SERENA®

DEVELOPMENT MANAGER

Getting Started Guide

Serena Proprietary and Confidential Information

Trademarks

Serena, TeamTrack, StarTool, PVCS, Comparex, Dimensions, Prototype Composer, Mariner and ChangeMan are registered trademarks of Serena Software, Inc. The Serena logo, Version Manager and Mover are trademarks of Serena Software, Inc. All other products or company names are used for identification purposes only, and may be trademarks of their respective owners.

U.S. Government Rights

Any Software product acquired by Licensee under this Agreement for or on behalf of the U.S. Government, its agencies and instrumentalities is "commercial software" as defined by the FAR. Use, duplication, and disclosure by the U.S. Government is subject to the restrictions set forth in the license under which the Software was acquired. The manufacturer is Serena Software, Inc., 1850 Gateway Drive, 4th Floor, San Mateo, California, 94404-4061.

Product version: 4.0.1

Publication date: 2013-02-22

Table of Contents

Welcome to Serena Development Manager

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

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 development using Serena Development Manager.

Before You Begin

See the Readme for the latest updates and known issues.

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 1.1 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`

Guide to Serena Development Manager Documentation

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

Manual or Tutorial	Description
Serena ALM Installation and Configuration Guide	Describes how to install and configure the Serena ALM 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.
Development Manager Connector for HP Quality Center Web Service Reference	Provides Information on the web services provided for the QC connector for Development Manager.
Serena Development Manager Readme	Provides information about platform requirements and known issues for Development Manager.

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

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

Serena Dimensions CM and RM Installing the Serena License Manager	Describes installing the Windows version of SLM. (PDF manual)
Serena Dimensions CM User's Guide	A user guide to the Dimensions CM Client Tools. (PDF manual)
Serena Dimensions CM Process Configuration Guide	Describes how to configure the process model using the Administration Console. (PDF manual)
Serena Business Manager Connector for HP Quality Center Implementation Guide	Describes how to configure and use the SBM connector for HP Quality Center. (PDF manual)
Serena Requirements Manager Installation Guide	Describes how to install and configure Serena Requirements Manager. (PDF manual)
Serena Requirements Manager Getting Started Guide	Gives an overview of Serena Requirements Manager. (PDF manual)
Serena Release Manager Getting Started Guide	Gives an overview of Serena Release Manager. (PDF manual)
Serena ALM Dashboard Installation and Configuration Guide	Describes how to install and configure the Serena ALM Dashboard reporting tool. (PDF manual)

Access the documentation and tutorials for the Serena Development Manager suite through the Serena Help server at `http://help.serena.com/alm/info_center/doc_index/index.html`.

License and Copyright Information for Third-Party Software

License and copyright information for third-party software included in this release can be found as part of the software download available at:

`http://www.serena.com/support`

Introduction to Serena Development Manager

Welcome to Serena ALM

Serena Development Manager is a key part 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 Serena Development Manager

Serena Development Manager includes the following pieces:

- *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 key process apps of Development Manager include:
    - **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.
    - **Dev Control Orchestrations**: This process app ties together the other process apps using Serena Business Manager orchestrated workflows.
    - **Products and Components**: Define and manage a logical view of the parts, or components, of your system. Bring control and visibility to the changes you are making to these components.
    - **Agile Planning**: Use Agile planning methodology to manage backlogs of stories and tasks, and optionally link them to your development change requests and tasks.
- 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 assets in the Dimensions CM repository with 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.

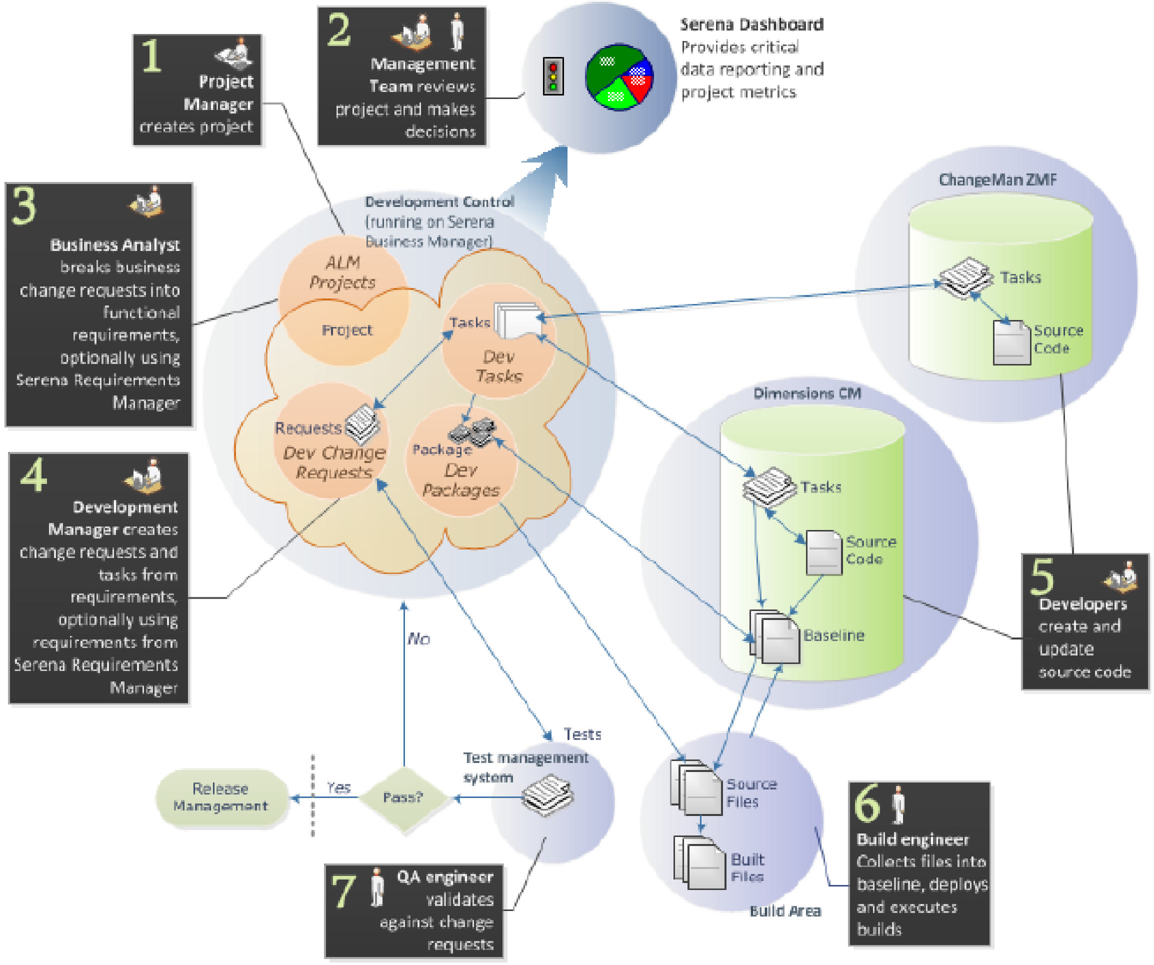
Related Topics

Usage Overview [page 12]

Usage Overview

The items 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*.

The following scenario does not include the use of products or components. For a discussion on products and components refer to Configuring and Using Products and Components.


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
4 Development Manager creates change requests and tasks from requirements, optionally using requirements from 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
5 Developers create and update source code
No
Tests
Test management system
Release Management
Yes
Pass?
7 QA engineer validates against change requests
Source Files
Built Files
Build Area
6 Build engineer Collects files into baseline, deploys and executes builds

Lets step through this scenario.

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 [page 20] . For details on creating projects, please see Creating and Managing Projects [page 30]. 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 [page 67].
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 [page 47].
3	The Business Analyst creates functional requirements based on business change requests. 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 [page 67].
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. To learn about the detailed workflow for change requests and tasks, please see Dev Change Requests Workflow [page 20] and Dev Tasks Workflow [page 22]. For information on creating change requests, please see Creating Dev Change Requests for the Requirements in an ALM Project [page 74]. For information on working with tasks, please see Creating and Working on Tasks [page 36].
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.

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 [page 24]. For details on creating and working with development packages, see Creating and Working with Development Packages [page 40].
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. 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.

Related Topics

User Roles [page 14]

Process Example [page 15]

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:

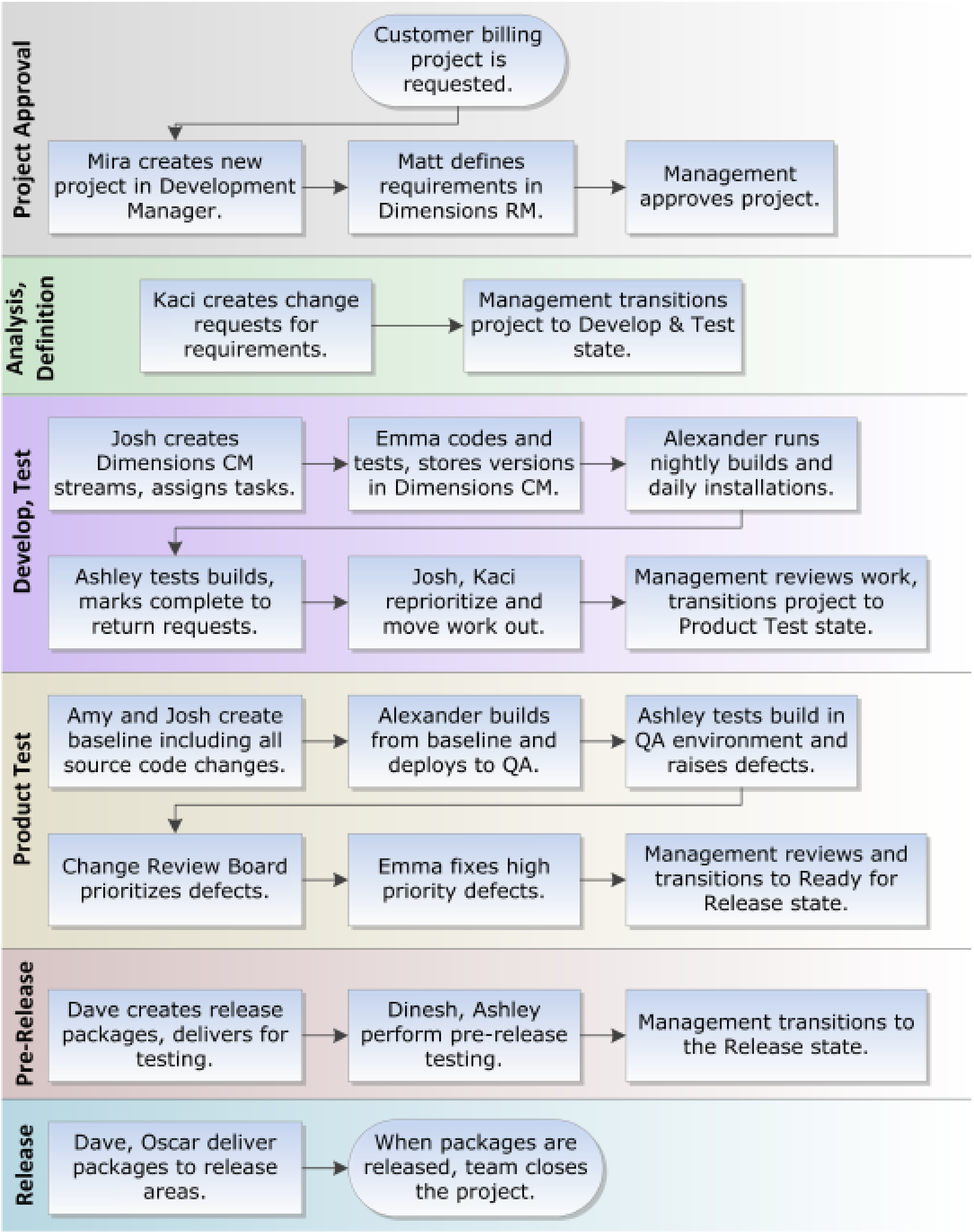
- Mira: Project Manager
- Matt: Business Analyst
- Kaci: Product Owner
- Eva: Component Manager
- Josh: Development Manager
- Emma: Development Engineer (member of Development Team)
- Amy: Build Manager (member of Build Team)
- Alexander: Build Engineer (member of Build Team)
- Dinesh: QA Manager (member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

Related Topics

Process Example [page 15]

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

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 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

Related Topics

Process Example [page 15]

Understanding the Out-of-the-Box Workflows

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.

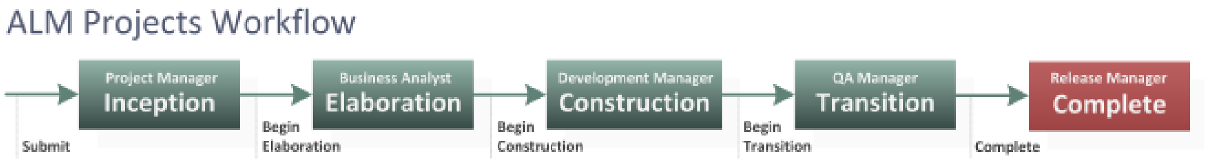
- **Projects**: You manage projects using the ALM Projects process app. You can relate projects as follows:
  - By relating a project to Dimensions CM streams and projects, you enable the association of source code and other managed assets related to the project.
  - By relating a project in ALM Projects 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.

- **Products**: You manage products using the Products and Components process app. Products represent long-lived shippable software systems, which may continue to exist over the duration of multiple ALM Projects.
- **Components**: You manage components using the Products and Components process app. Components are reusable, buildable sub-pieces of a larger software system. Components are composed together to form Products.

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

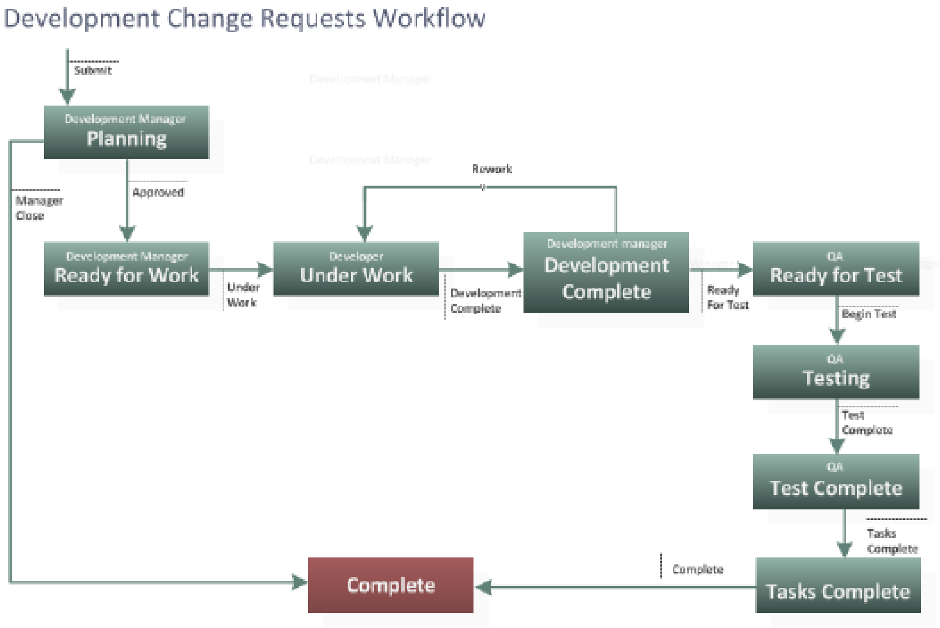
Change requests follow the workflow illustrated below. This workflow is implemented in the Dev Change Requests process app. This illustration is exported directly from SBM. Note that most work on change requests is expect to happen during the **Develop & Test** state in a project.

