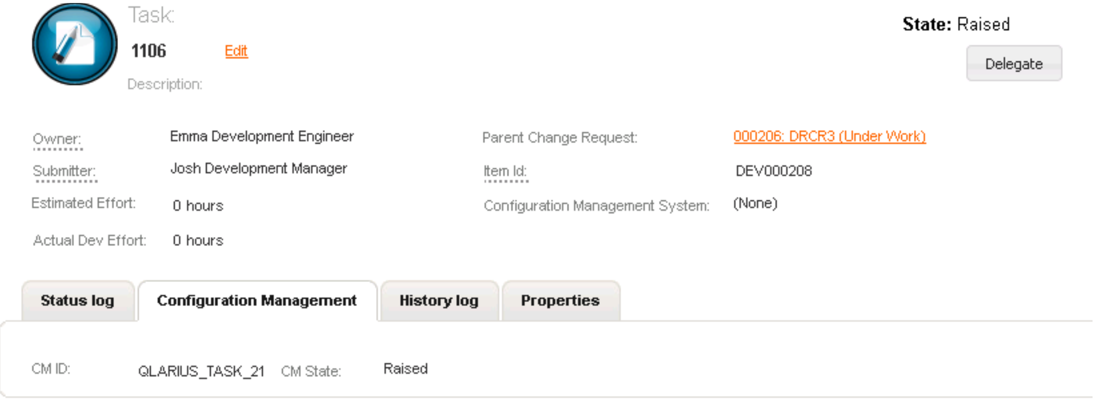
*Sites

Selected Site:

| Name |
|---|

## Working on Tasks

Open a task to update and work on it. The example below shows a dev task that is related to a ChangeMan ZMF package.



Task: 1106

Description:

State: Raised

| | | | |
|---|---|---|---|
| Owner: | Emma Development Engineer | Parent Change Request: | 000206: DRCR3 (Under Work) |
| Submitter: | Josh Development Manager | Item Id: | DEV000208 |
| Estimated Effort: | 0 hours | Configuration Management System: | (None) |
| Actual Dev Effort: | 0 hours | | |

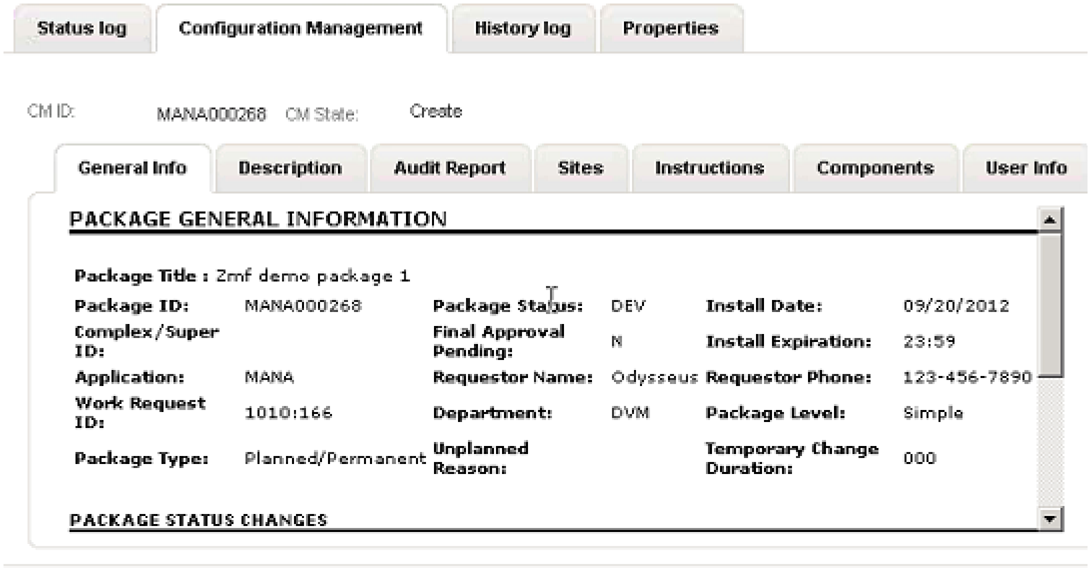
CM ID: QLARIUS_TASK_21 CM State: Raised

From a task, you can:

- Review information about the task, including its change history, Dimensions CM relationships, Quality Center relationships, associated change requests, and attachments.
- Transition a task from one state to the next. For example, to start working on a new task, click the **Begin Work** button.
- Click the **Edit** link to modify properties such as the owner of the task, estimated work, and attachments.

On the **Configuration Management** tab, you will see the details of the Dimensions CM request or the ChangeMan ZMF package.

For a ZMF package you will see a number of tabs relating to the selected package. The Development Control generated ID relating to the package in ChangeMan ZMF is shown in the **Work Request ID** field.



CM ID: MANA000268 CM State: Create

PACKAGE GENERAL INFORMATION

Package Title : Zmf demo package 1

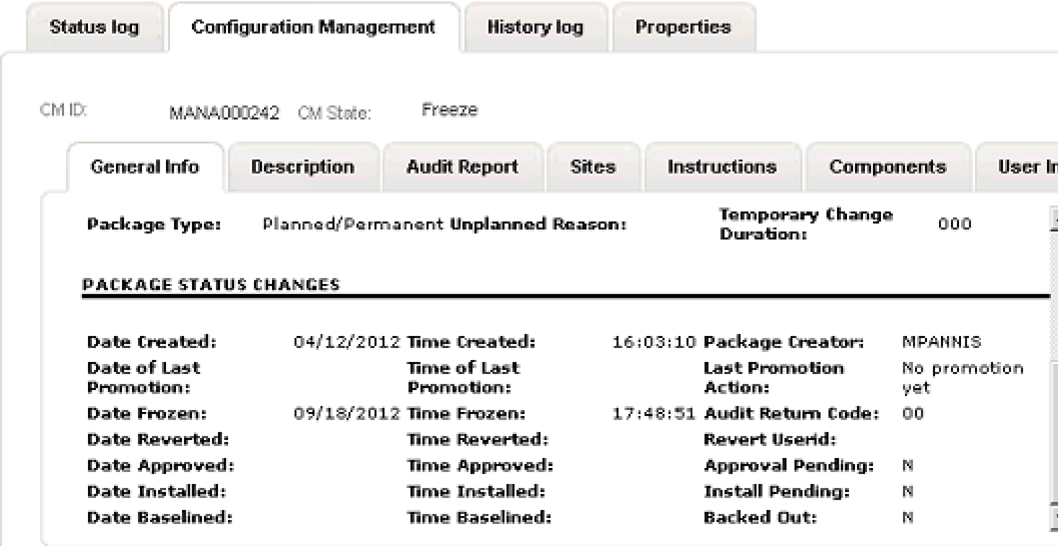
| | | | | | |
|---|---|---|---|---|---|
| Package ID: | MANA000268 | Package Status: | DEV | Install Date: | 09/20/2012 |
| Complex/Super ID: | | Final Approval Pending: | N | Install Expiration: | 23:59 |
| Application: | MANA | Requestor Name: | Odysseus | Requestor Phone: | 123-456-7890 |
| Work Request ID: | 1010:166 | Department: | DVM | Package Level: | Simple |
| Package Type: | Planned/Permanent | Unplanned Reason: | | Temporary Change Duration: | 000 |

PACKAGE STATUS CHANGES

Other fields of note can be seen farther down under Package Status Changes:

The **Audit Return Code**, if not blank, shows that the package has been audited

The **Date Frozen** and **Time Frozen** fields, if present, show that the package has been frozen



CM ID: MANA000242 CM State: Freeze

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Package Type: | Planned/Permanent | Unplanned Reason: | | Temporary Change Duration: | 000 | | |

PACKAGE STATUS CHANGES

| | | | | | |
|---|---|---|---|---|---|
| Date Created: | 04/12/2012 | Time Created: | 16:03:10 | Package Creator: | MPANNIS |
| Date of Last Promotion: | | Time of Last Promotion: | | Last Promotion Action: | No promotion yet |
| Date Frozen: | 09/18/2012 | Time Frozen: | 17:48:51 | Audit Return Code: | 00 |
| Date Reverted: | | Time Reverted: | | Revert Userid: | |
| Date Approved: | | Time Approved: | | Approval Pending: | N |
| Date Installed: | | Time Installed: | | Install Pending: | N |
| Date Baselined: | | Time Baselined: | | Backed Out: | N |

### *Lifecycle of Dimensions CM tasks*

Once created and delegated to a development engineer, the code is worked on in Dimensions CM and the task is actioned from **Under Work** to **Peer Review** and then to **Complete**. These transitions take place in Dimensions CM.

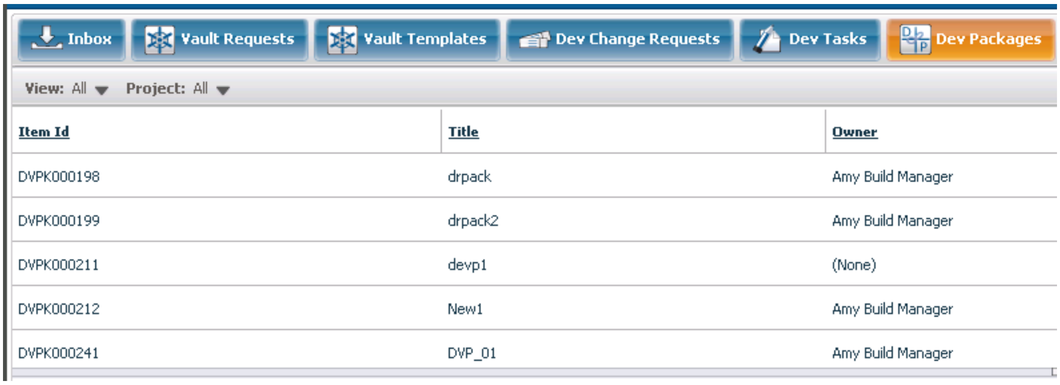
### *Lifecycle of ChangeMan ZMF tasks*

Once created and delegated to a development engineer, the code is worked on in ChangeMan ZMF. The change package is then Moved to **Frozen**, but it must be **audited** first. These transitions can be performed from either ChangeMan ZMF or Development Control. When the package is **Frozen** it is automatically transtioned to **Complete** in Development Control.

## Creating and Working with Development Packages

From the **Dev Packages** tab, you can list, update, and report on development packages. Development packages collect together all work associated with a release. You can gather a group of change requests related to a project in a dev package. You can also associate them with a baseline in Dimensions CM. This allows you to deploy the specific files in

Dimensions CM that are related to the change requests, and run any build processes required to test the changes.



View: All Project: All

| Item Id | Title | Owner |
|---|---|---|
| DVPK000198 | drpack | Amy Build Manager |
| DVPK000199 | drpack2 | Amy Build Manager |
| DVPK000211 | devp1 | (None) |
| DVPK000212 | New1 | Amy Build Manager |
| DVPK000241 | DVP_01 | Amy Build Manager |

By clicking the arrow for the **View** option, you can select:

- *Mine* to view dev packages owned by you.
- Select a state to view only dev packages in a specific lifecycle state.

By clicking the arrow for the **Project** option, you can select dev change requests that belong to a specific project.