Related Topics

SBM Workflow for Change Requests [page 21]

Integrating Change Requests with HP Quality Center / ALM [page 21]

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.

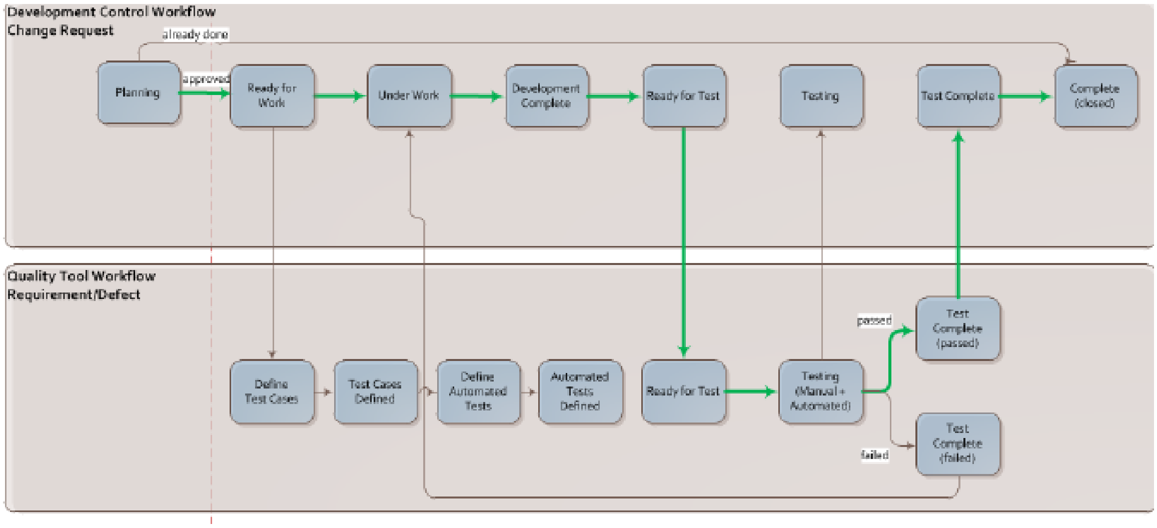
2. **Under Work**: Developers can now work on tasks associated with the request. As work is complete the developer checks updated source code files into Dimensions CM. Once this work associated with the request (and its related tasks) is ready to be compiled and tested, the request is transitioned by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

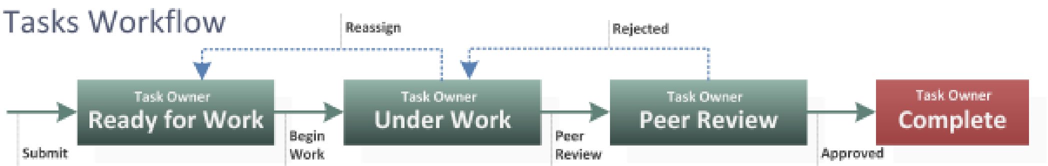


The test states in the change request workflow are mapped to workflow states in the Quality Center / ALM workflow. When the change request enters the *Ready for Test* state, it is associated with test cases in Quality Center.

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:

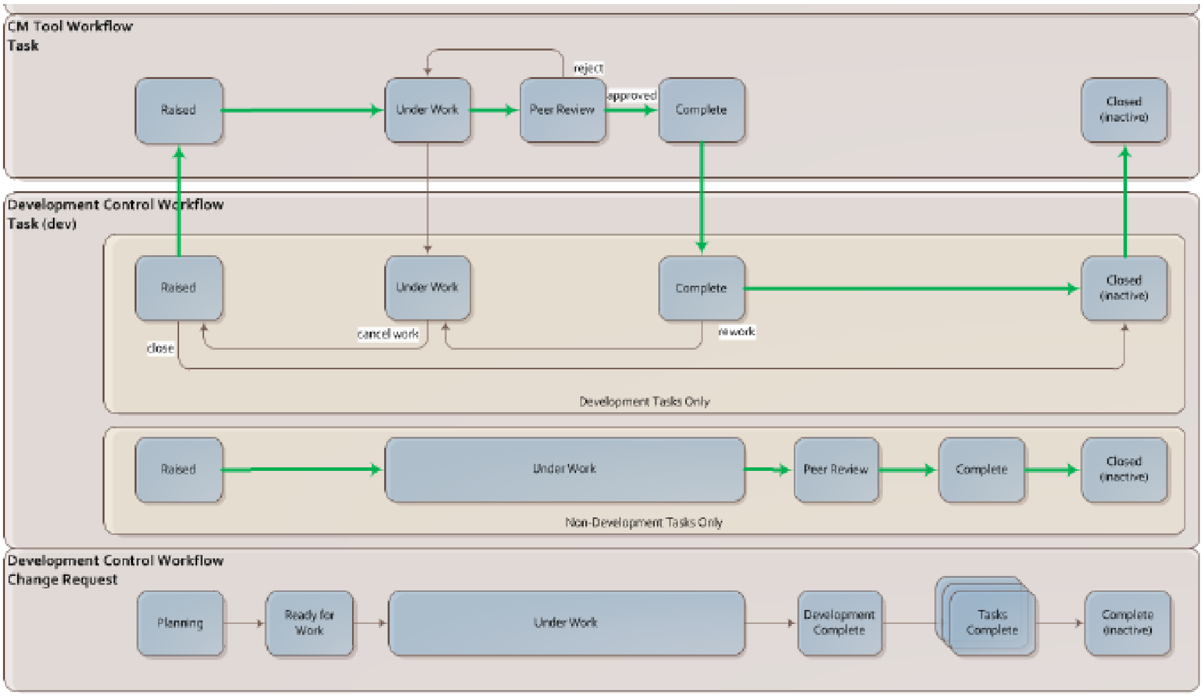
1. **Raised**: The task is submitted. If you are using Dimensions CM, a related change request is also created in Dimensions CM or ChangeMan ZMF. To start work, the Development Manager assigns an owner and transitions to the **Under Work** state. The related item in Dimensions CM or ChangeMan ZMF is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

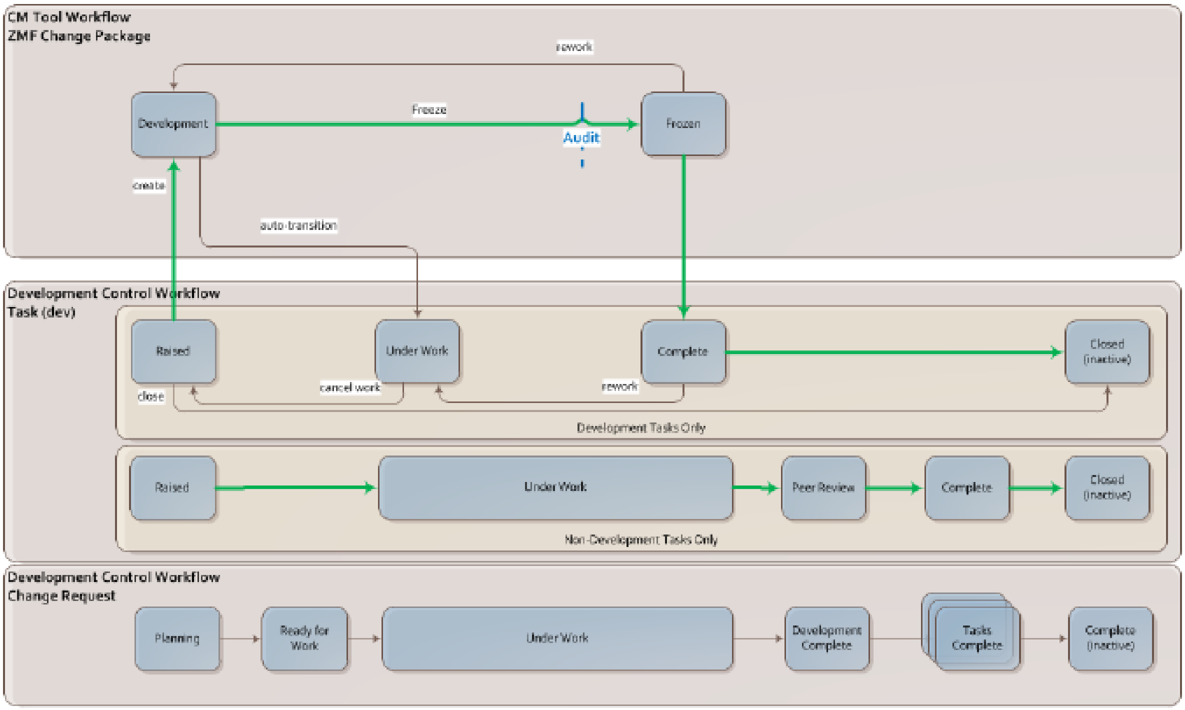
The following illustration clarifies the relationship between tasks in Dev Tasks and related tasks in Dimensions CM. Note that this is just one example of how tasks in Development Control might be linked to tasks in Dimensions CM for purposes of tracking work.



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 complete, 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

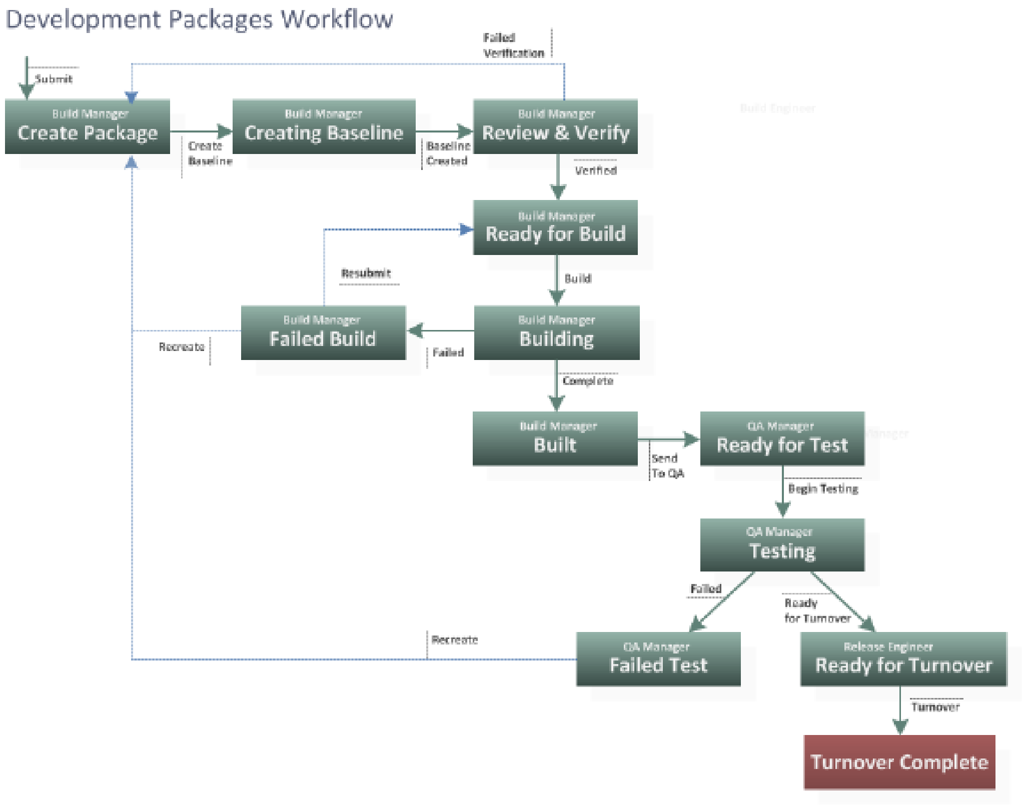
Use the Dev Packages process app to manage the creation and validation of release packages and baselines, including testing builds and approving for final turnover. Most of the work managed within the Dev Packages process app takes place during the **Product Test** project state.

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.

Related Topics

Dev Packages Workflow Diagram [page 25]

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.

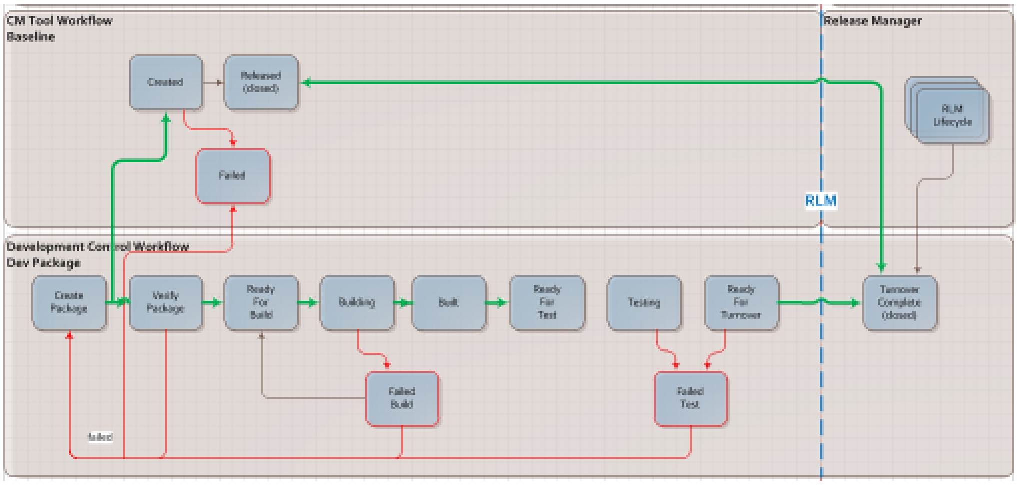
4. **Ready for Test** and **Testing**: QA staff test the deployed build. Information about the packages is synchronized to requirements in the test management tool. Test cases for validating those requirements are associated with the requirements. QA run their tests and update with pass / fail information, 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.



Products and Components

The product and component states (and corresponding SBM states) are:

- Draft (inactive)
- In use (active)
- Deprecated (active)
- Out of Use (inactive)

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.