To open a change request, click on the corresponding row

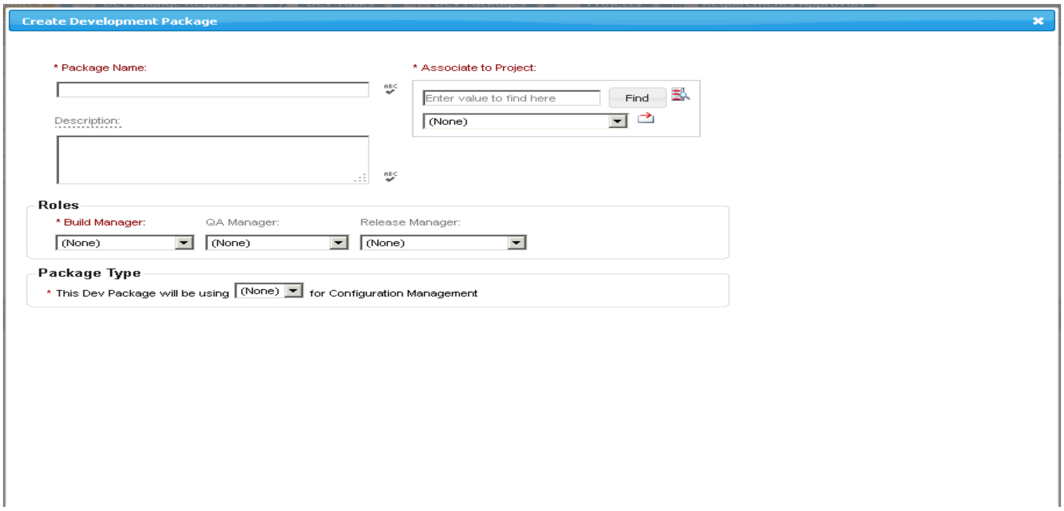
You can also find and track specific packages by running and creating reports from the **Reports** view. For more on reporting in SBM, see the SBM online help.

## Creating New Packages

You can submit a new change request either:

- By clicking the **Actions** button and selecting *New Task*.
- By selecting an ALM project from the Projects tab, selecting the **Packages tab**, and clicking the **Create Dev Package** button. In this case you will not see the **Associate to Project** fields as the dev package will automatically be related to the selected project.

When you create a new dev package, a page like the following appears.



* Package Name:

Description:

* Associate to Project:
Enter value to find here
(None)

Roles
* Build Manager: (None)
QA Manager: (None)
Release Manager: (None)

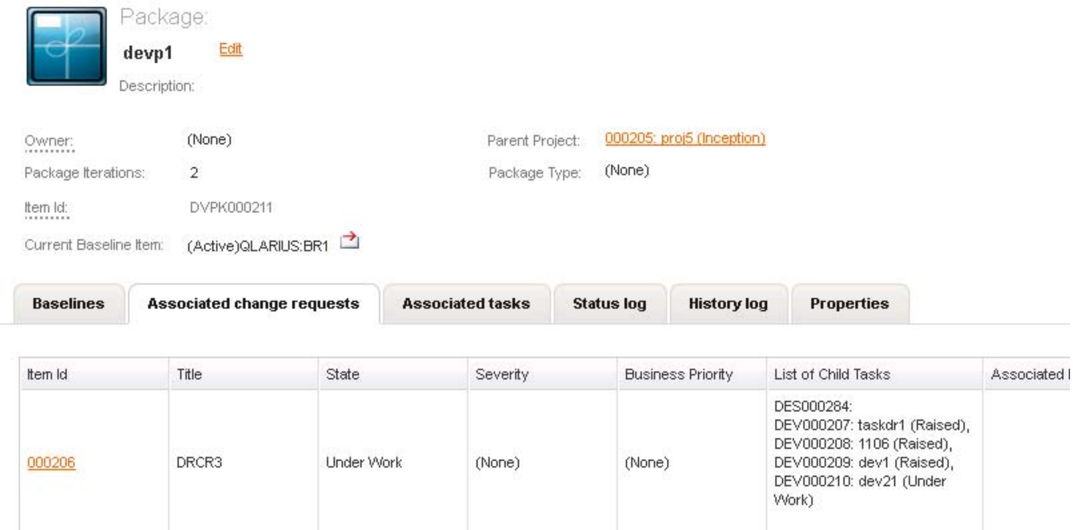
Package Type
* This Dev Package will be using (None) for Configuration Management

When you create a new package, you set core properties for the package, including:

- The title and description.
- The Release Engineer, Build Engineer, and QA Manager to assign to the package.
- Associated ALM project. This then supplies the name of the Dimensions product and project / stream.

## Working on Packages

Open a package to update and work on it. A package might look something like this.



Package:
devp1 Edit
Description:

| | | | |
|---|---|---|---|
| Owner: | (None) | Parent Project: | 000205: proj5 (Inception) |
| Package Iterations: | 2 | Package Type: | (None) |
| Item Id: | DVPK000211 | | |
| Current Baseline Item: | (Active)QLARIUS:BR1 | | |

| Item Id | Title | State | Severity | Business Priority | List of Child Tasks | Associated I |
|---|---|---|---|---|---|---|
| 000206 | DRCR3 | Under Work | (None) | (None) | DES000284:<br>DEV000207: taskdr1 (Raised),<br>DEV000208: 1106 (Raised),<br>DEV000209: dev1 (Raised),<br>DEV000210: dev21 (Under Work) | |

From a package, you can:

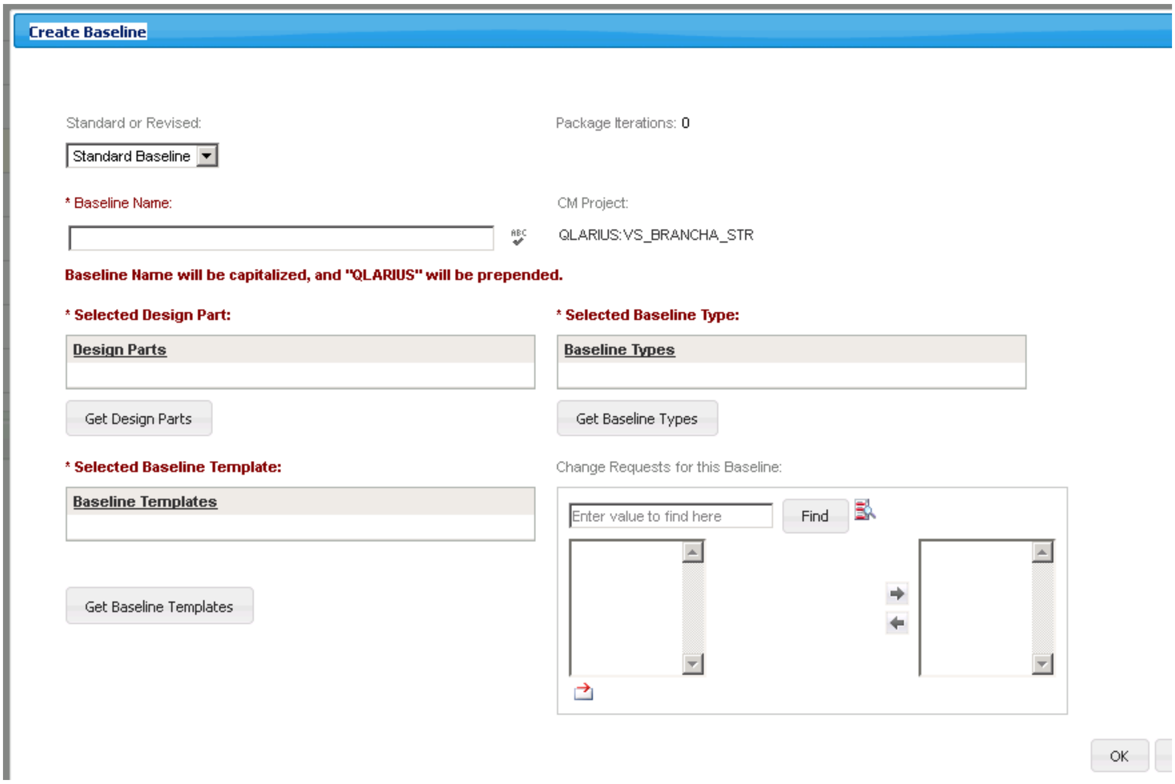
- Review information about the package, including its change history, associated Dimensions CM baselines, and associated ALM project.
- Transition the package to the next state in the workflow, for example if the package is in the **Ready To Build** state, you can click the **Build** button to transition it to the **Building** state. Note that you need the appropriate role, such as Build manager, to transition the package, otherwise the required button will not appear.
- Click the **Edit** button to change core properties for the package, including associated change requests, title and description, and the ALM project that is related to the package.

## Creating New Dimensions CM Baselines

You can create new Dimensions CM baselines from a selected dev package on the Dev Packages tab. When you create new baselines, you choose which change requests to associate with the new baseline. You can also create a new revised baseline.

**To create new Dimensions CM baselines:**

1 From a development package that is in the **Create Package** state, click the **Create Baseline** button.



Create Baseline

Standard or Revised: Standard Baseline

Package Iterations: 0

* Baseline Name:

CM Project: QLARIUS:VS_BRANCHA_STR

Baseline Name will be capitalized, and "QLARIUS" will be prepended.

* Selected Design Part: Design Parts

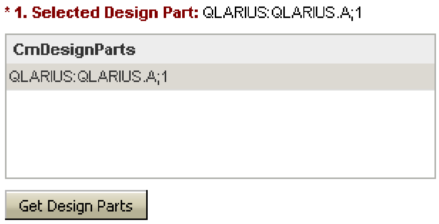
* Selected Baseline Type: Baseline Types

* Selected Baseline Template: Baseline Templates

Change Requests for this Baseline:

2 Enter the baseline name in the **Baseline Name** field. This is the name that will appear on the baseline in Dimensions CM. This must be defined in all capital letters.

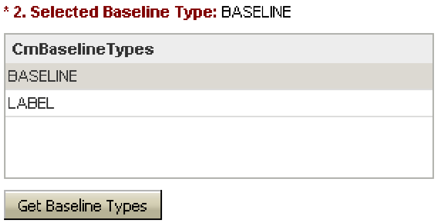
3 In the **Selected Design Part field**, click the **Get Design Parts** button to list all options. Select a design part from which to scope items selected for the baseline..



* 1. Selected Design Part: QLARIUS:QLARIUS.A;1

| CmDesignParts |
|---|
| QLARIUS:QLARIUS.A;1 |

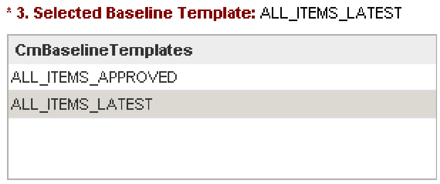
4 In the **Selected Baseline Type** field, click the **Get Baseline Types** button and select a baseline type.



* 2. Selected Baseline Type: BASELINE

| CmBaselineTypes |
|---|
| BASELINE |
| LABEL |

5 In the **Selected Baseline Template** field, click the **Get Baseline Templates** button and select a baseline template.