Related Topics

What Types of Development Projects Will You Manage? [page 26]

Is Your Development Process Agile? [page 27]

What Are Your Workflow States? [page 27]

How Do You Manage Test Cases? [page 27]

What Reports Does Your Organization Need? [page 27]

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 [page 47]. To learn more about configuration Dashboard views, please see the *Serena Development Manager Installation and Configuration Guide*.

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
   ```

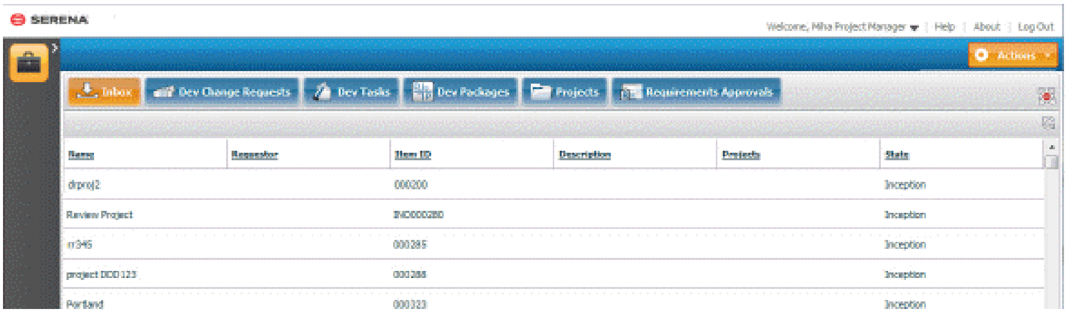
   For example, if the server is called dev and the port number is 8085, and you are using the Serena-supplied custom shell:

   ```
   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		IND000280			Inception
rr345		000285			Inception
project DDD123		000288			Inception
Portland		000323			Inception

NOTE If your role allows you to use products and components, the Products and Components tabs will be displayed.

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.

Related Topics

Creating one or More Dev Change Requests From Requirements [page 72]

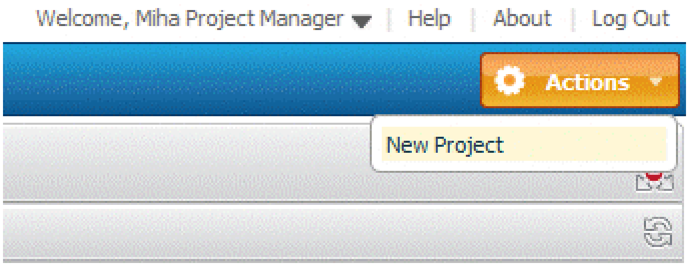
Example Project in ALM Projects [page 31]

Displaying and Creating Related Change Requests [page 33]

Creating and Relating Development Packages [page 33]

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:

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

- The type of project, from the **Project 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

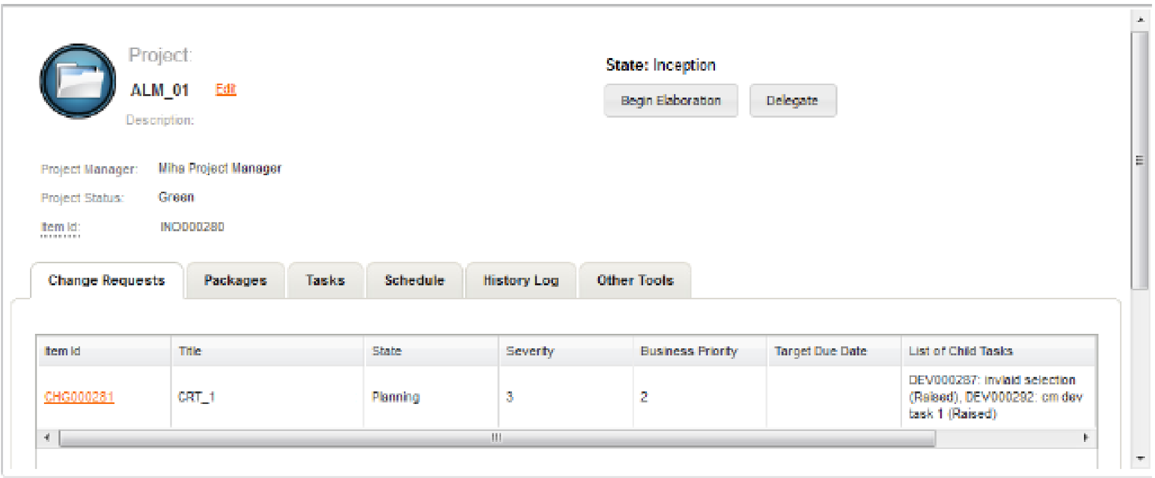
- Configuration Management Settings. You can select ChangeMan ZMF, Dimensions CM, or both as a Configuration Management provider in which to carry out your development work.

- Components: You can relate the project to an affected product and components, and used components. This may be done as an edit, post creation.

NOTE To fill in the "Component" or "Configuration Management Settings" sections of the form for Dimensions CM projects, you will need to be a registered user in Dimensions CM (so it may be the Development Manager who fills in these fields later).

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

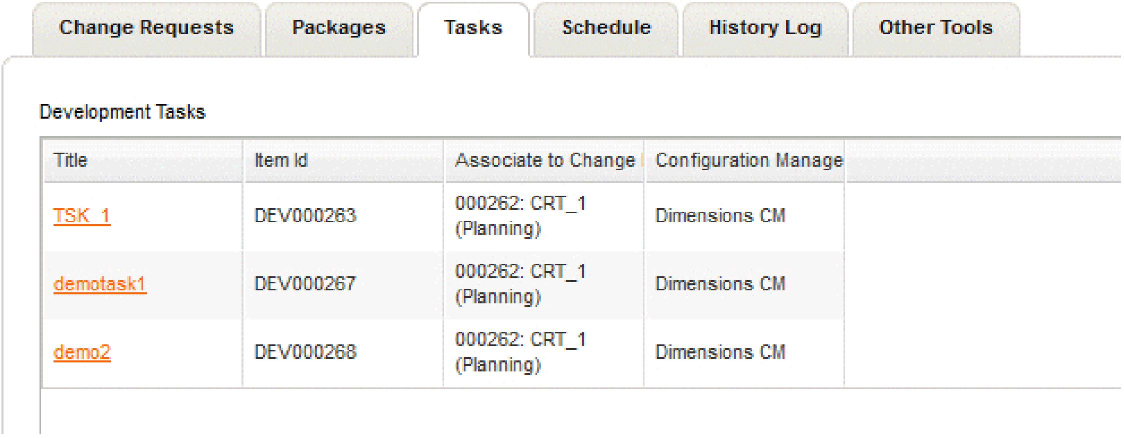
Description:

State: Inception

Project Manager:	Miha Project Manager
Project Status:	Green
Item Id:	INC000280

Item Id	Title	State	Severity	Business Priority	Target Due Date	List of Child Tasks
CHG000281	CRT_1	Planning	3	2		DEV000287: invalid selection (Raised), DEV000292: cm dev task 1 (Raised)

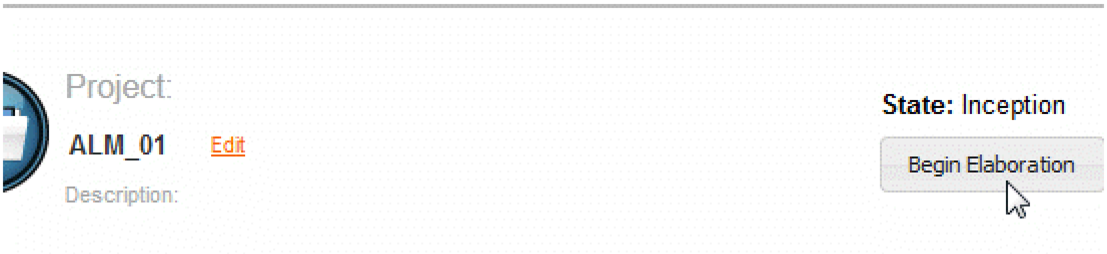
Clicking the Tasks tab shows that it has three tasks.



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

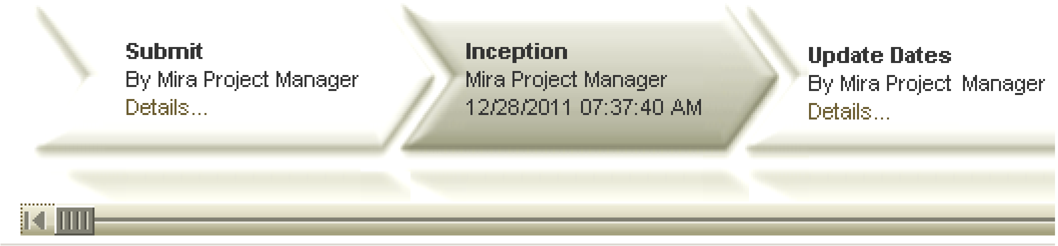
You can transition a project to the next state in the lifecycle by pressing the appropriate transition button, such as **Begin Elaboration** or **Begin Construction**. Note that you need the required role, for example Project Manager, to transition the project, otherwise the required button will not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

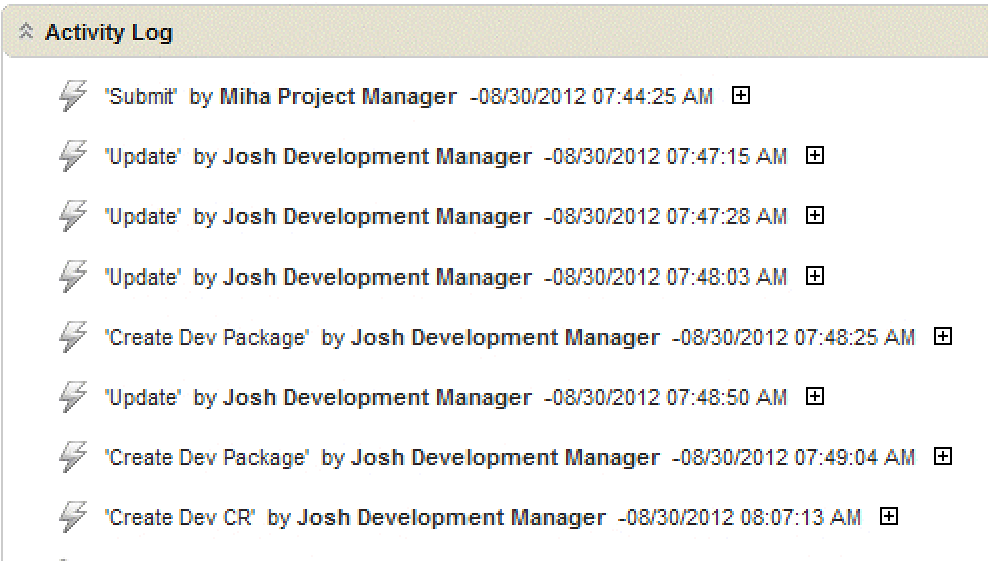


Submit
By Mira Project Manager
Details...

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

Update Dates
By Mira Project Manager
Details...

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 ⊞

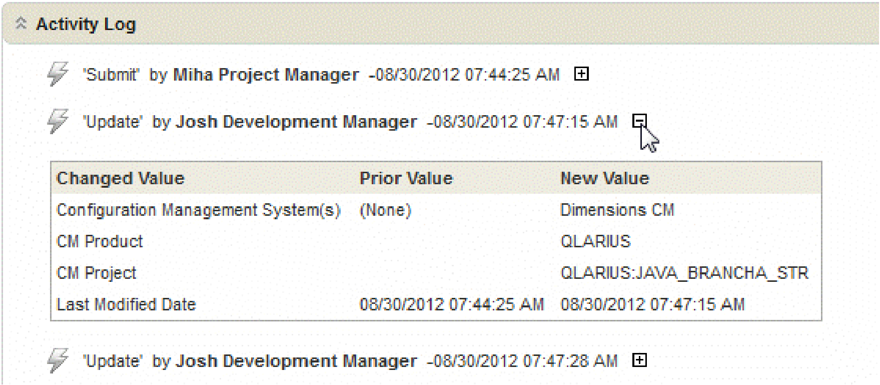
'Create Dev Package' by **Josh Development Manager** -08/30/2012 07:48:25 AM ⊞

'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.



'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.

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.

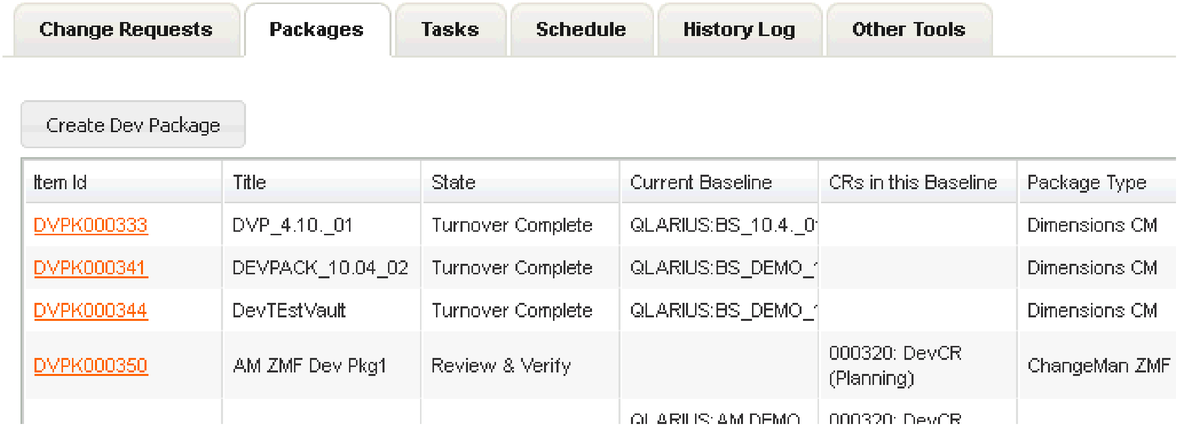
To create new change requests, click the **Create Dev CR** button.

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 [page 35] 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.



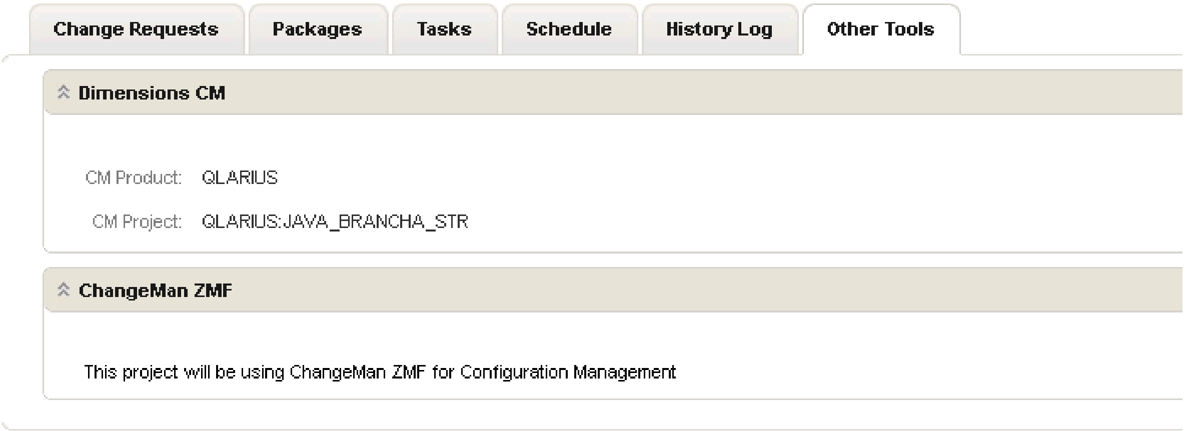
Item Id	Title	State	Current Baseline	CRs in this Baseline	Package Type
DVPK000333	DVP_4.10._01	Turnover Complete	QLARIUS:BS_10.4._0		Dimensions CM
DVPK000341	DEVPACK_10.04_02	Turnover Complete	QLARIUS:BS_DEMO_		Dimensions CM
DVPK000344	DevTEstVault	Turnover Complete	QLARIUS:BS_DEMO_		Dimensions CM
DVPK000350	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 [page 41] 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

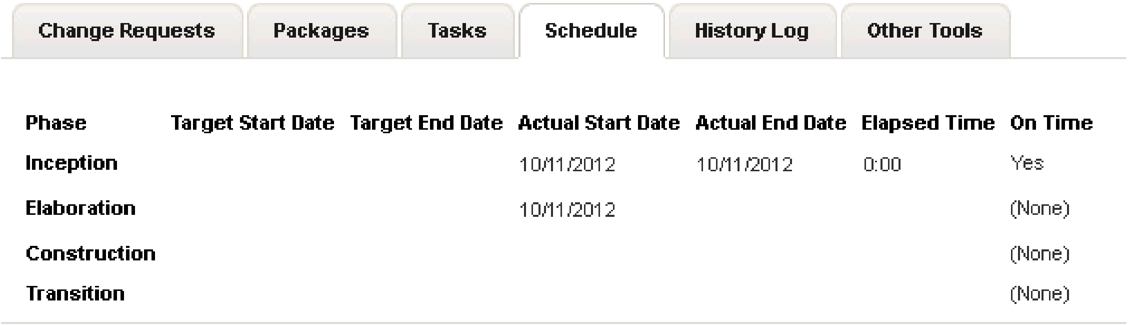
CM Project: QLARIUS:JAVA_BRANCHA_STR

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.

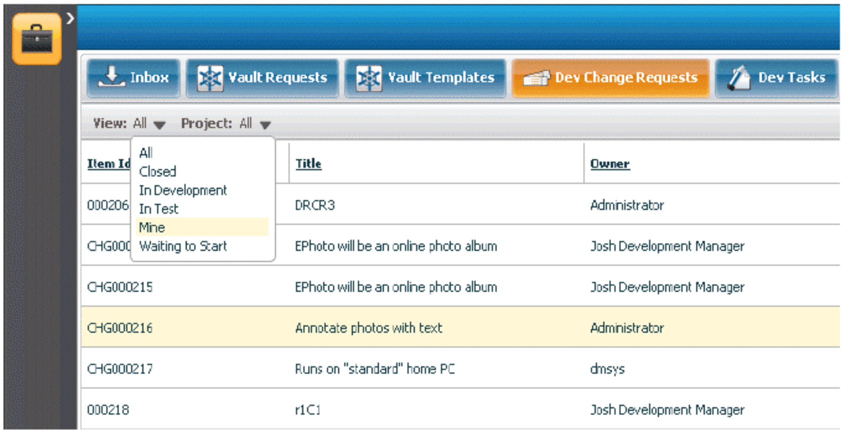


Phase	Target Start Date	Target End Date	Actual Start Date	Actual End Date	Elapsed Time	On Time
Inception			10/11/2012	10/11/2012	0:00	Yes
Elaboration			10/11/2012			(None)
Construction						(None)
Transition						(None)

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.

You can filter the list according to various criteria.



View: All Project: All

All
Closed
In Development
In Test
Mine
Waiting to Start

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

Related Topics

Creating New Change Requests [page 35]

Working on Change Requests [page 35]

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 *Create Dev Change Request* form appears.

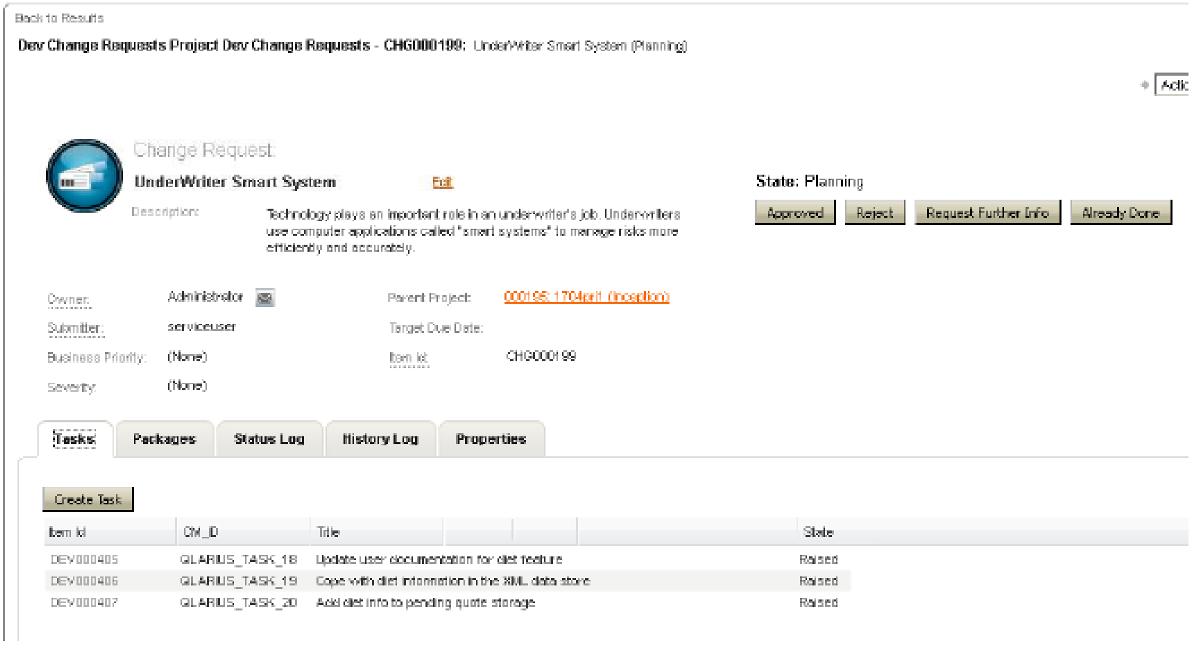
If you clicked the **Create Dev CR** button from a selected project, you will not see the **Associate to Project** fields as the change request will automatically be related to the selected project, otherwise you will need to select a project

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 **Change Request 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

State: Planning

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.

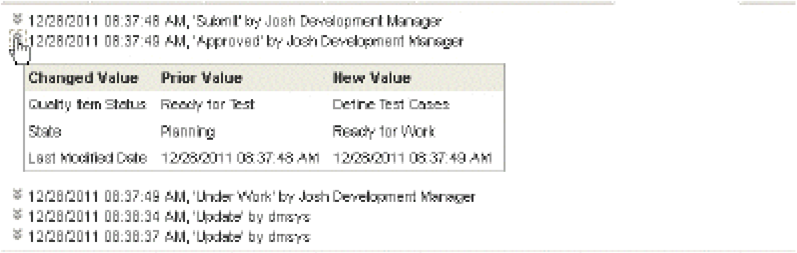
Owner:	Administrator	Parent Project:	000195: 1704prj1 (Inception)
Submitter:	serviceuser	Target Due Date:	
Business Priority:	(None)	Item Id:	CHG000199
Severity:	(None)		

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

NOTE If you have a role that allows you to work with products and components, a Component tab that displays affected products and components.

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.
- If you are using the Products and Components app you can display the affected product and relate the change request to one or more affected components.
- 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

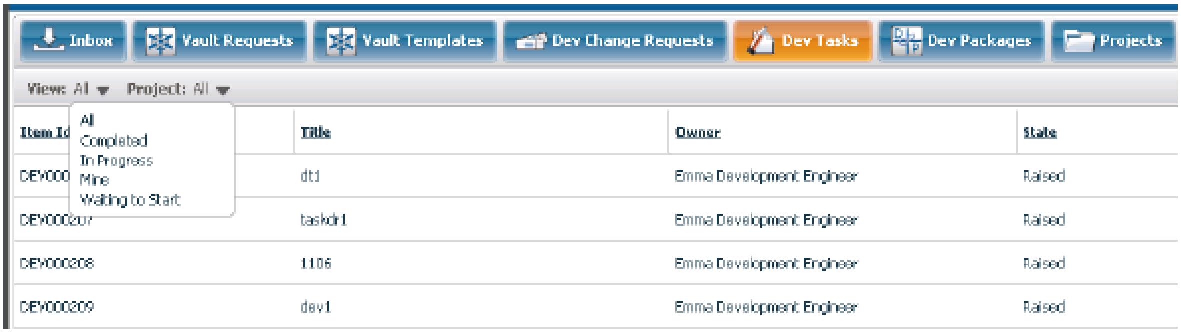
From the Dev Tasks tab, you can list, update, and report on development tasks. You can relate tasks to projects in the **Projects** tab, and relate tasks to Dimensions CM tasks.

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

Related Topics

Creating New Tasks [page 37]

Creating New Tasks [page 37]

Working on Tasks [page 39]

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.

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

- From the **Task Type** field, the type of task, such as **Design** or **Development**.
- The title and description.
- The owner and estimated effort.
- If you are using the Products and Components app you can relate the change request to one affected component.

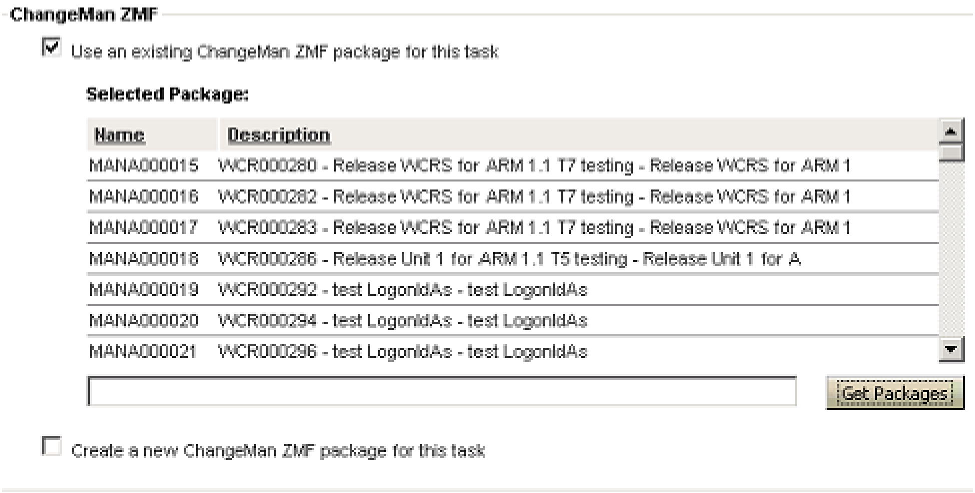
If the task type is Development, you will need to select a **Configuration Management System**. The selection available here depends on what was selected for **Configuration Management Settings** for the owning project, and by default, can be Dimensions CM or ChangeMan ZMF.

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

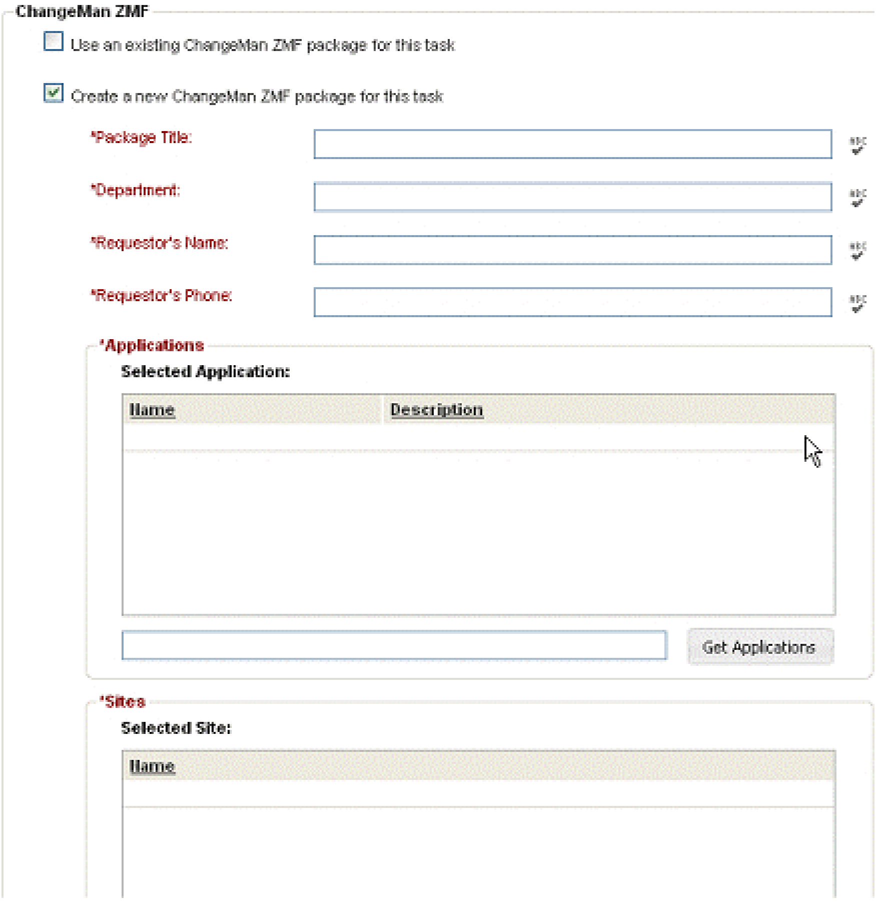
☑ Use an existing ChangeMan ZMF package for this task

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

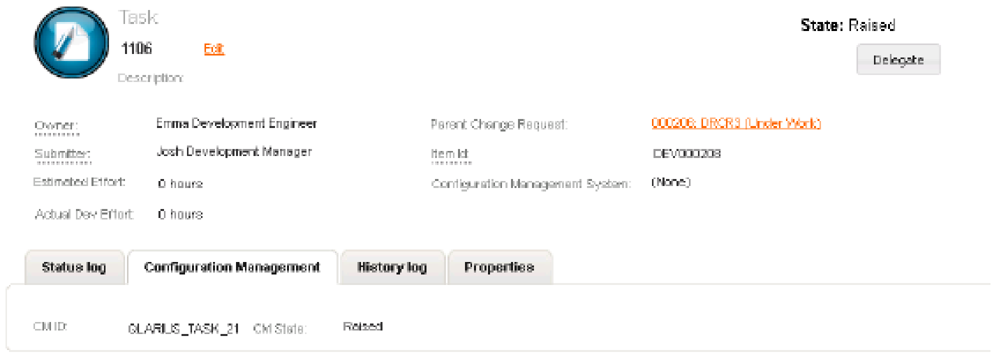
*Sites

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.