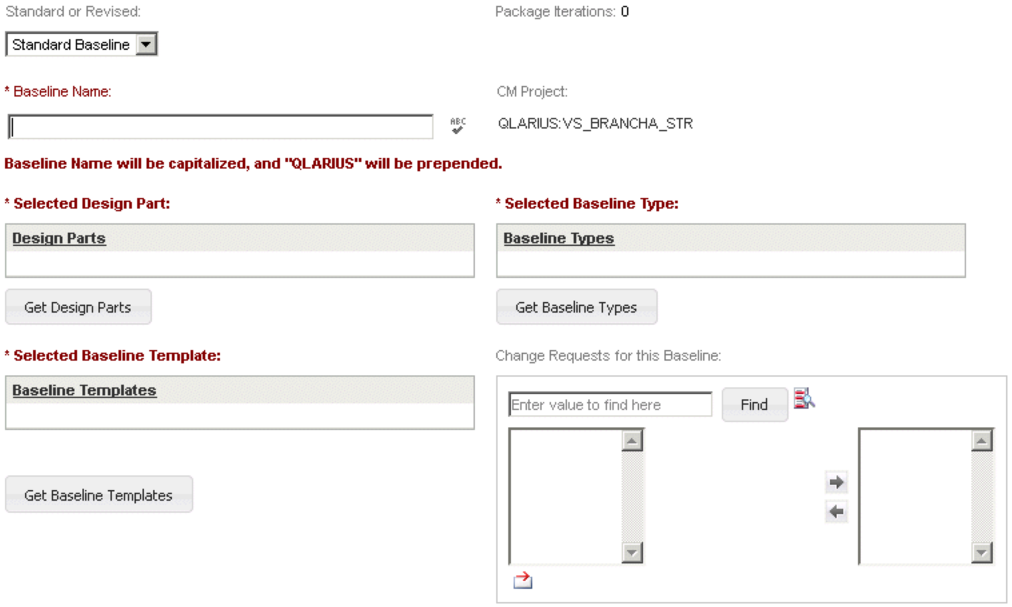


* 3. Selected Baseline Template: ALL_ITEMS_LATEST

CmBaselineTemplates

ALL_ITEMS_APPROVED

ALL_ITEMS_LATEST



Standard or Revised: Standard Baseline

Package Iterations: 0

* Baseline Name:

CM Project: QLARIUS:VS_BRANCHA_STR

Baseline Name will be capitalized, and "QLARIUS" will be prepended.

* Selected Design Part: Design Parts

* Selected Baseline Type: Baseline Types

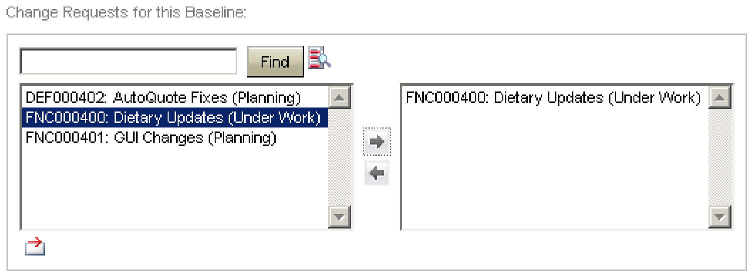
* Selected Baseline Template: Baseline Templates

Change Requests for this Baseline: Enter value to find here

6 Search for change requests that you want to associate with the baseline, by selecting them in the **Change Requests for this Baseline** field.

Tasks associated with these change requests are in turn related to corresponding tasks in Dimensions CM; and files that have been updated in Dimensions CM in response to those tasks will be included in the baseline.

To use this field, enter part of the name of the change requests and click the **Find** button to locate them. Then select the required change request and click the arrow to move them to the right-hand list



Change Requests for this Baseline:

DEF000402: AutoQuote Fixes (Planning)
FNC000400: Dietary Updates (Under Work)
FNC000401: GUI Changes (Planning)

FNC000400: Dietary Updates (Under Work)

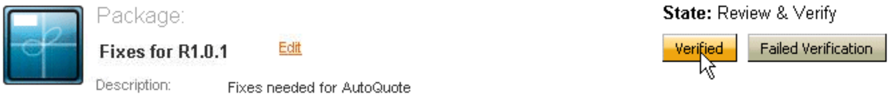
If that doesn't work, click the ➦ button. On the dialog box that appears, click the **Lookup** button to display a list of all available change requests, or enter a specific ID or title in the Item Id or Title field.

7 If you are revising an existing baseline, you must complete the options under Revised Baseline Settings:

- Under **Previous Baseline Project**, locate the project in ALM Projects that the package and baseline belong to.
- Then, locate the dev package to which the baseline is associated, under **Previous Baseline Dev Package**.
- Finally, locate the existing baseline in the **Previous Baseline** field.
- Based on the above searches, you can now update the change requests associated with this baseline. This will create a new revised baseline in Dimensions CM. The changes you make to associated change requests will in turn modify the tasks and associated versioned files related to the baselines in Dimensions CM.

8 Once you create the new baseline, it will be associated with the current package.

9 Once the baseline has been successfully created in Dimensions CM, the dev package will be transitioned to *Review and Verify*. At this point the Build Manager will verify the baseline and either select **Failed Verification**, meaning that another baseline will need to be created for the dev package, or select **Verified**, meaning that it is *Ready for Build*.


Package:

Fixes for R1.0.1

Description: Fixes needed for AutoQuote

State: Review & Verify

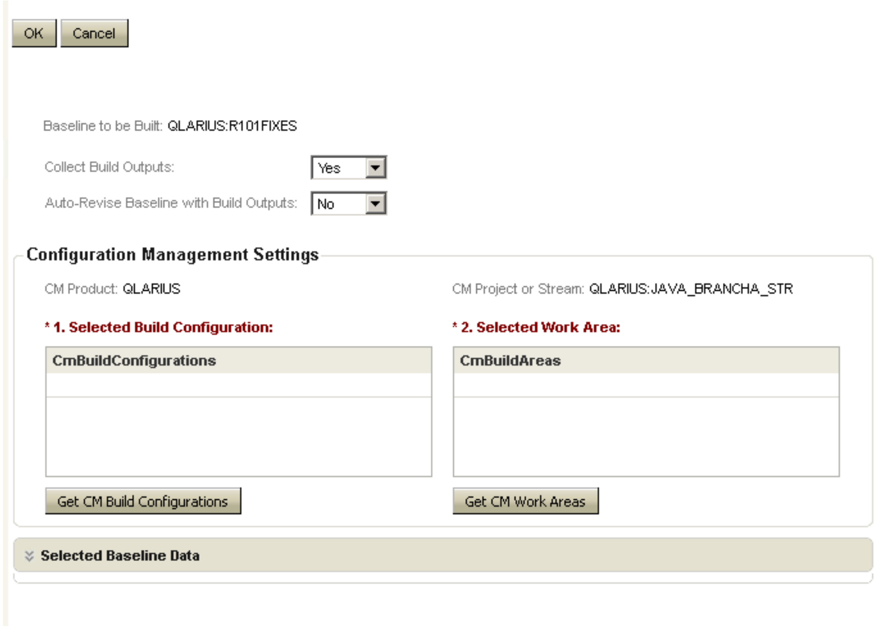
## Starting a Build

You can start a build from development packages that have been verified and are in the *Ready for Build* state.

**To start a build:**

1 From a package in the **Ready for Build** state, click the **Build** button. Normally you will need to be logged in as the Build Manager to do this.

A form like the following appears.



Baseline to be Built: QLARIUS:R101FIXES

Collect Build Outputs: Yes

Auto-Revise Baseline with Build Outputs: No

**Configuration Management Settings**

CM Product: QLARIUS

CM Project or Stream: QLARIUS:JAVA_BRANCHA_STR

* 1. Selected Build Configuration:

* 2. Selected Work Area:

2 The build will be based on the baseline in the **Current Baseline** field, and will include the change requests listed in the **CRs in this Baseline** list.

3 Click the **Get CM Build Configurations** button and select a build configuration

4 Click the **Get CM Work Areas** button and select a work area.

5 Click **OK**.

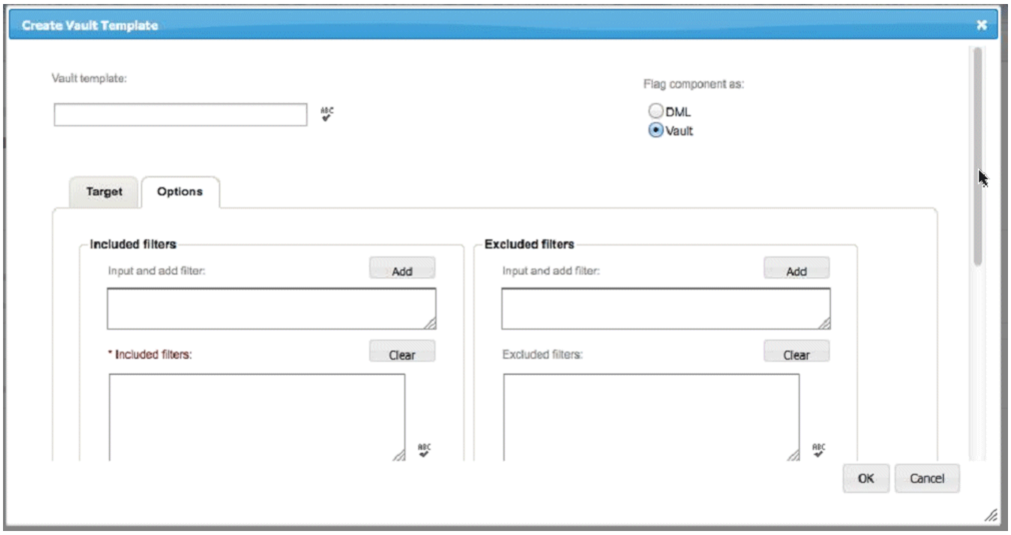
6 The package is transitioned to the **Building** state while the build task is executed by Dimensions CM. Once the build task is complete, the package is transitioned either to the **Failed Build** state (if the build failed), or the **Built** state (if the build succeeded). If the build failed, you must start again with a fresh package, once the issues have been addressed. If the build succeeded, you can transition the package to the next state, **Ready For Test**, by clicking the **Send to QA** button.

# Turning Over a Dev Package Using the Release Vault Integration

If you are using the Release Vault integration, you have the option of publishing the dev package for release via Serena Release Manager. For this you will need to have installed the Vault Request process app, and have created the necessary templates. For further details, see the *Release Manager Getting Started Guide* and the *Release Manager Installation and Configuration Guide*.

**To create a vault template:**

1 On the **Vault Requests** tab of the Serena ALM UI, click the **Actions** button, and select *New Vault Template*.



Create Vault Template

Vault template:

Flag component as:
○ DML
● Vault

2 Enter a name for the Vault Template

3 Select **Vault**.

4 On the Options tab, enter any required filters for the **Included filters.**

a Type a filter expression that can include wildcard expressions. For example `**/*.exe` to include .exe files.

b Click **Add** to add it to the Included Filters text box. Repeat this to add further filters.

5 Repeat the above for any required **Excluded filters**.

6 Click the **Get design part** button and select a design part.

7 Click the **Get vault stage** button and select a stage to which the baseline is to be deployed.

8 Enter a **Baseline stem** for the prefix to be included in the name of the baseline.

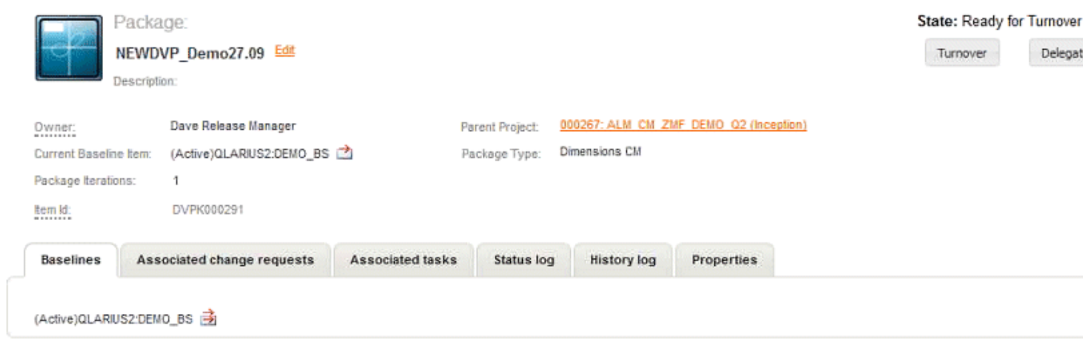
9 On the Target tab, Click the **Get DimCM Project** button and select a project from the list. This is the project to which the baseline will be deployed.