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: GLARUS_TASK_21 CM State: Raised

From a task, you can:

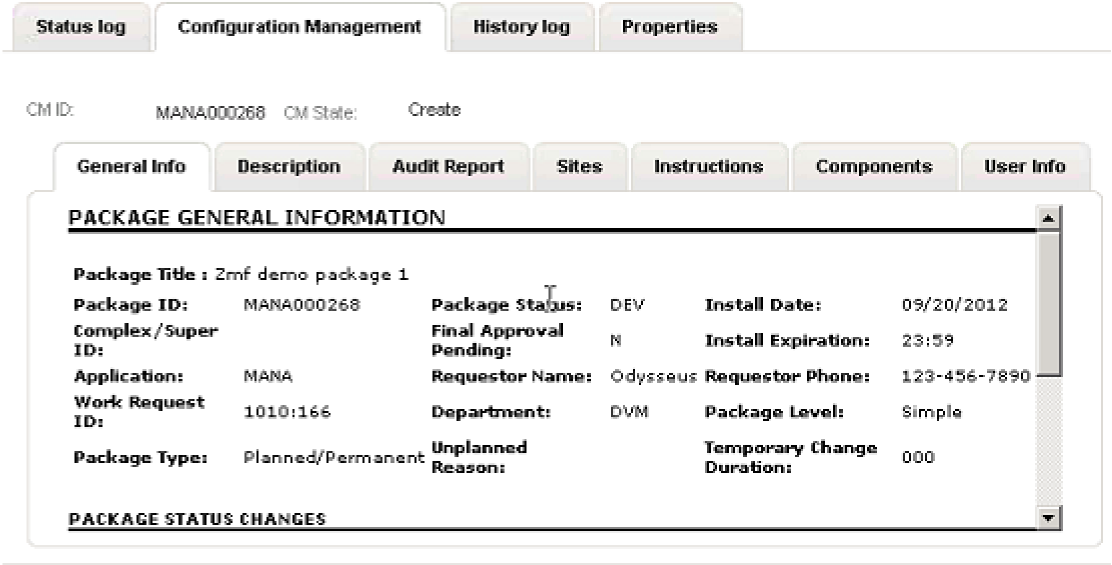
- Review information about the task, including its change history, Dimensions CM relationships, Quality Center relationships, associated change requests, and attachments.

- 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

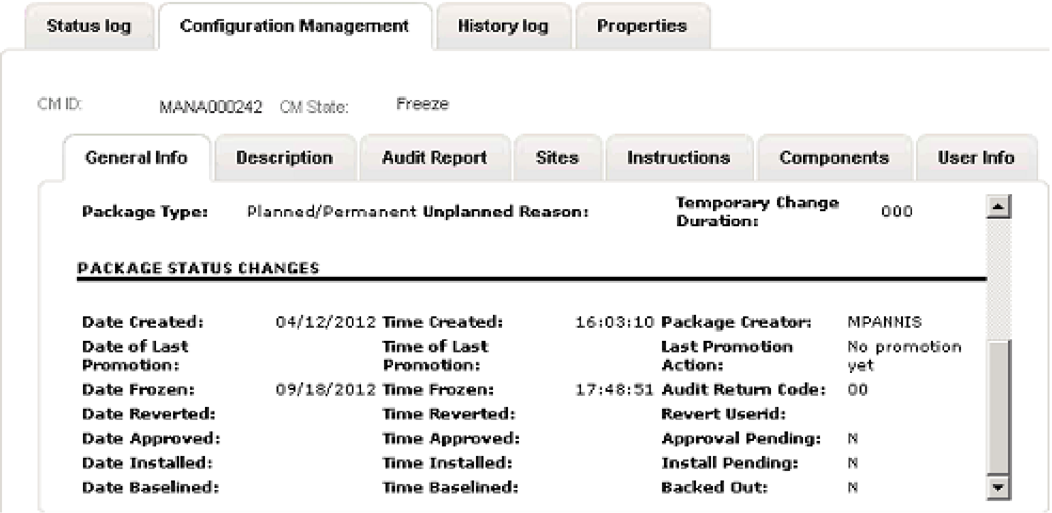
Package Title : Zmf demo package 1

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

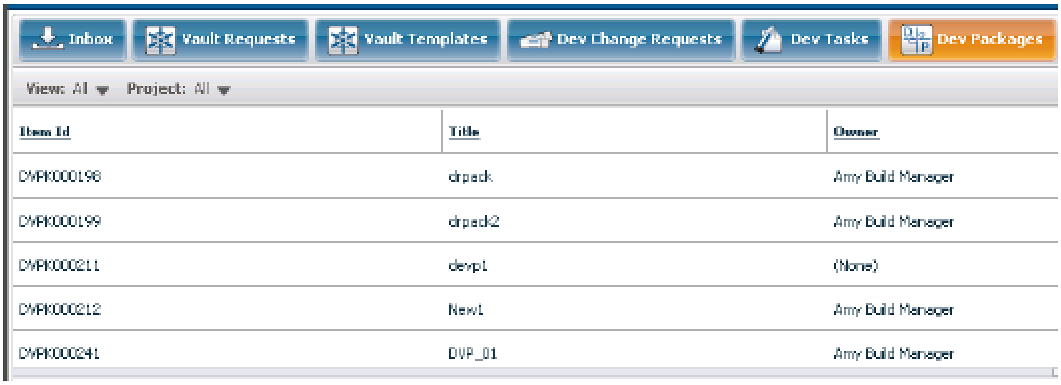
Once created and delegated to a development engineer, the code is worked on in ChangeMan ZMF. The change package is then Moved to **Frozen**, but it must be **audited** first. These transitions can be performed from either ChangeMan ZMF or Development Control. When the package is **Frozen** it is automatically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

Products and Components

If you are using products and components and the Dev Task affects a component that has a related CM design part, a CM request will be created in the corresponding design part.

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.



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 dev package either:

- By clicking the **Actions** button and selecting *New Dev Package*.
- 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.

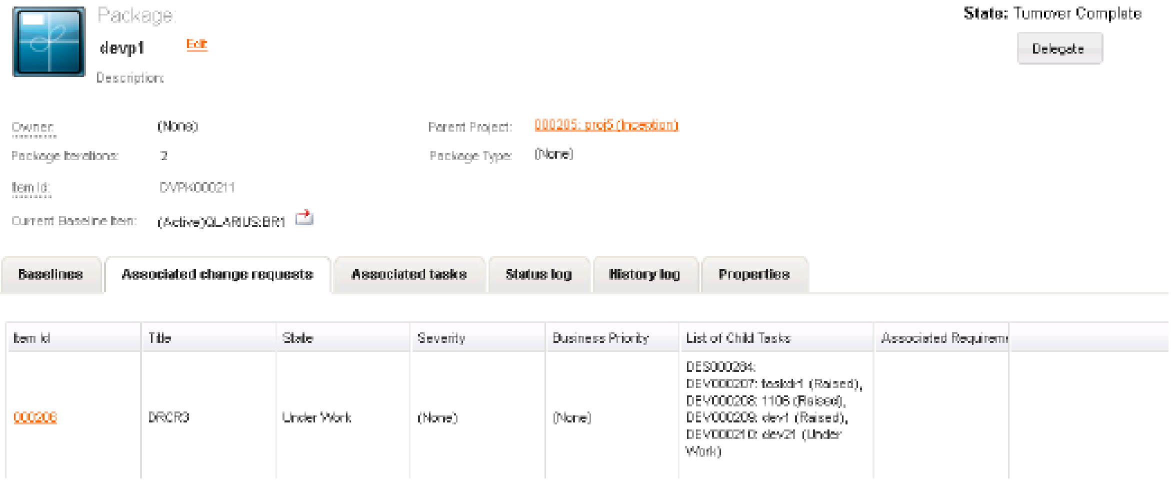
When you create a new dev package, a page like the following appears.

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:

State: Turnover Complete
Delegate

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 Requirem
000208	DRCR3	Under Work	(None)	(None)	DES000284: DEV000207: taskdr1 (Raised), DEV000208: 1106 (Raised), DEV000209: dev1 (Raised), 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:

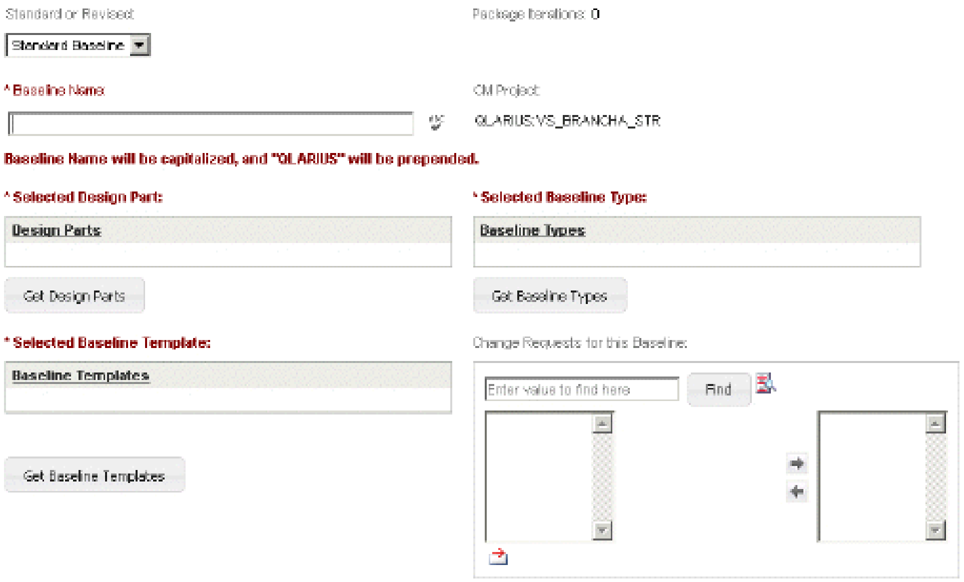
1. From a development package that is in the **Create Package** state, click the **Create Baseline** button.

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..

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

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



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.

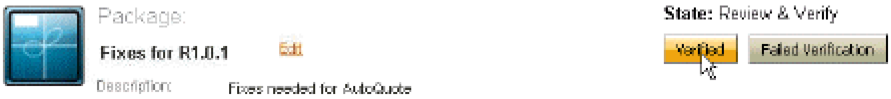
   To use this field, enter part of the name of the change requests and click the **Find** button to locate them. Then select the required change request and click the arrow to move them to the right-hand list

   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

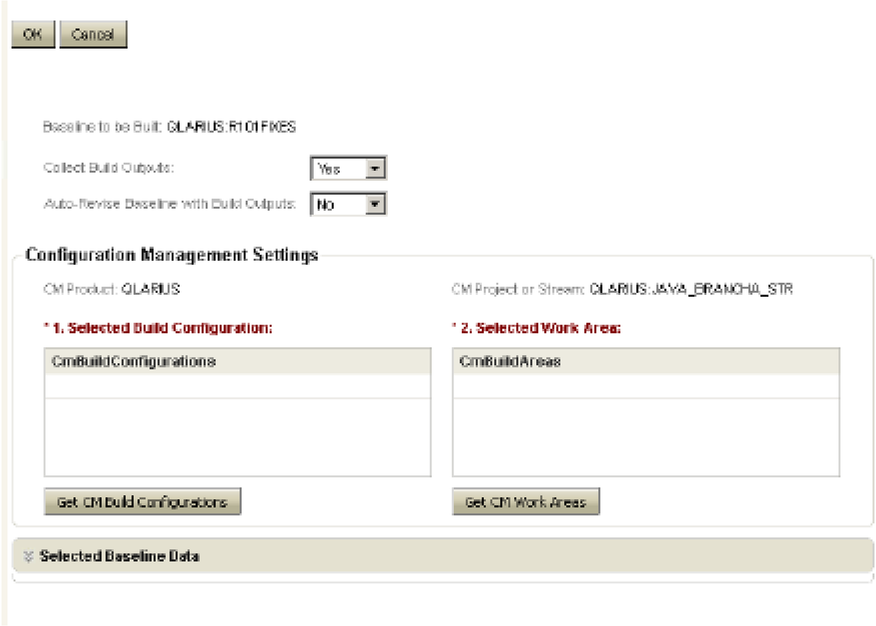
Starting a Build

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**.

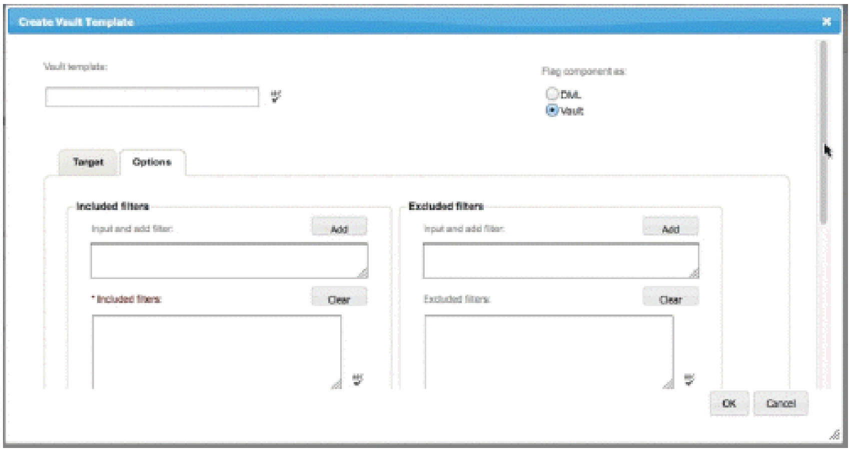
6. The package is transitioned to the **Building** state while the build task is executed by Dimensions CM. Once the build task is complete, the package is transitioned either to the **Failed Build** state (if the build failed), or the **Built** state (if the build succeeded). If the build failed, you 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.

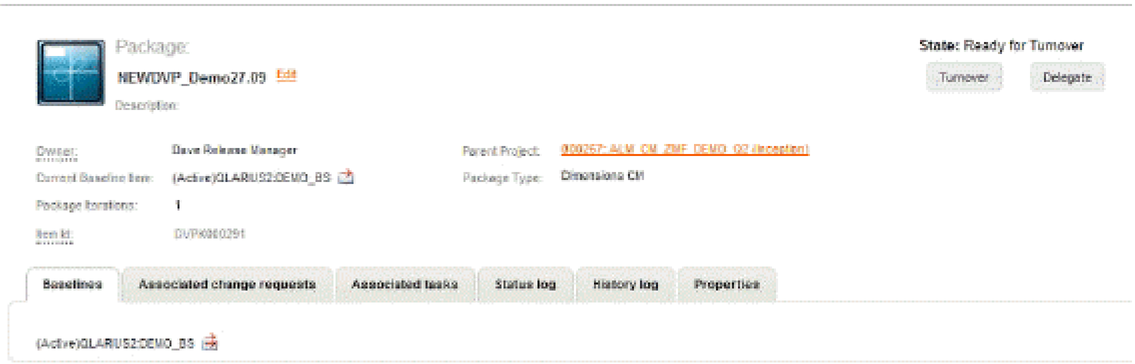
9. On the Target tab, Click the **Get DimCM Project** button and select a project from the list. This is the project to which the baseline will be deployed.

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)QLARUS2:DEMO_BS

Package Iterations: 1

Item Id: DVP#000291

Parent Project: 000267: ALM_CM_ZMF_DEMO_02 (Inception)

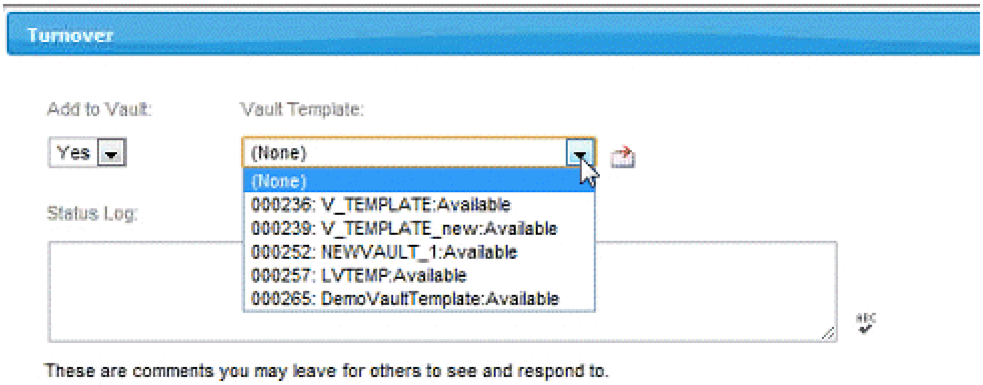
Package Type: Dimensions CM

(Active)QLARUS2: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

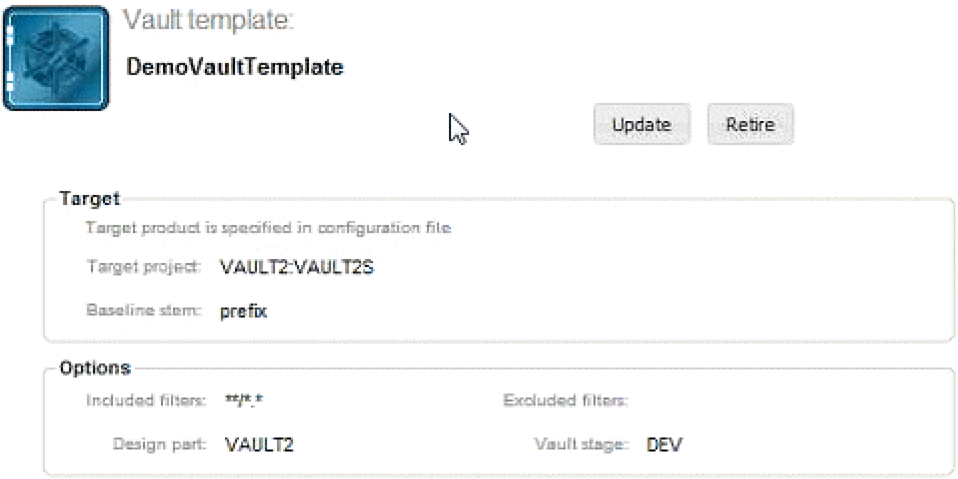
Vault Template: (None)

- (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

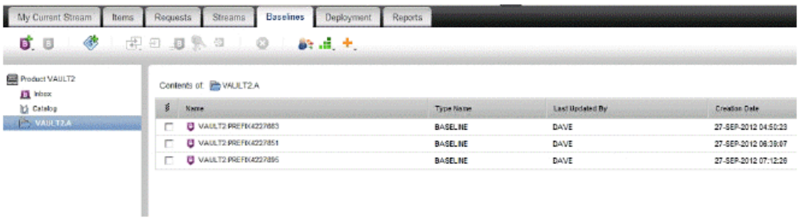
Options
Included filters: **/*.*
Excluded filters:
Design part: VAULT2
Vault stage: DEV

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.

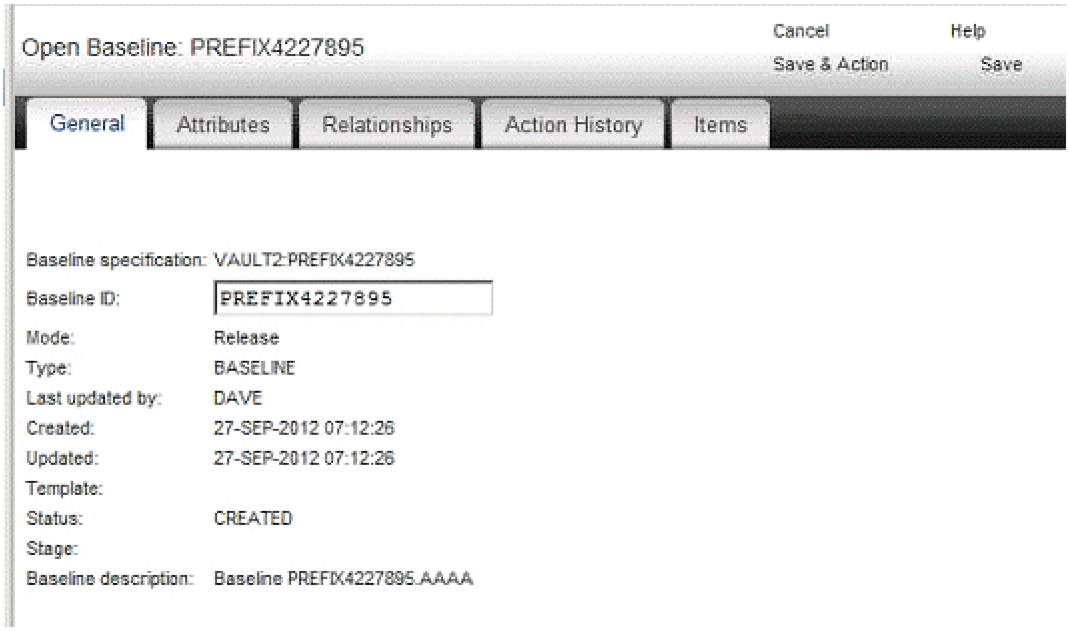
4. Click **OK** on the Turnover page. The dev package will move to **Turnover Complete**.

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



Contents of: VAULT2.A

#	Name	Type Name	Last Updated By	Creation Date
□	VAULT2:PREFIX4227883	BASELINE	DAVE	27-SEP-2012 04:50:23
□	VAULT2:PREFIX4227851	BASELINE	DAVE	27-SEP-2012 06:39:07
□	VAULT2:PREFIX4227895	BASELINE	DAVE	27-SEP-2012 07:12:26



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*.

Configuring and Using Products and Components

Products and Components

Development Manager's Products and Components process app allows you to define and manage a logical view of the component parts of your system. This helps bring control and visibility to the changes you are making to these components

Products

You manage products using the Products and Components process app. Products represent long-lived shippable software systems, which may continue to exist over the duration of multiple ALM Projects.

A product can be broken down into multiple components, and components themselves may be broken down into multiple components, or make use of other components. So a Component has a single "Owned by" parent Product, a single "Owned by" parent Component (or it is a top-level Component with no "Owned by" parent Component), and it has a set of (zero or more) "Used in" parent Components, which make use of it.

Creating a Product

1. As a component manager, click the Actions button and select New Product.

   The Create Product dialog box appears.

2. In the Name field, enter the name.

3. If you are using Product Line, find using a filter and select a product line.

4. In the Description field, enter a description.

5. From the Component Manager list, select a Component Manager.

   The Component Manager field is populated.

6. From the State list, select In Use, this allows you to create components for the product.

   Click the CM Products button and select a Dimensions CM product from the list.

   NOTE The user completing this field must be a Dimensions CM user.

7.

8. Click OK.

Editing a Product

You can edit a products properties.

1. As a component manager, either:

    Double-click on the name of the product you want to edit to view the product detail in a new window,

    or,

    Click on the name of the product you want to edit to view the product detail in the bottom half of the window.

2. Click the Edit link.

    The product information appears.

3. Edit the fields as appropriate.

4. Click OK.

5. If you have opened a new window close the window.

Components

You manage components using the Products and Components process app. Components are reusable, buildable sub-pieces of a larger software system. Components are composed together to form Products.

Components are similar to, and equate to, design parts in Dimensions CM. In fact DVM's Product can be linked to a Dimensions CM product, and DVM's Component can be linked to a CM design part (although neither link is mandatory, and it may be that you have a more coarse-grained breakdown structure in DVM than in Dimensions CM).

Each component has:

- A Component Manager, this role is for information only and does not enforce anything.
- A Component Authority, who is the component System Matter Expert (SME), the person to contact should problems arise.
- A Part Number.
- The same lifecycle as the product.

Sub-Components

Sub-components:

- Are just components that have a parent.
- Need to be created from a parent component that is in the state In Use.
- Have some fields pre-populated with default values when created.

Relationships

Sub-components are "Owned by" another component, and this relationship is made automatically when a sub-component is created from a parent component. As well as the "Owned by" relationship between two components, there is also a "Used in" relationship between two components that allows you to model the reuse of components (within a product, or between products).

Moving a component so that its "Owned by" parent will be changed must be done:

- By the Component Manager.
- By selecting the child component, pressing the Move button, and picking the new "Owned by" parent component

Editing a component so that its "Used in" parent(s) will be changed must be done:

- By the Component Manager.
- By selecting the child component, pressing the Edit link, and picking the new "Used in" parent component(s)

Components that can be the ALM Project's "Used Components", are the same set that can be the ALM Project's "Affected Components".

Reporting

You can report on the number of defects by component over time.

Creating a Component

1. As a component manager, click the Actions button and select New Component.

   The Create Component dialog box appears.

2. In the Name field, enter the name.
3. In the Component Type field, find using a filter and select a component type.
4. In the Description field, enter a description.
5. From the Component Authority list, select a component authority.
6. If you are using the Supplier field, find using a filter and select a product type.
7. In the Part Number field, either enter a part number or leave to be auto assigned.
8. The Owned by Product field, find using a filter and select a product.

   The read-only product Component Manager field is populated.

9. From the CM Design Part field, click the Get CM Design Parts button and select the design part to relate the component to.

   NOTE The user completing this field must be a Dimensions CM user.

10. Click OK.

Editing a Component

You can edit a components properties.

1. As a component manager, either:

   Double-click on the name of the component you want to edit to view the component detail in a new window,

   or,

   Click on the name of the component you want to edit to view the component detail in the bottom half of the window.

2. Click the Edit link.

   The component edit window appears.

3. Edit the fields as appropriate.
4. Click OK.
5. If you have opened a new window close the window.

Creating a Sub-Component

1. As a component manager, from the Components tab, double-click on your parent component.

   The Component dialog box appears.

2. If the parent component is still in the state Draft, click the State button Draft.

   The State is now In Use.

3. Click the Create Sub-Component button.

4. In the Name field, enter the name.

5. In the Component Type field, find using a filter and select a component type.

6. In the Description field, enter a description.

7. From the Component Authority list, select a component authority.

8. If you are using it, in the Supplier field, find using a filter and select a product type.

9. You can view the owning product information of the Owned by Product by clicking on the icon to the right of the field.

   The read-only Component Manager field is populated.

10. From the CM Design Part field, click the Get CM Design Parts button and select the design part to relate the component to.

    NOTE The user completing this field must be a Dimensions CM user.

11. From the Component Manager list, select a component manager.

12. Click OK.

Deleting a Component or Sub-Component

The Delete button is not always available.

1. As a component manager, from the Components tab, double-click on the component you want to delete.

   The Component dialog box appears.

2. Click the Delete button.

   The confirm permanent deletion window is displayed.

3. Click the Delete button.

   A window confirming the deletion is displayed.

4. Close the window.

Creating a Used-in Relationship

You create a Used-in Relationship from Sub-Component to a Component.

1. As a component manager, from the edit window of the component or sub-component that you are indicating will be used in another component, from the Used in Components list select the component it is used in.

2. Click OK.

   The Component window appears, with the Used in Components field updated.

3. Close the window.

Removing a Used-in relationship from Sub-Component to a Component

1. As a component manager, from the edit window of the component or sub-component that you are indicating is not used in another component, from the Used in Components list clear the component it is used in.

2. Click OK.

   The component window appears, with the Used in Components field updated.

3. Close the window.

Changing the Parent Component of a Sub-Component

1. As a component manager, from the edit window of the component or sub-component that you are indicating is not used in another component, from the Used in Components list clear the component it is used in.

2. Click the Move button.

   The Owned by Component window is displayed.

3. Select the new owning component.

4. Click OK.

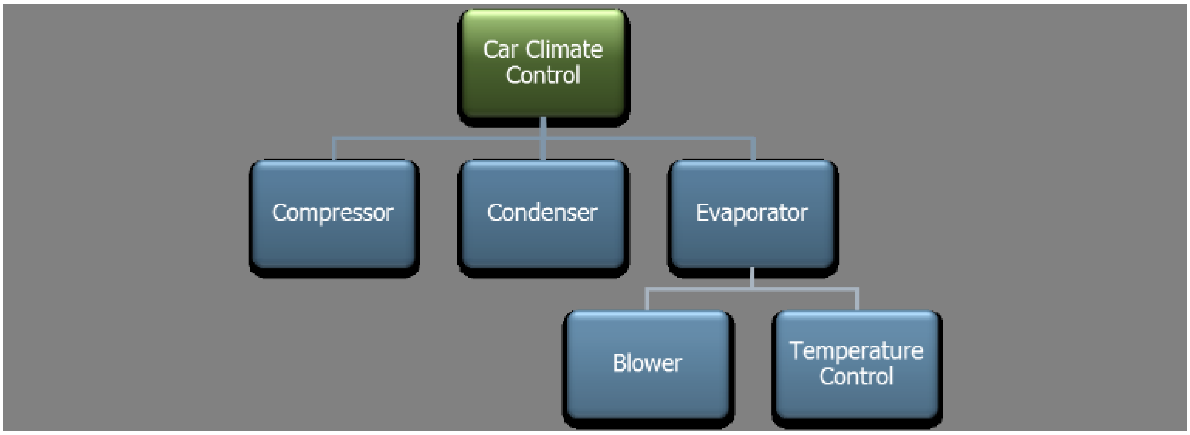
   The component window appears, with the Owned by Components field updated.