10 Click **Save**.

You can also create a template by creating a vault request and saving it as a template.

### To turn over a release package:

When a dev package has passed testing and reached the **Ready for Turnover** state the final transition in Development Manager is **Turnover**.



Package:
NEWDVP_Demo27.09
Description:

State: Ready for Turnover

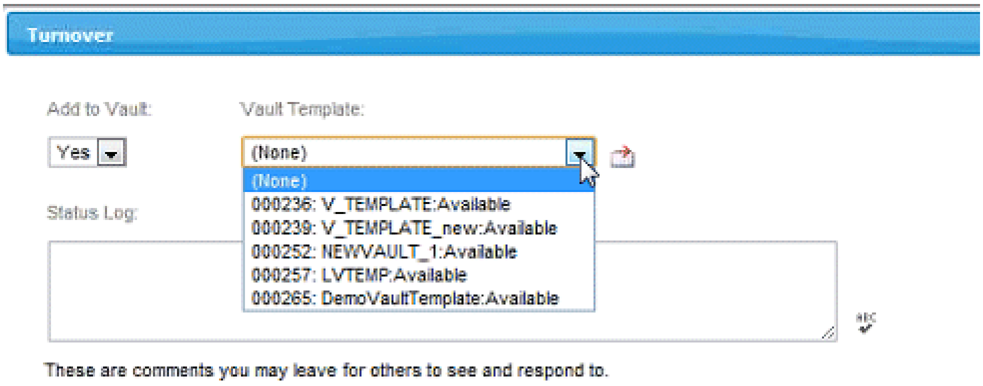
Owner: Dave Release Manager
Current Baseline Item: (Active)QLARIUS2:DEMO_BS
Package Iterations: 1
Item Id: DVPK000291
Parent Project: 000267: ALM_CM_ZMF_DEMO_Q2 (Inception)
Package Type: Dimensions CM

(Active)QLARIUS2:DEMO_BS

1 Click the **Turnover** button. This displays the Turnover page:

2 If you want to publish the dev package to the Release Vault, select **Yes** for **Add to Vault**

3 Select a template from the **Vault Template** list.



Turnover

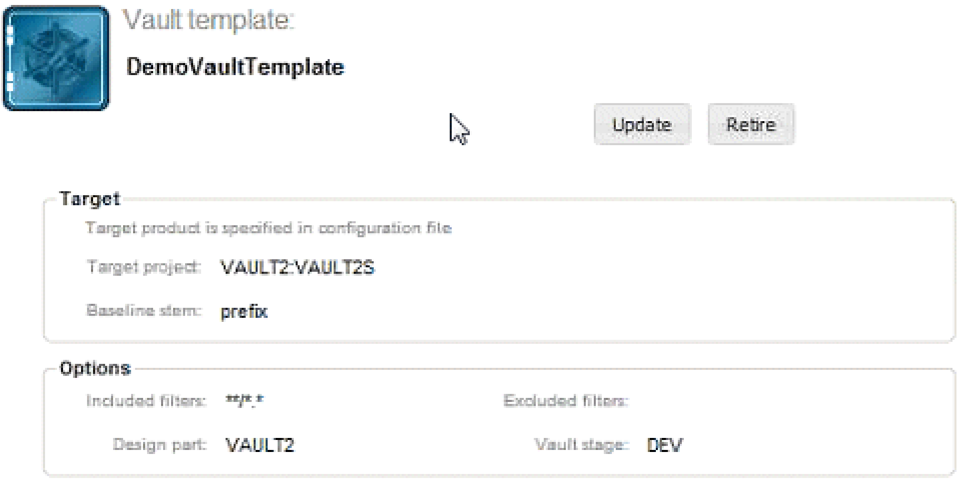
Add to Vault: Yes
Vault Template: (None)

(None)
000236: V_TEMPLATE:Available
000239: V_TEMPLATE_new:Available
000252: NEWVAULT_1:Available
000257: LVTEMP:Available
000265: DemoVaultTemplate:Available

Status Log:

These are comments you may leave for others to see and respond to.

The vault template determines which files are included in the baseline for turnover. You can click the button next to the template list to view or edit the template.



Vault template:

DemoVaultTemplate

Target

Target product is specified in configuration file

Target project: VAULT2:VAULT2S

Baseline stem: prefix

Options

Included filters: **/*.*

Excluded filters:

Design part: VAULT2

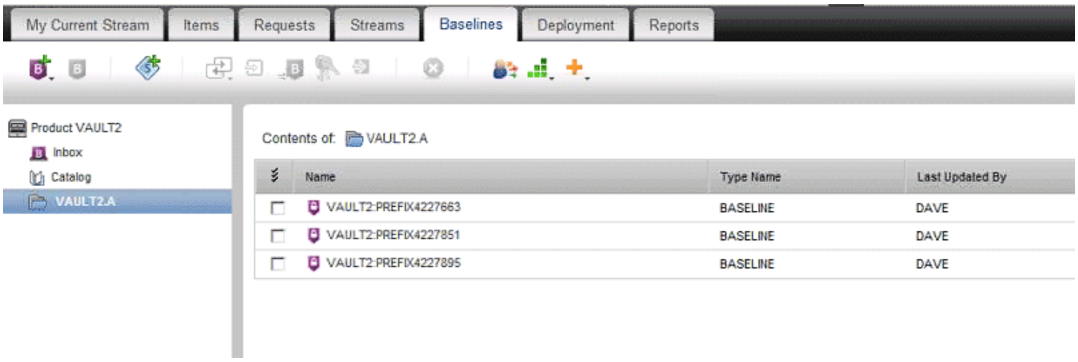
Vault stage: DEV

The **Target Product** is specified in the configuration file `dm_qlarius.properties`. For details, see *Designating the Details for Each Provider* in the *Development Manager Installation and Configuration Guide*.

The **Options** specify which items are included or excluded, the design part and the stage.

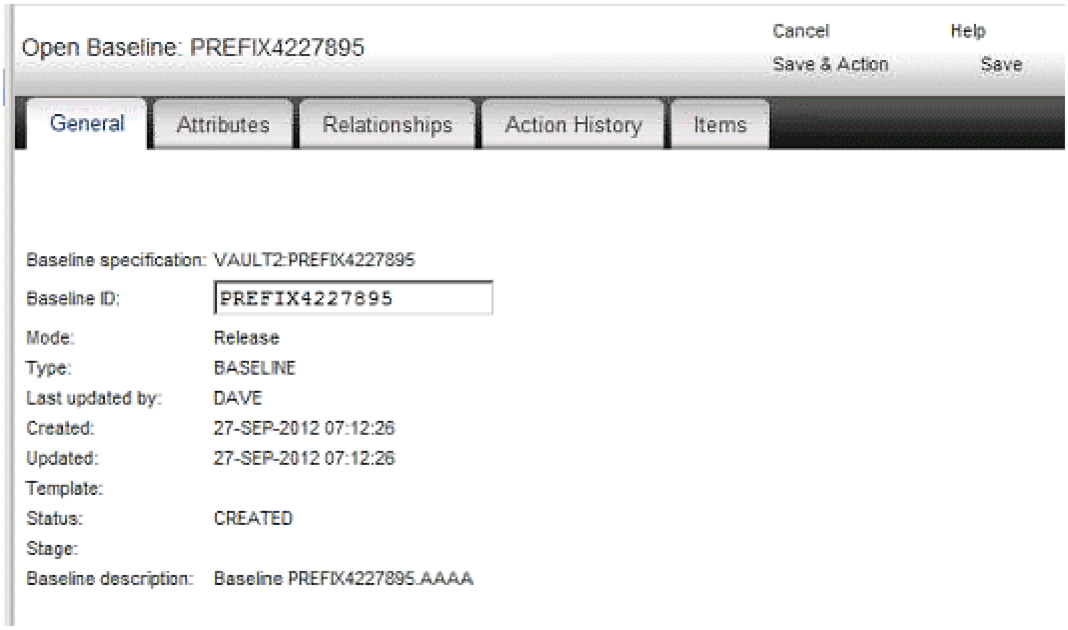
4 Click **OK** on the Turnover page. The dev package will move to **Turnover Complete**.

A baseline will be created in Dimensions CM under the specified target product.



Contents of: VAULT2.A

| | Name | Type Name | Last Updated By |
|---|---|---|---|
| □ | VAULT2:PREFIX4227663 | BASELINE | DAVE |
| □ | VAULT2:PREFIX4227851 | BASELINE | DAVE |
| □ | VAULT2:PREFIX4227895 | BASELINE | DAVE |



Open Baseline: PREFIX4227895

Baseline specification: VAULT2:PREFIX4227895
Baseline ID: PREFIX4227895
Mode: Release
Type: BASELINE
Last updated by: DAVE
Created: 27-SEP-2012 07:12:26
Updated: 27-SEP-2012 07:12:26
Template:
Status: CREATED
Stage:
Baseline description: Baseline PREFIX4227895.AAAA

The baseline will now be available to be specified as a deployment unit in Serena Release manager. Note that Serena Release Manager identifies and lists the deployment units based on the deployment unit projects that are selected for the release package, and any filters on status that are defined in the configuration files.

For details, see the *Release Manager Getting Started Guide*.

# Displaying Project Metrics with the Serena® Dashboard

Serena Development Manager can be used with Serena Dashboard, a rich, graphical reporting tool that allows you to carefully monitor key performance indicators (KPIs). The Dashboard is fully configurable, and can provide graphical information on any aspects of your projects that managers or executives need to track.

Documentation is provided separately for Serena Dashboard through the Serena Help server at http://help.serena.com/alm/sdb/index.html.

# Getting to Know Dimensions CM

With Dimensions CM, you can manage your source code assets by collaborating on shared development projects and streams with your team members. You can synchronize your local workspace with Dimensions CM, automatically delivering your changes to the shared repository, as well as copying other users' changes to your workspace. With Dimensions CM, you can collect files related to a milestone, such as a release build, into a baseline that can be deployed for testing or release.

With Development Manager, you can synchronize tasks from the Dev Tasks process app running on Serena Business Manager to Dimensions CM requests. Developers can then work on tasks in context of their favorite integrated development environment (IDE).

Dimensions CM provides a number of different clients to choose from, to best suite the different working styles on your team. You can use the Dimensions Web client, desktop client, IDE clients, or even command-line clients dependending on what best suits you.

To get started learning more about Dimensions CM, please see *Getting Started with Dimensions CM*.

# Integrating with HP Quality Center

Use the Development Manager Connector for HP Quality Center to synchronize defects between the Dev Change Request process app and Quality Center. When users submit and update defects in one system, the defects are automatically submitted and updated in the other system. In this way, if your QA teams use Quality Center to record test results, you can ensure that all work against a project is tracked in Serena Development Manager.

The heart of the Connector for HP Quality Center is a set of Web Services that provide an API to key Quality Center features, such as submitting and updating defects. SBM process apps can interact with Quality Center using these Web services.For instructions on installing and configuring the Connector, please see the *Development Manager Connector for HP Quality Center / ALM*.