5. Close the window.

Products and Components Example

This example assumes that all the corresponding Release Vault (Dimensions CM) products and design parts have been created.

Product Car Climate Control exists in your Release Vault (Dimensions CM), the equivalent product is created using the DVM Product and Components app. Optionally, the product is associated with a product line.



The product is broken down into the following components and sub-components:

- Compressor
- Condenser

- Evaporator
- Blower
- Temperature Control

The components have a component type, for example:

- In-House
- Application
- COTS (Common off the Shelf)
- OEM
- OSS (Open Source Software)

The component states and corresponding SBM states are:

- Draft (inactive)
- In use (active)
- Deprecated (active)
- Out of Use (inactive)

Relating Products and Components in Development

Once you have modelled your product, components, and their relationships, you can relate products and components to an ALM Project to give you control and visibility of what parts of the system the project will change.

When using products and components in ALM projects you create and edit ALM projects, DCRs, and tasks as explained earlier in the document and perform somme additional steps.

If you are relating ALM projects, DCRs, and tasks to Dimensions CM design parts you must have a Dimensions CM user account.

Working with ALM Projects

An ALM project can have an Affected relationship to one product. This is the product that the Project is to cause change in.

An ALM project that has a related affected product can also affect none, one, or more components, and can be used in none, one, or more components.

The affected components are the components that the ALM project is "allowed" to change, and the used components are just informational (Components that the ALM Project expects to make use of in builds, but doesn't expect to make changes to).

Each of the affected components for an ALM project must be part of the affected product, or be a component (from another product) that is "Used In" a component from the affected product (or trace back through further "Used In" parent components back to a component from the affected product).

The same applies to the used components.

Editing an ALM Project

Typically, the ALM project manager creates an ALM project in the usual way. Once created the development manager edits the ALM project to complete the Components area fields.

1. As a development manager, from the project edit dialog box, in the Components area, for the Affected Product field, find and select the affected product.
2. Click the Get Components button.

   The Affect Components and Affect Components lists appear.
3. Select the affected components.

Displaying the Project Component Information

You can display a project’s component information.

- From the window of a project, cr, click the Components tab.

Working with Development Change Requests

A development change request (DCR) can have none, one, or more affected components.

The affected components are the components that the DCR is "allowed" to change.

Each of the affected components for a DCR must be among the components that are affected components of the DCR's parent ALM Project (or their child components as mentioned above).

You can then create DCRs scoped by components. A DCR can affect none, one, or more components.

Creating a DCR

You create a DCR in the usual way and perform these additional steps to complete fields in the Components area. You can select one or more components.

1. From the Create Dev Change Request dialog box, once you have an associated project, in the Components area, for the Affected Product field, click the Get Affected Components button.

   The Affect Components lists appear.
2. Select the affected components.

Editing a DCR

You edit a DCR in the usual way and perform these additional steps to complete fields in the Components area.

1. From the Update (request) dialog box, once you have an associated project, in the Components area, for the Affected Components field, click the Get Affected Components button.

   The Affect Components lists appear.
2. Select the affected components.

Working with Development Tasks

Once you have created DVM requests you can create DVM tasks from them to allocate units of work. A task can only affect one component. The Task Type will be Development. A Dimensions CM task is automatically created and related to the DVM task. A relationship is also created to the affected design part, if mapped.

The affected component is the component that the Development Task is "allowed" to change (if it has one).

The affected component for a development task must be among the Components that are Affected Components of the Development Task's parent DCR (or their child components as mentioned above).

Additionally, when a Development Task causes a request to be created in Dimensions CM, it will try to use the design part associated to the Development Task's Affected Component as the affected design part for the request.

Creating a Development Task

You create a development task in the usual way and perform these additional steps to complete the Affected Component field in the Components area. The associated development change request must have at least one affected component related. You can select only one component to be affected.

1. From the Create Development Task dialog box, once you have an associated development change request that has at least one affected component related, in the Components area, for the Affected Component field, click the Get Affected Components button.

   The Affected Components list appear.

2. Select one affected component.

Editing a Development Task

You edit a development task in the usual way and perform these additional steps to complete the Affected Component field in the Components area. The associated development change request must have at least one affected component. You can select only one component to be affected.

1. From the Update (task) dialog box, once you have an associated development change request that has at least one affected components, in the Components area, for the Affected Component field, click the Get Affected Components button.

   The Affected Components lists appear.

2. Select one affected component.

Using Agile Planning with Development Manager

This section consists of the following topics:

Introduction

The Agile functionality provided with Development Manager enables you to use Agile methodologies to manage your software development. This section describes the features available in an out-of-the-box implementation of Development manager. An overview of the terms and concepts involved is given below.

Roles

The roles that are specific to Agile are the Product Owner, the Scrum Master, and the Team Member.

Product Owner

The Product Owner creates and manages the Product Backlog.

The Product Owner creates User Stories from requirements based on the customers' needs. These stories are prioritized according to the business needs or ROI. This user has the privileges to create, assign, block stories, and move stories from the Product Backlog to the Release Backlog.

Scrum Master

The Scrum Master helps co-ordinate the process of addressing the tasks being worked on by the development team and dealing with any blockers that are impeding those tasks.

The Scrum Master has the privileges to create sprint backlogs, manage Releases and Sprint backlogs, create stories, move stories from the Product Backlog to Release Backlog, and to the Sprint Backlog, and to block stories. The Scrum Master creates and manages the sprint team.

Team Member

A team member carries out the tasks that are assigned to them. They are able to view the stories and tasks in the backlogs. If you have the necessary roles to update dev change requests or dev tasks, you will be able to update the details of Agile stories or tasks. You will not, however, be able to view the items in the Product Owner's or Scrum Master's landing pages, such as backlogs and sprints.

Stories and Tasks

Product features, requirements, or defects, are defined in terms of user stories. These stories have tasks allocated in order to satisfy them.

User Story

A user story describes a requirement or product feature in the form of "As a <user type> I want to <do some action> so that <desired result>". For example: "As a wiki user I want a tools menu on the edit screen so that I can easily apply font formatting".

In the out-of-the box implementation of Development manager, a story corresponds to a dev change request and follows its workflow. If you have installed the Agile features of Development Manager, there will be some additional fields on the Change Request forms. A story is estimated in story points.

Task

A task is something that needs to be carried out to implement a user story. A user story can have multiple tasks associated with it. In the out-of-the-box implementation of Development manager, a task corresponds to a dev task and follows its workflow. If you have installed the Agile features of Development Manager, there will be some additional fields on the Dev Task forms. A task is estimated in hours.

Backlogs

Product Backlog

The product backlog contains the user stories that need to be addressed for the product that have not yet been moved to the release backlog.

Release Backlog

The release backlog consists of stories that are scheduled for the next release but have not yet been moved to the sprint backlog for a specific sprint team.

Sprint Backlog

The sprint backlog is created by the scrum master in conjunction with the development team by selecting stories from the release backlog until the Development Team feels it has enough work to fill the sprint.

Sprint

A time period, for example 4 weeks, in which development occurs on a set of backlog items that the team has committed to.

Monitoring Progress

Taskboard

The Task board shows the tasks for the current sprint, indicating their status such as whether a story is in elaboration, in progress, or complete, and whether it is blocked.

Blocker

A blocker is something that is preventing a story or task from being worked on, such as a dependency on another task or hardware/software problem etc. A blocker has some text describing the problem and is visible on the Scrum Master's Landing Page.

Burndown Charts

The Sprint burn down chart is a chart showing remaining work in the sprint backlog in story points and task hours. Updated every day, it gives a simple view of the sprint progress.

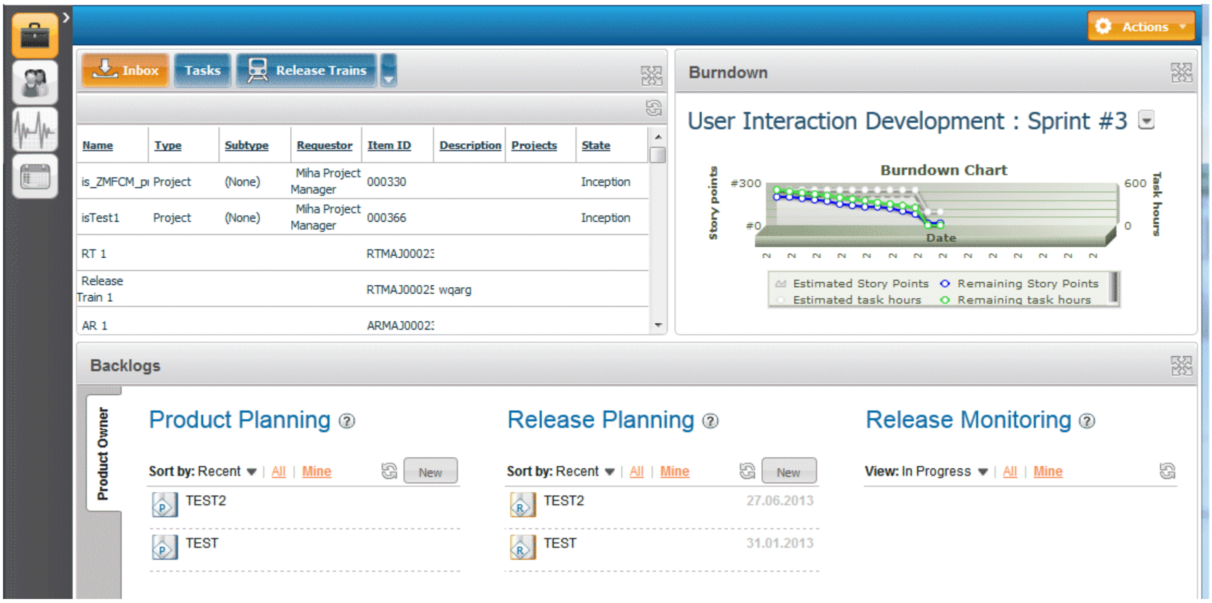
The Release Burndown shows remaining story points for the rest of the product release.

Velocity

The sprint velocity is the total effort a team is capable of in a sprint based on the number of story points completed from the last sprint's stories/features. This is a guideline for the team and assists them in understanding how many stories they can do in a sprint.

Using the Agile Landing Page

By default, if you have the Product Owner role, you will be presented on logging in with a Landing page like the one shown below.



Name	Type	Subtype	Requestor	Item ID	Description	Projects	State
is_ZMFCM_pı	Project	(None)	Miha Project Manager	000330			Inception
isTest1	Project	(None)	Miha Project Manager	000366			Inception
RT 1				RTMAJ0002З			
Release Train 1				RTMAJ00025	wqarg		
AR 1				ARMAJ0002З			

Product Planning:

- TEST2
- TEST

Release Planning:

- TEST2 27.06.2013
- TEST 31.01.2013

The upper part of this window consist of:

- A left-hand upper pane that contains your inbox and other Serena ALM applications you are using.
- A right-hand upper pane that contains a burndown chart. By default, it will display an active sprint (if one is available)

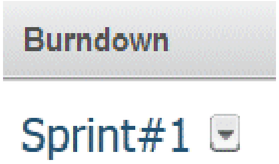
Using this page, you can:

- Display a burndown chart for a selected backlog or sprint.
- Manage backlogs using the Backlogs pane.

The Backlogs pane is different if you are a Scrum Master. For details, see The_Backlogs_pane.htm [page 59]

To choose which burndown chart is displayed:

Click the down arrow and select a sprint or release backlog.

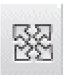


The burndown chart shows the estimated and remaining work in the selected backlog.

The lower part of this window contains the Backlogs pane.

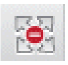
To expand any pane to fill the whole window:

Click the button at its top right.



To collapse an expanded pane:

Click the button again.



The Backlogs Pane

The lower part of the Landing page for a Product Owner or Scrum master contains the Backlogs pane.

The Product Owner's Backlog pane contains:

- Product Planning: A list of product backlogs
- Release Planning: A list of release backlogs
- Release Monitoring: A list of release backlogs according to status

For details of managing backlogs as the Product Owner, see Working With the Product Owners Backlogs Pane [page 59].

If you are a Scrum Master, the upper panes of your default landing page will be similar, but the Backlogs pane will contain:

- Release Backlogs
- Sprint Backlogs
- Blockers

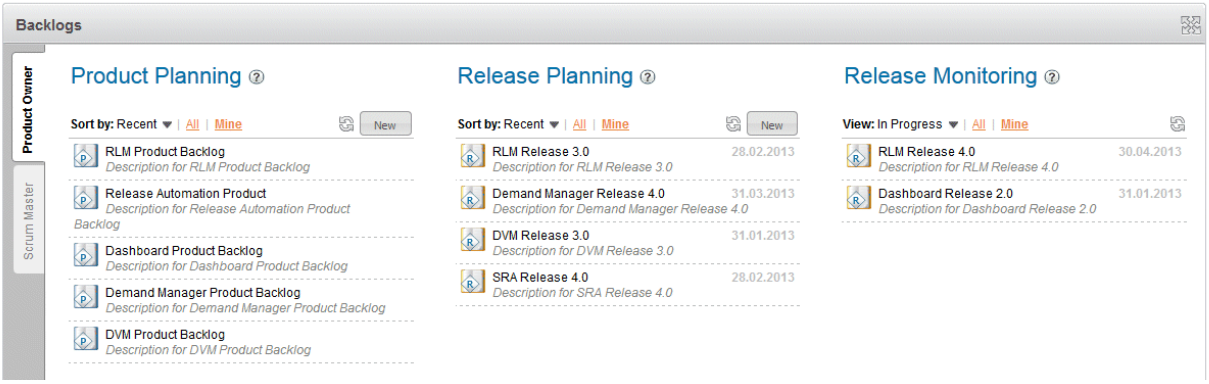
For details of managing backlogs as the Scrum Master, see Working With the Scrum Masters Backlogs Pane [page 62].

Working With the Product Owner's Backlogs Pane

The Backlogs pane allows you to manage product and release backlogs.