Chapter 4

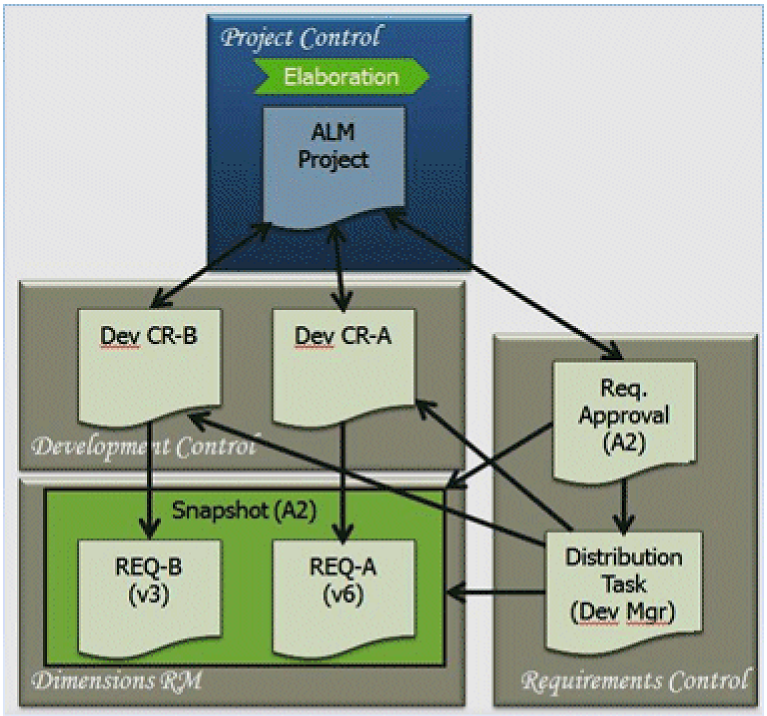
# Using the Requirements Manager Integration

# Introduction

If you have installed the Requirements Manager option in addition to the Development Manager components, you will have an additional *Req Approval* tab containing the Requirement Approval process app. For details of installing and using Serena RQM, consult the documentation provided with that product, in particular the:

- Serena Requirement Manager Installation and Configuration Guide
- Serena Requirement Manager Getting Started Guide

In RQM, the Business Analyst(s) will use an RM collection related to the ALM Project (and an RM document based on it) to break down and organize the project's requirements as needed. Once a Requirements Approval ticket has been approved, the Business Analyst distributes the requirements to one or more teams (i.e. Development, QA, Documentation) by issuing distribution tasks.



The Requirements Approval process app consists of three sub-projects:

- Approval Process
- Approval Polling
- Distribution Tasks

Serena Development Manager is largely concerned with the Distribution Tasks sub-project which is concerned with processing a distribution task. The other sub-projects are concerned with the approval and polling processes that results in the requirement approval ticket being approved. The distribution process creates a Distribution Task ticket corresponding to the requirements approval that the Development Manager will then own and can use to satisfy the individual requirements by creating dev change requests associated with them.

Requirements are created in RQM in relation to an ALM project. This project will be visible within the ALM Projects tab.

When working with requirements, you can

- Select a distribution task and view and create dev change requests based on the requirements related to it. See "Creating Dev Change Requests from Requirement Distribution Tasks" on page 59.
- Select a project and view and create dev change requests based on the requirements related to it. See "Creating Dev Change Requests for the Requirements in an ALM Project" on page 65.
- Select a dev change request and view and add or remove requirements related to it. See "Viewing and Associating Requirements for a Dev Change Request" on page 67.
- View the requirements that are included in a selected dev package. See "Viewing the Requirements for a Dev Package" on page 69.

# Creating Dev Change Requests from Requirement Distribution Tasks

Use the **Requirements Approvals** tab to view requirement distribution tasks that are assigned to you, and to create dev change requests related to their requirements. A single dev change request can be created for multiple requirements or multiple dev change requests can be created for a single requirement.

When a requirements approval has been approved in RQM, requirement distribution tasks are generated and assigned to the development manager.

A requirements approval contains one or more distribution tasks. Each distribution task consists of one or more requirements.

From a selected distribution task, you can:

- Create a single dev change request and assign one or more requests
- Create a separate dev change request for each requirement you select.
- View the requirements belonging to the distribution task and any dev change requests that have already been assigned to them.

## Viewing a Distribution Task

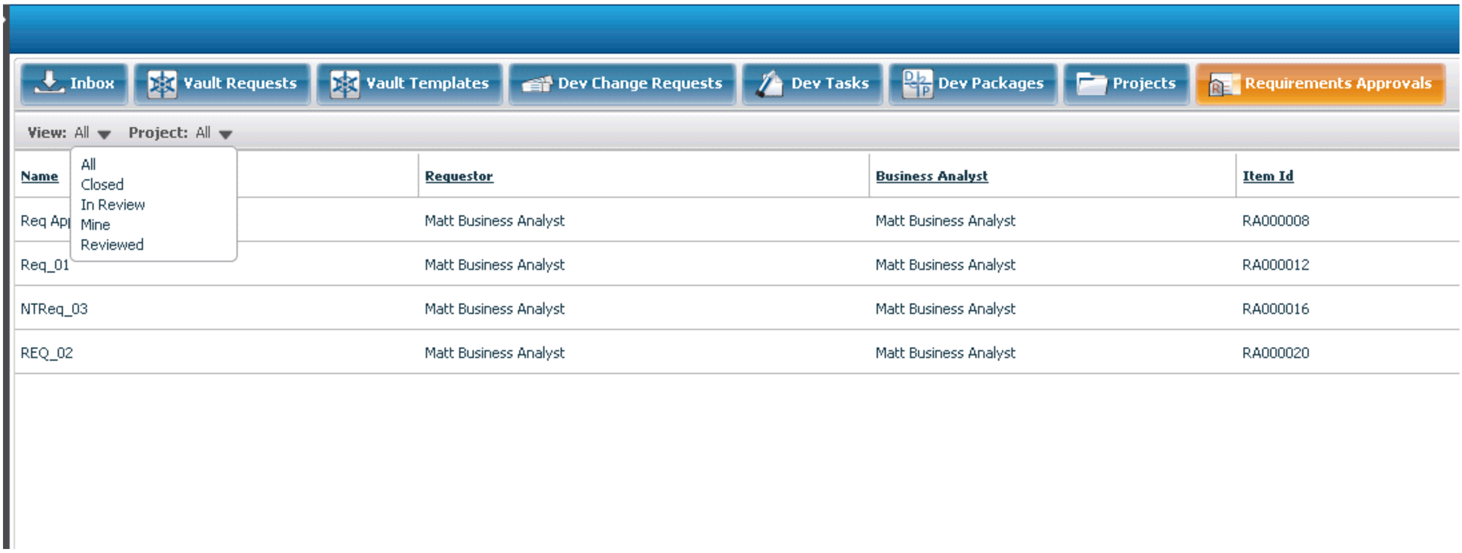
You select a distribution task by first selecting the requirement approval to which it belongs.

To view requirements approvals, click the **Requirements Approvals** tab.

By clicking the arrow for the **View** option, you can select:

- *Mine* to view change requests owned by you.
- Select a state to view only requirements approvals in a specific lifecycle state.

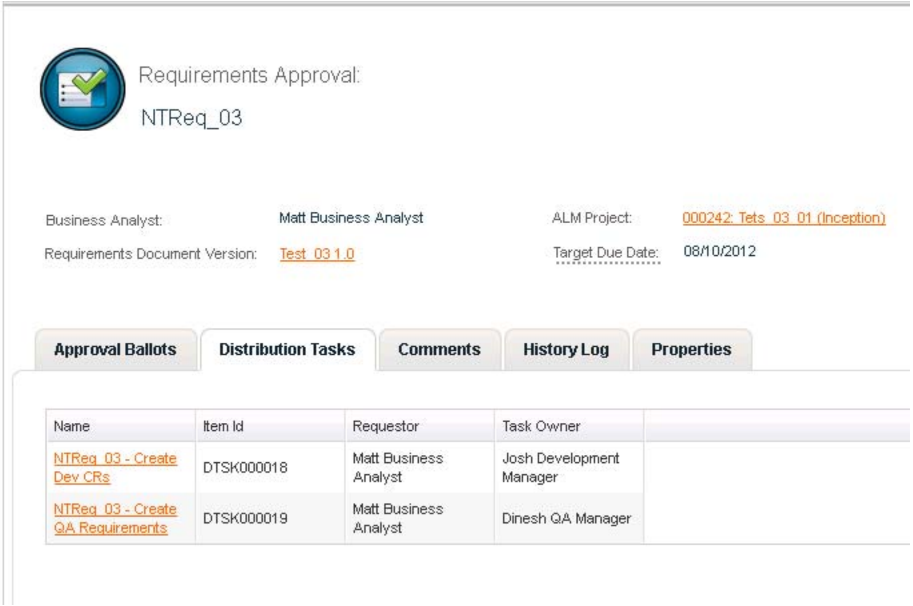
By clicking the arrow for the **Project** option, you can select dev change requests that are associated to a specific project.



| Name | Requestor | Business Analyst | Item Id |
|---|---|---|---|
| Req Ap | Matt Business Analyst | Matt Business Analyst | RA000008 |
| Req_01 | Matt Business Analyst | Matt Business Analyst | RA000012 |
| NTReq_03 | Matt Business Analyst | Matt Business Analyst | RA000016 |
| REQ_02 | Matt Business Analyst | Matt Business Analyst | RA000020 |

**To open a requirements approval:**

1 click on the corresponding row.



Business Analyst: Matt Business Analyst

Requirements Document Version: Test_03 1.0

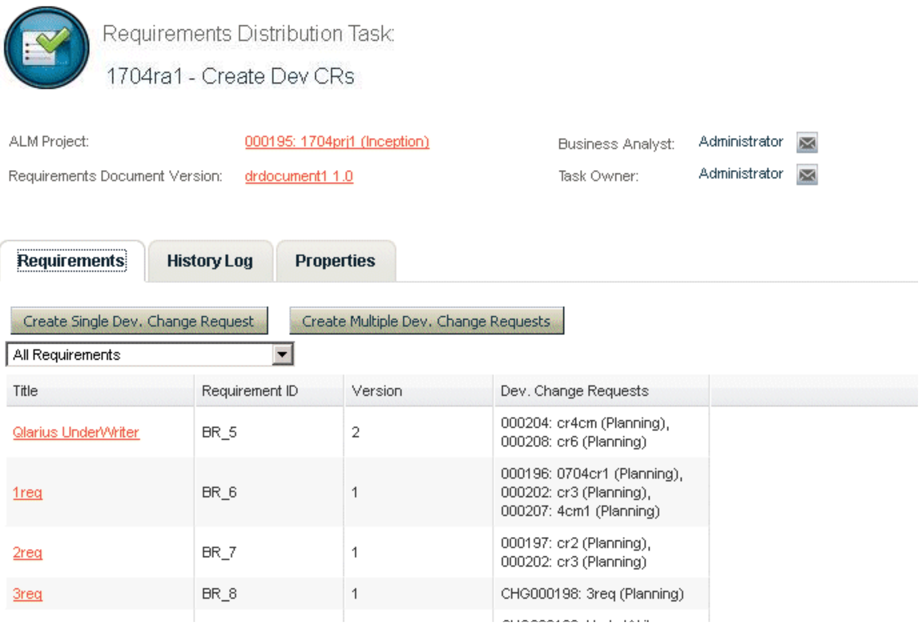
ALM Project: 000242: Tets_03_01 (Inception)

Target Due Date: 08/10/2012

| Name | Item Id | Requestor | Task Owner |
|---|---|---|---|
| NTReq_03 - Create Dev CRs | DTSK000018 | Matt Business Analyst | Josh Development Manager |
| NTReq_03 - Create QA Requirements | DTSK000019 | Matt Business Analyst | Dinesh QA Manager |

2 To view the distribution tasks, select the **Distribution Tasks** tab.

**3** To open a distribution task, click the title link.


Requirements
History Log
Properties
Create Single Dev. Change Request
Create Multiple Dev. Change Requests
All Requirements

Requirements Distribution Task:
1704ra1 - Create Dev CRs

| | | | |
|---|---|---|---|
| ALM Project: | 000195: 1704prj1 (Inception) | Business Analyst: | Administrator |
| Requirements Document Version: | drdocument1 1.0 | Task Owner: | Administrator |

| Title | Requirement ID | Version | Dev. Change Requests |
|---|---|---|---|
| Qlarius UnderWriter | BR_5 | 2 | 000204: cr4cm (Planning),<br>000208: cr6 (Planning) |
| 1req | BR_6 | 1 | 000196: 0704cr1 (Planning),<br>000202: cr3 (Planning),<br>000207: 4cm1 (Planning) |
| 2req | BR_7 | 1 | 000197: cr2 (Planning),<br>000202: cr3 (Planning) |
| 3req | BR_8 | 1 | CHG000198: 3req (Planning) |

**4** Select the **Requirements** tab

You can select:

- **All Requirements** to view all the requirements belonging to the distribution task
- **With Development Change Requests** to view only requirements that have a dev change request assigned to them.
- **Without Development Change Requests** to view only requirements that do not have a dev change request assigned to them.
- Click a requirement ID to view the requirement.

## Creating a Single Dev Change Request from One or More Requirements

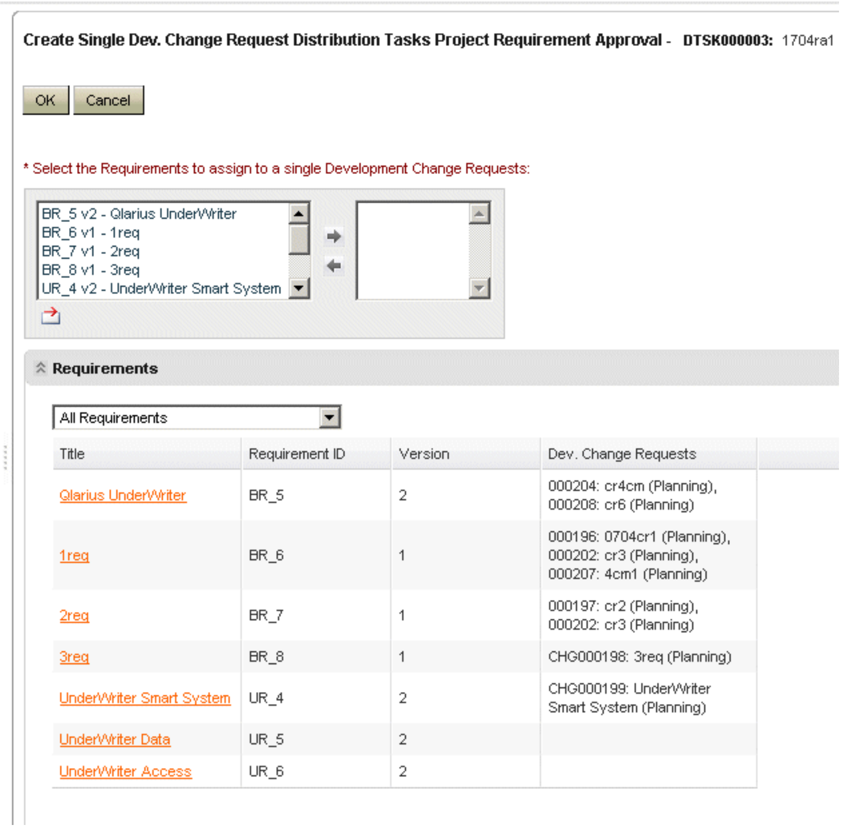
**To create a single dev change request from one or more requirements:**

**1** Select the distribution task as above.

**2** Select the Requirements tab.

**3** Click the **Create Single Dev Change Request** button.

Alternatively, you can select the ALM Project to which the requirements belong, and on the Change Requests tab, click the **Create Dev CR** button



**Create Single Dev. Change Request Distribution Tasks Project Requirement Approval - DTSK000003:** 1704ra1

* Select the Requirements to assign to a single Development Change Requests:

BR_5 v2 - Qlarius UnderWriter
BR_6 v1 - 1req
BR_7 v1 - 2req
BR_8 v1 - 3req
UR_4 v2 - UnderWriter Smart System

**Requirements**

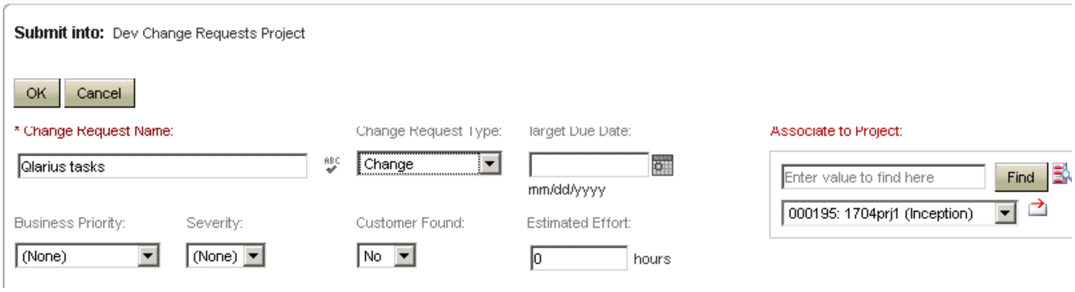
All Requirements

| Title | Requirement ID | Version | Dev. Change Requests |
|---|---|---|---|
| Qlarius UnderWriter | BR_5 | 2 | 000204: cr4cm (Planning),<br>000208: cr6 (Planning) |
| 1req | BR_6 | 1 | 000196: 0704cr1 (Planning),<br>000202: cr3 (Planning),<br>000207: 4cm1 (Planning) |
| 2req | BR_7 | 1 | 000197: cr2 (Planning),<br>000202: cr3 (Planning) |
| 3req | BR_8 | 1 | CHG000198: 3req (Planning) |
| UnderWriter Smart System | UR_4 | 2 | CHG000199: UnderWriter Smart System (Planning) |
| UnderWriter Data | UR_5 | 2 | |
| UnderWriter Access | UR_6 | 2 | |

**4** Select the requirements and move them to the right-hand list.

**5** Click OK.

A Submit Dev Change Request page is displayed.



**Submit into:** Dev Change Requests Project

* Change Request Name: Qlarius tasks

Change Request Type: Change

Target Due Date: mm/dd/yyyy

Associate to Project: Enter value to find here
000195: 1704prj1 (Inception)

Business Priority: (None)

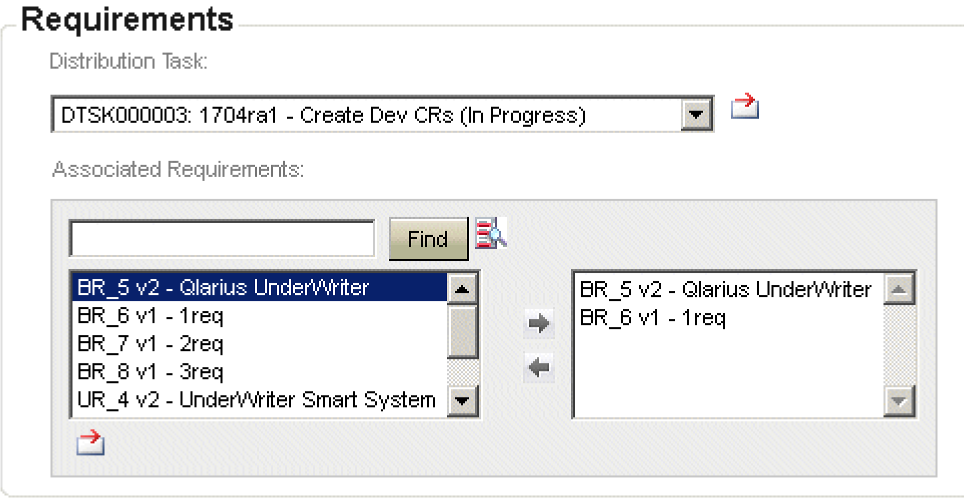
Severity: (None)

Customer Found: No

Estimated Effort: 0 hours

**6** Enter a name for the dev change request.

7 At the bottom of the page, the requirements you selected are listed, but you can add or remove requirements from the list.



Requirements

Distribution Task:

DTSK000003: 1704ra1 - Create Dev CRs (In Progress)

Associated Requirements:

BR_5 v2 - Qlarius UnderWriter
BR_6 v1 - 1req
BR_7 v1 - 2req
BR_8 v1 - 3req
UR_4 v2 - UnderWriter Smart System

BR_5 v2 - Qlarius UnderWriter
BR_6 v1 - 1req

8 Click **OK** and the dev change request will be created.

## Creating one or More Dev Change Requests From Requirements

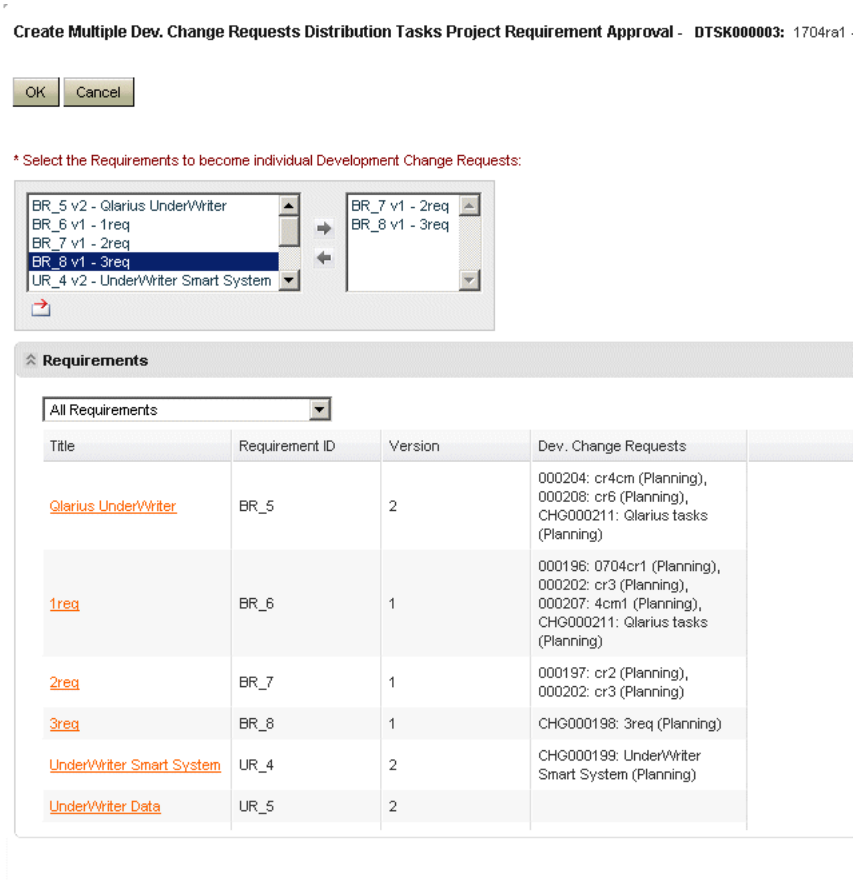
**To create one or more dev change requests from requirements:**

1 Select the distribution task as above.

2 Select the Requirements tab.

3 Click the **Create Multiple Dev Chenge Requests** button.

4 On the Create Multiple Dev CRs form, select the requirements. You can use the shift key to select multiple requirements.



Create Multiple Dev. Change Requests Distribution Tasks Project Requirement Approval - **DTSK000003:** 1704ra1 -

* Select the Requirements to become individual Development Change Requests:

BR_5 v2 - Qlarius UnderWriter
BR_6 v1 - 1req
BR_7 v1 - 2req
BR_8 v1 - 3req
UR_4 v2 - UnderWriter Smart System

BR_7 v1 - 2req
BR_8 v1 - 3req

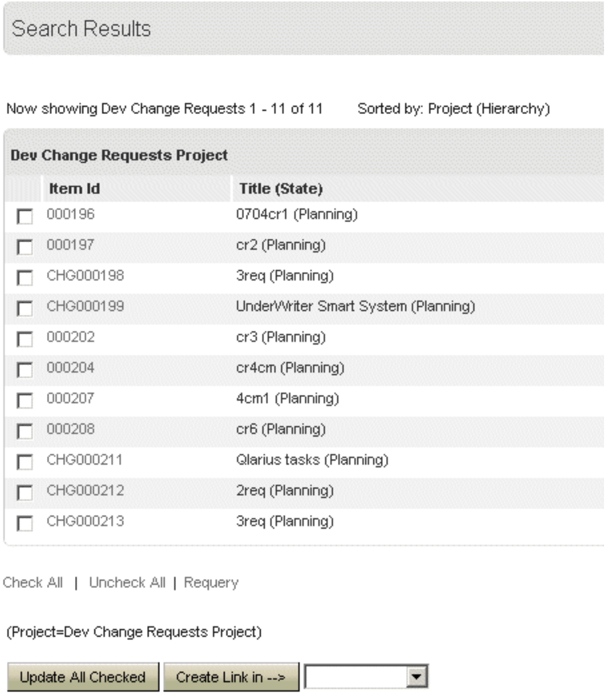
**Requirements**

All Requirements

| Title | Requirement ID | Version | Dev. Change Requests |
|---|---|---|---|
| Qlarius UnderWriter | BR_5 | 2 | 000204: cr4cm (Planning), 000208: cr6 (Planning), CHG000211: Qlarius tasks (Planning) |
| 1req | BR_6 | 1 | 000196: 0704cr1 (Planning), 000202: cr3 (Planning), 000207: 4cm1 (Planning), CHG000211: Qlarius tasks (Planning) |
| 2req | BR_7 | 1 | 000197: cr2 (Planning), 000202: cr3 (Planning) |
| 3req | BR_8 | 1 | CHG000198: 3req (Planning) |
| UnderWriter Smart System | UR_4 | 2 | CHG000199: UnderWriter Smart System (Planning) |
| UnderWriter Data | UR_5 | 2 | |

5 Click **OK**.

After some processing, the dev CRs will be created. If you go to the **Dev Change Requests** tab to look for the items owned by you, you will see that there is a dev change request created for each selected requirement with a title that is based on the title of the requirement and in the initial state *Planning*.



Search Results

Now showing Dev Change Requests 1 - 11 of 11 Sorted by: Project (Hierarchy)

Dev Change Requests Project

| | Item Id | Title (State) |
|---|---|---|
| □ | 000196 | 0704cr1 (Planning) |
| □ | 000197 | cr2 (Planning) |
| □ | CHG000198 | 3req (Planning) |
| □ | CHG000199 | UnderWriter Smart System (Planning) |
| □ | 000202 | cr3 (Planning) |
| □ | 000204 | cr4cm (Planning) |
| □ | 000207 | 4cm1 (Planning) |
| □ | 000208 | cr6 (Planning) |
| □ | CHG000211 | Qlarius tasks (Planning) |
| □ | CHG000212 | 2req (Planning) |
| □ | CHG000213 | 3req (Planning) |

Check All | Uncheck All | Requery

(Project=Dev Change Requests Project)

The ALM project to which they belong is the same one to which the requirements belong in RQM.

# Creating Dev Change Requests for the Requirements in an ALM Project

From the ALM Projects process app, you can:

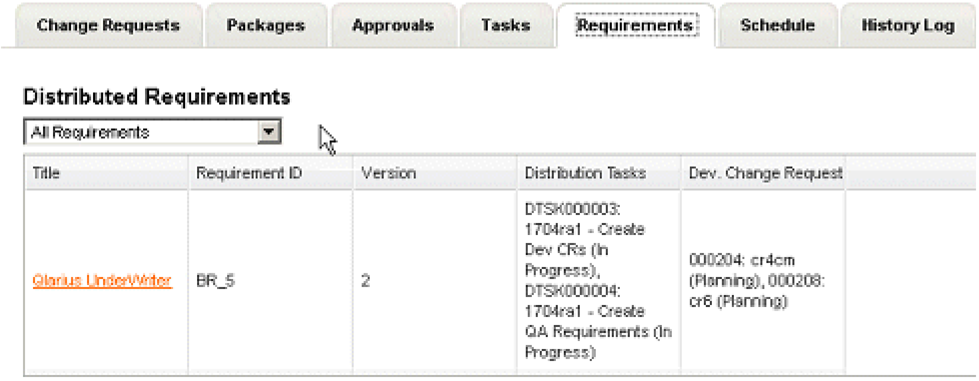
- View the requirements related to the project
- View the Requirement Approvals related to the project.

## Viewing the Requirements Related to a Project

**To view the requirements related to an ALM project:**

**1** Select the project

**2** Select the Requirements tab.



**Distributed Requirements**

All Requirements

| Title | Requirement ID | Version | Distribution Tasks | Dev. Change Request |
|---|---|---|---|---|
| Glarius UnderWriter | BR_5 | 2 | DTSK000003: 1704ra1 - Create Dev CRs (In Progress), DTSK000004: 1704ra1 - Create QA Requirements (In Progress) | 000204: cr4cm (Planning), 000208: cr6 (Planning) |

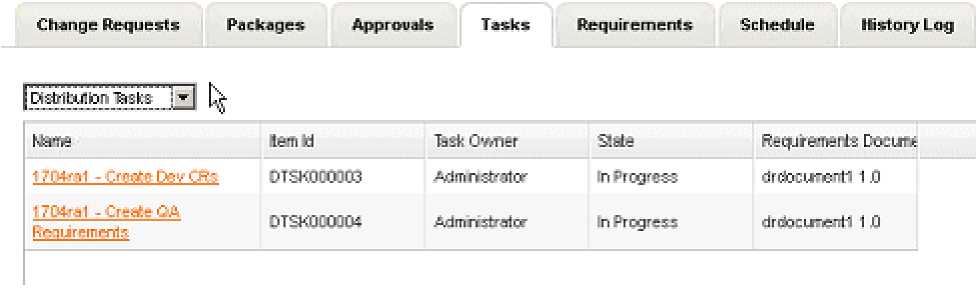
To view the requirements:

- To view all the requirements for all requirement approvals related to the project, select **All**.
- To view only requirements that have a dev change request assigned to them, select **With Dev CRs**.
- To view only requirements that do not have a dev change request assigned to them, select **Without Dev CRs**.
- To view an individual requirement, click the **Title** link.

Note that the requirements displayed here are only ones that have been distributed for this ALM project, i.e. have a distribution task created for them. It does not show all requirements related to the project.

**To view the distribution tasks for an ALM project:**

- Click the Tasks tab.
- Select Distribution Tasks from the drop down list



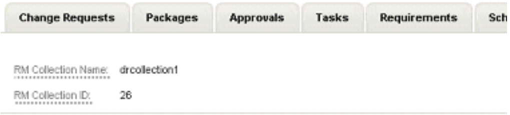
Distribution Tasks

| Name | Item Id | Task Owner | State | Requirements Docume |
|---|---|---|---|---|
| 1704ra1 - Create Dev CRs | DTSK000003 | Administrator | In Progress | drdocument1 1.0 |
| 1704ra1 - Create QA Requirements | DTSK000004 | Administrator | In Progress | drdocument1 1.0 |

To view a distribution task, click its item ID.

**To view the RM collections for an ALM project:**

Click the Dimensions RM tab.

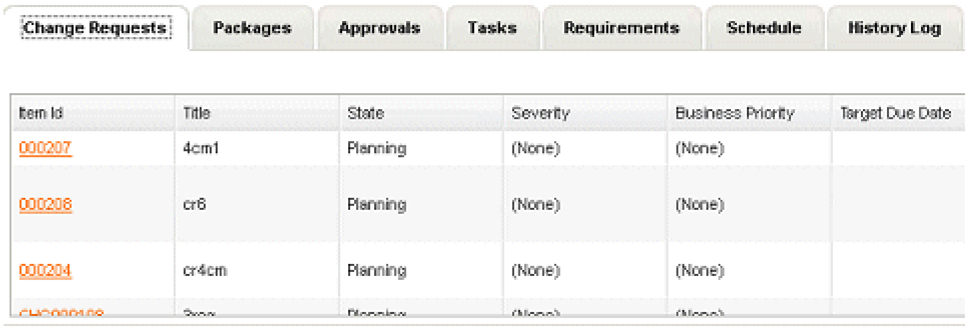


RM Collection Name: drcollection1

RM Collection ID: 26

**To view the requirements related to dev change requests for an ALM project:**

Click the Change Requests tab.

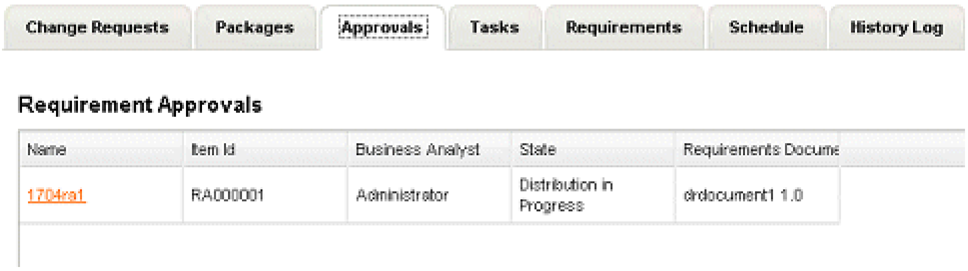


| Item Id | Title | State | Severity | Business Priority | Target Due Date |
|---|---|---|---|---|---|
| 000207 | 4cm1 | Planning | (None) | (None) | |
| 000208 | cr6 | Planning | (None) | (None) | |
| 000204 | cr4cm | Planning | (None) | (None) | |
| CHG000108 | 3xoo | Planning | (None) | (None) | |

Any requirement(s) that are related to a dev change request are shown in the Requirements column.

**To view the Requirement Approvals related to an ALM project:**

Click the Approvals tab.



Requirement Approvals

| Name | Item Id | Business Analyst | State | Requirements Docume |
|---|---|---|---|---|
| 1704ra1 | RA000001 | Administrator | Distribution in Progress | drdocument1 1.0 |

Any requirement approvals that are related to the project are listed. You can click the name to view the requirement approval.

# Viewing and Associating Requirements for a Dev Change Request

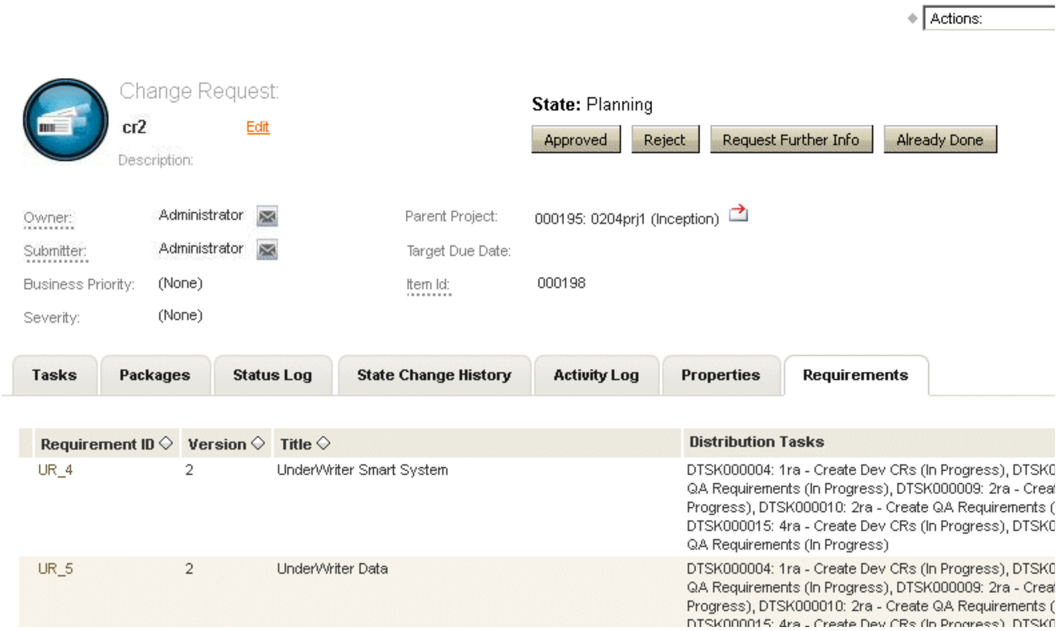
For a selected dev change request, you can:

- View the requirements that are already associated with it
- Edit the dev change request to add or remove associated requirements.

**To view the requirements for a dev change request:**

1 Select the dev change request.

2 Click the Requirements tab.



Change Request:
cr2
Description:

State: Planning

Owner: Administrator
Submitter: Administrator
Business Priority: (None)
Severity: (None)
Parent Project: 000195: 0204prj1 (Inception)
Target Due Date:
Item Id: 000198

| Requirement ID ◇ | Version ◇ | Title ◇ | Distribution Tasks |
|---|---|---|---|
| UR_4 | 2 | UnderWriter Smart System | DTSK000004: 1ra - Create Dev CRs (In Progress), DTSK0 QA Requirements (In Progress), DTSK000009: 2ra - Crea Progress), DTSK000010: 2ra - Create QA Requirements ( DTSK000015: 4ra - Create Dev CRs (In Progress), DTSK0 QA Requirements (In Progress) |
| UR_5 | 2 | UnderWriter Data | DTSK000004: 1ra - Create Dev CRs (In Progress), DTSK0 QA Requirements (In Progress), DTSK000009: 2ra - Crea Progress), DTSK000010: 2ra - Create QA Requirements ( DTSK000015: 4ra - Create Dev CRs (In Progress), DTSK0 |

3 To view a requirement, click the requirement ID.

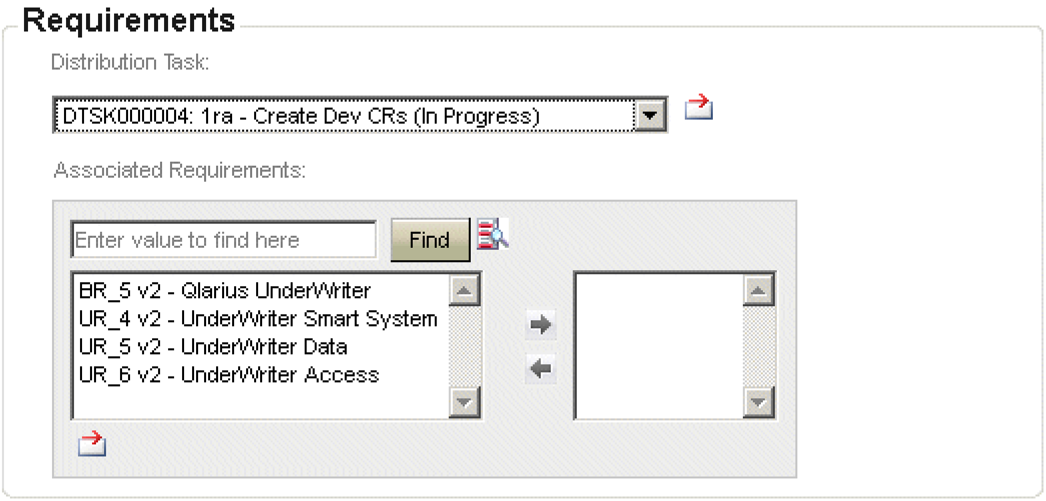
**To change the associated requirements for a dev change request:**

1 Select the dev change request.

2 Click the **Edit** link.

3 Scroll to the Requirements section.

4 Select a distribution task from the drop down list and click the **Find** button.



Requirements

Distribution Task:
DTSK000004: 1ra - Create Dev CRs (In Progress)

Associated Requirements:
Enter value to find here

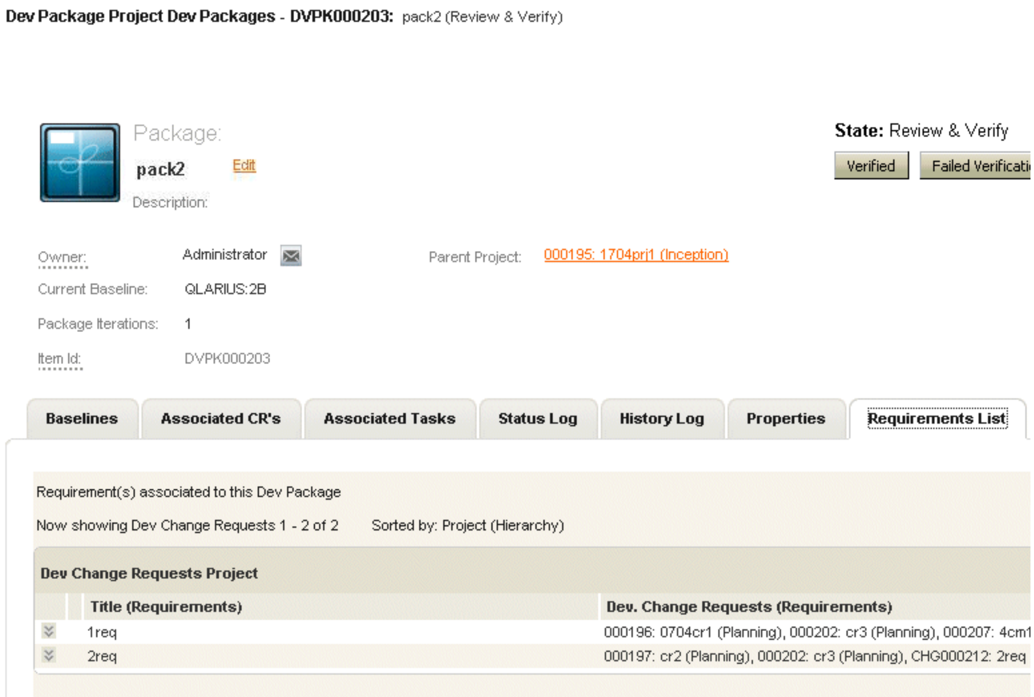
BR_5 v2 - Qlarius UnderWriter
UR_4 v2 - UnderWriter Smart System
UR_5 v2 - UnderWriter Data
UR_6 v2 - UnderWriter Access

5 Add or remove the requirements to the right-hand list that you want to associate with the dev change request.

# Viewing the Requirements for a Dev Package

Select the dev package.

- Click the **Requirements List** tab to view the requirements and their associated dev change requests.



**Dev Package Project Dev Packages - DVPK000203:** pack2 (Review & Verify)

Package:
pack2
Description:

**State:** Review & Verify

Owner: Administrator

Parent Project: 000195: 1704prj1 (Inception)

Current Baseline: QLARIUS:2B

Package Iterations: 1

Item Id: DVPK000203

Requirement(s) associated to this Dev Package

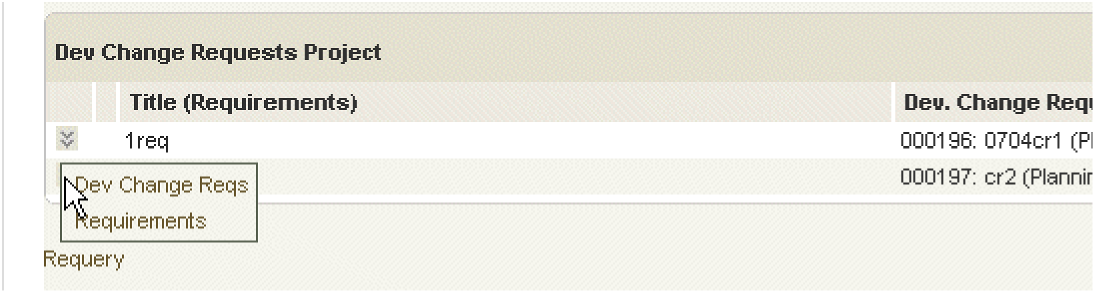
Now showing Dev Change Requests 1 - 2 of 2 Sorted by: Project (Hierarchy)

Dev Change Requests Project

| | Title (Requirements) | Dev. Change Requests (Requirements) |
|---|---|---|
| ⌄ | 1req | 000196: 0704cr1 (Planning), 000202: cr3 (Planning), 000207: 4cm1 |
| ⌄ | 2req | 000197: cr2 (Planning), 000202: cr3 (Planning), CHG000212: 2req |

You can click the ⌄ icon to the left of the requirement title and select:

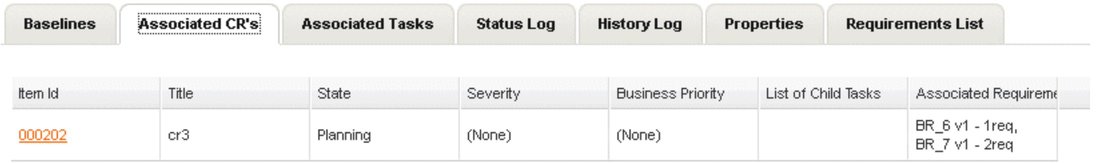
- **Requirements** to view the requirement
- **Dev Change Reqs** to view the dev change requests.



Dev Change Requests Project

| | Title (Requirements) | Dev. Change Requ |
|---|---|---|
| ⌄ | 1req | 000196: 0704cr1 (Pl |
| | | 000197: cr2 (Plannir |

- Click the **Associated CRs** tab to view the associated dev change requests.and any requirements that are assigned to them



| Item Id | Title | State | Severity | Business Priority | List of Child Tasks | Associated Requireme |
|---|---|---|---|---|---|---|
| 000202 | cr3 | Planning | (None) | (None) | | BR_6 v1 - 1req,<br>BR_7 v1 - 2req |