From this pane, the Product Owner can:

- List backlogs for Product Planning and Release Planning
- Add a new product or release backlog
- Delete a product backlog
- Select a product or release backlog to view or edit its details



Product Planning

RLM Product Backlog
Description for RLM Product Backlog

Release Automation Product
Description for Release Automation Product Backlog

Dashboard Product Backlog
Description for Dashboard Product Backlog

Demand Manager Product Backlog
Description for Demand Manager Product Backlog

DVM Product Backlog
Description for DVM Product Backlog

Release Planning

RLM Release 3.0 28.02.2013
Description for RLM Release 3.0

Demand Manager Release 4.0 31.03.2013
Description for Demand Manager Release 4.0

DVM Release 3.0 31.01.2013
Description for DVM Release 3.0

SRA Release 4.0 28.02.2013
Description for SRA Release 4.0

Release Monitoring

RLM Release 4.0 30.04.2013
Description for RLM Release 4.0

Dashboard Release 2.0 31.01.2013
Description for Dashboard Release 2.0

To sort the Product or Release backlogs:

Click the arrow in the heading and select:

- **Recent** to sort the backlogs in descending order of the last time you accessed them
- **Name** to sort in the backlogs in alphabetical order

To filter the Release Monitoring backlogs:

Click the arrow in the heading and select:

- **Planning**

- **In Progress**
- **Completed**

To filter the list of Product or Release backlogs or Release Monitoring backlogs:

Select:

- **All** to view all backlogs
- **Mine** to view backlogs created by you

To create a new product backlog:

1. Under Product Planning, click the **New** Button.
2. On the New Backlog page, enter the **Product Name** and optionally, a **Description**.
3. Click **Process**.

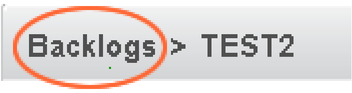
To create a new release backlog:

1. Under Release Planning, click the **New** Button.
2. On the New Backlog page, enter the **Release Name** and select the **State**.
3. Optionally, enter a **Description**, a **Start Date**, and an **End Date**.
4. Click **Process**.

To View the page for a backlog:

Click the row for the backlog. This will display the backlog in the Backlogs pane.

You can return to the backlogs by clicking **Backlogs** in the breadcrumbs.

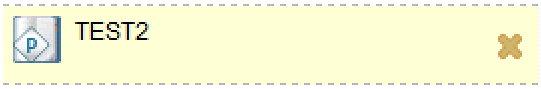


To Edit a backlog:

Click the row for the backlog, then click the **Edit** link. for details, see Managing a Backlog [page 60]

To Delete a product backlog:

1. Hover over the row and click the delete button on the right.



2. Reply **Yes** in confirmation dialog.

Managing a Backlog

The Backlog page enables you to manage a selected Product or release backlog.

Use the backlog page to:

- Edit the details of a backlog
- Add or edit a story
- Change the rank of a story
- Move stories into the backlog
- Display the burndown chart for a release backlog

To edit a product backlog:

Click the **Edit** link. You can change:

- Product Name
- Description

To edit a release backlog:

Click the **Edit** link. You can change:

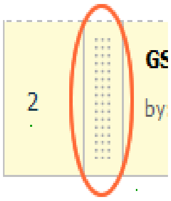
- Release Name
- Description
- Start Date
- End Date
- State: Planning, In Progress, or Completed

To show the burndown chart for a release backlog:

Click the **Show Burndown** link at the top right. This will display a burndown chart in the backlog page.

To change the rank of a story:

Drag on the area next to the rank number and move it up or down to the required position.



To create a new story:

Click the **New Story** button

> This displays a Create Dev Change Request page that is similar to that described in Creating New Change Requests [page 35], except that it has a header containing a **Story Points** field.

1. Select a value for Story Points.
2. Enter the required values for the dev change request.

To view or edit a story:

1. Click its name link.

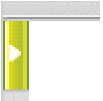
   The Change Request page is displayed to enable you to edit its details.

The Stories Panel

Use the Stories panel to move stories into any product, release or sprint backlog to which you have permissions to edit that is currently selected in the Backlog page. You can move stories by selecting them in the Stories panel and draging them into the backlog.

To show or hide the Stories panel:

To show the Stories panel, click the arrow to the left of the Backlog pane:

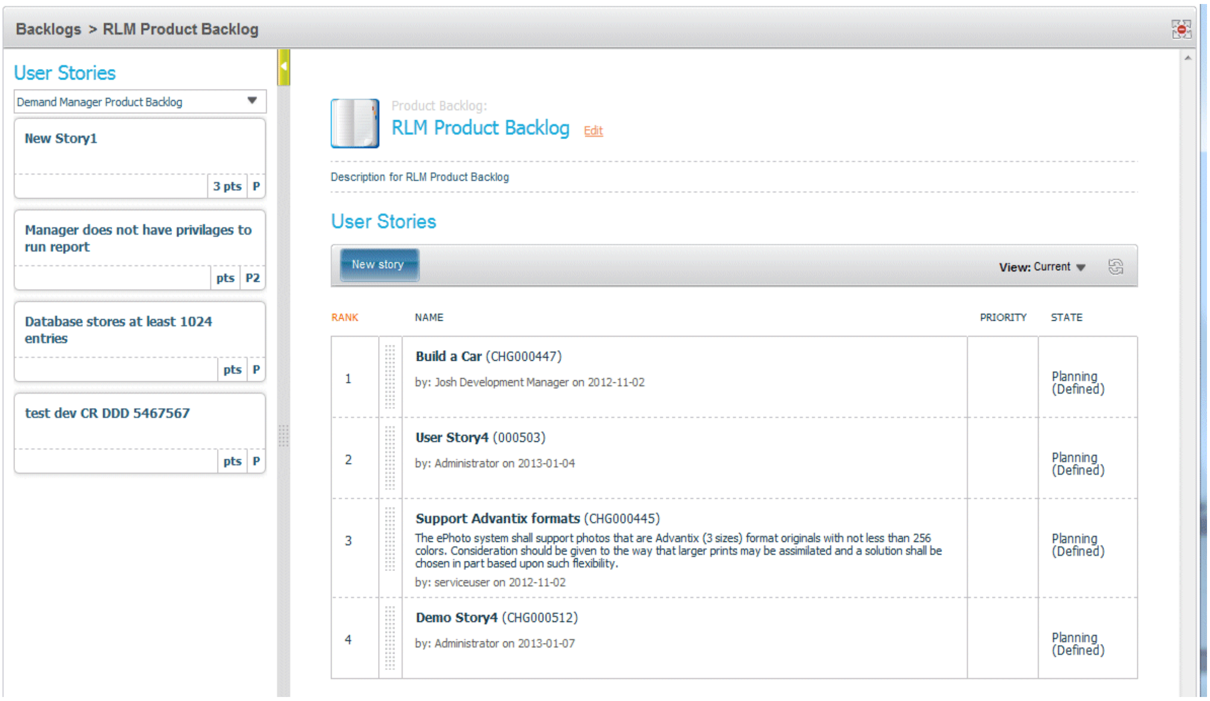


To hide it, click the arrow again.

To display stories in the Stories panel:

Open the drop-down list and select:

- A backlog or
- A category for stories that are not in a backlog.



Backlogs > RLM Product Backlog

User Stories

Demand Manager Product Backlog

New Story1 — 3 pts — P

Manager does not have privilages to run report — pts — P2

Database stores at least 1024 entries — pts — P

test dev CR DDD 5467567 — pts — P

Product Backlog:

RLM Product Backlog Edit

Description for RLM Product Backlog

User Stories

RANK	NAME	PRIORITY	STATE
1	**Build a Car** (CHG000447) by: Josh Development Manager on 2012-11-02		Planning (Defined)
2	**User Story4** (000503) by: Administrator on 2013-01-04		Planning (Defined)
3	**Support Advantix formats** (CHG000445) The ePhoto system shall support photos that are Advantix (3 sizes) format originals with not less than 256 colors. Consideration should be given to the way that larger prints may be assimilated and a solution shall be chosen in part based upon such flexibility. by: serviceuser on 2012-11-02		Planning (Defined)
4	**Demo Story4** (CHG000512) by: Administrator on 2013-01-07		Planning (Defined)

To move a story from the stories panel into the selected backlog:

Select one or more stories and drag them to the required position in the backlog.

Note that you cannot drag stories back into the Stories panel.

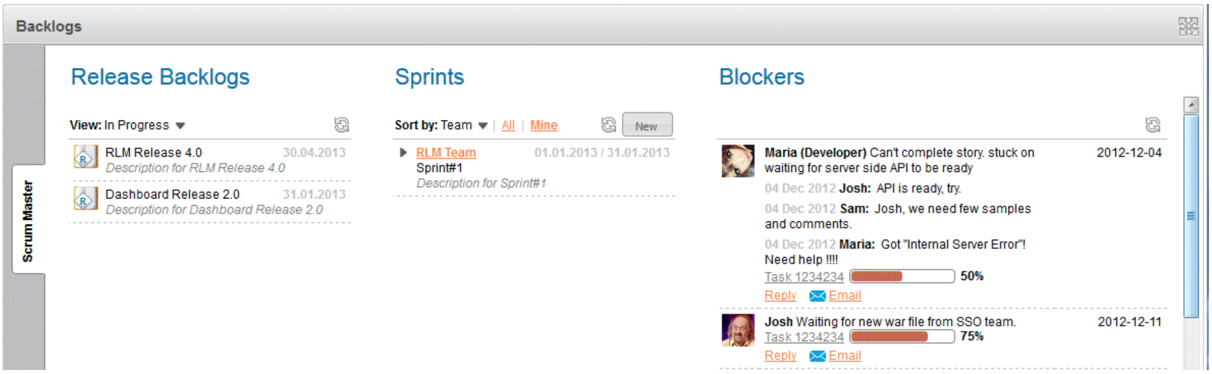
Working With the Scrum Master's Backlogs Pane

The Scrum Master's Backlogs pane allows you to manage release and sprint backlogs.

From this pane, the Scrum Master can:

- List backlogs for Releases and Sprints
- Add a new sprint
- Select a release or sprint backlog to view or edit its details
- View any current blockers for stories or tasks

If you are a Scrum Master, your default landing page will be similar to that of the Product Owner, but the Backlogs pane will look like the example below.



Backlogs

Scrum Master

Release Backlogs

View: In Progress

RLM Release 4.0 30.04.2013
Description for RLM Release 4.0

Dashboard Release 2.0 31.01.2013
Description for Dashboard Release 2.0

Sprints

Sort by: Team | All | Mine

New

RLM Team 01.01.2013 / 31.01.2013
Sprint#1
Description for Sprint#1

Blockers

Maria (Developer) Can't complete story. stuck on waiting for server side API to be ready 2012-12-04

04 Dec 2012 **Josh:** API is ready, try.

04 Dec 2012 **Sam:** Josh, we need few samples and comments.

04 Dec 2012 **Maria:** Got "Internal Server Error"! Need help !!!!

Task 1234234 50%

Reply Email

Josh Waiting for new war file from SSO team. 2012-12-11

Task 1234234 75%

Reply Email

It will contain:

- Release Backlogs for which you are the Scrum Master
- Sprint Backlogs
- Blockers for backlogs for which you are the Scrum Master

You cannot create or edit a release backlog but you can create and edit its stories, and move stories to it.

To sort the release backlogs:

Click the arrow in the heading and select:

- **Planning**
- **In Progress**
- **Completed**

To filter the sprint backlogs:

Select:

- **All** to view all backlogs
- **Mine** to view backlogs created by you

To sort the sprint backlogs:

Click the arrow in the heading and select:

- **Recent** to sort the sprints in descending order of the last time you accessed them
- **Name** to sort in the sprints in alphabetical order
- **Team** to sort in order of team

To expand the list of sprints for a team:

Click the arrow ▶ next to the team name.

To collapse it, click the arrow again.

To create a new sprint:

1. Under Release Planning, click the **New** Button.
2. On the New Backlog page, enter the **Sprint Name** and select a **Team**.
3. Optionally, enter a **Description**, a **Start Date**, and an **End Date**.
4. Click **Process**.

To edit the details of a team:

1. Click the name link for the team in the Sprints list
2. Enter the **Team Name** and select a **Scrum Master**.
3. Optionally, enter the **Velocity** in story points and/or hours, and enter a **Description**.
4. Click **Process**.

To reply to a blocked story with a comment:

Click the blocked story's reply link and enter some text. Your comment is added to the blocker details.

To view a blocked story in a new window:

Click the blocked story's name link.

To View the page for a sprint:

Click the row for the sprint. This will display the sprint items in the Backlogs pane. For details, see Managing a Sprint. [page 64]

You can return to the backlogs by clicking **Backlogs** in the breadcrumb link in the header of the Backlogs pane.

Managing a Sprint

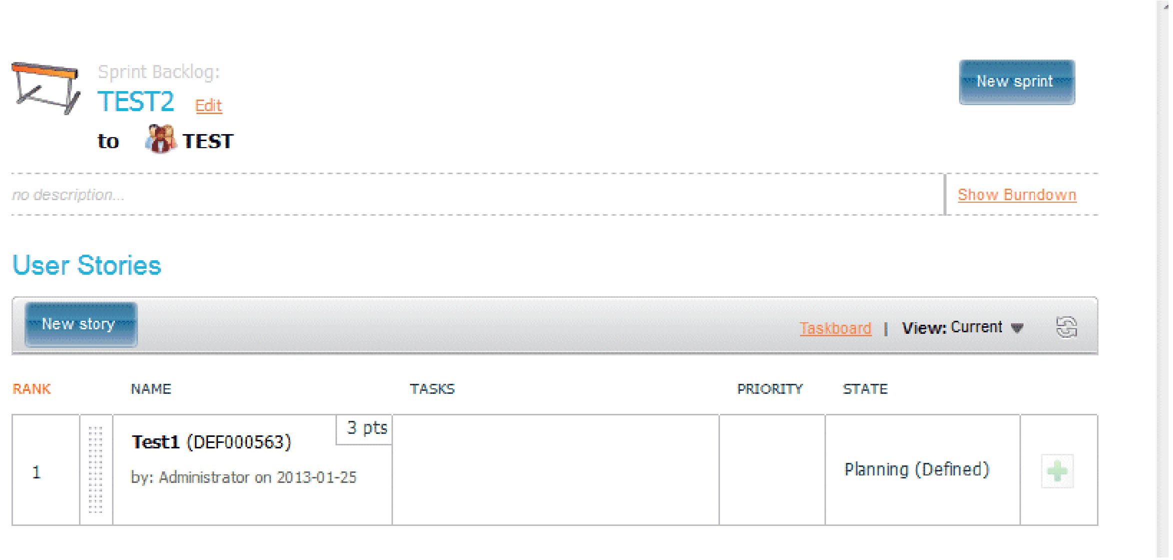
The backlog page for a sprint shows stories and their associated tasks.

On this page you can:

- Edit the details of a sprint
- Create a new sprint
- Add or edit a story
- Add or edit a task
- Change the rank of a story
- Move stories into the backlog
- Display the burndown chart for a sprint

You can choose between two views of the sprint items:

- **Backlog**: a view of the backlog of stories
- **Taskboard**: a view of the tasks listed by status.



Sprint Backlog:
TEST2
to TEST

no description...

User Stories

RANK	NAME	TASKS	PRIORITY	STATE
1	Test1 (DEF000563) 3 pts by: Administrator on 2013-01-25			Planning (Defined)

To edit the details of the sprint:

Click the **Edit** link. You can change:

- Sprint Name
- Description
- Start Date
- End Date

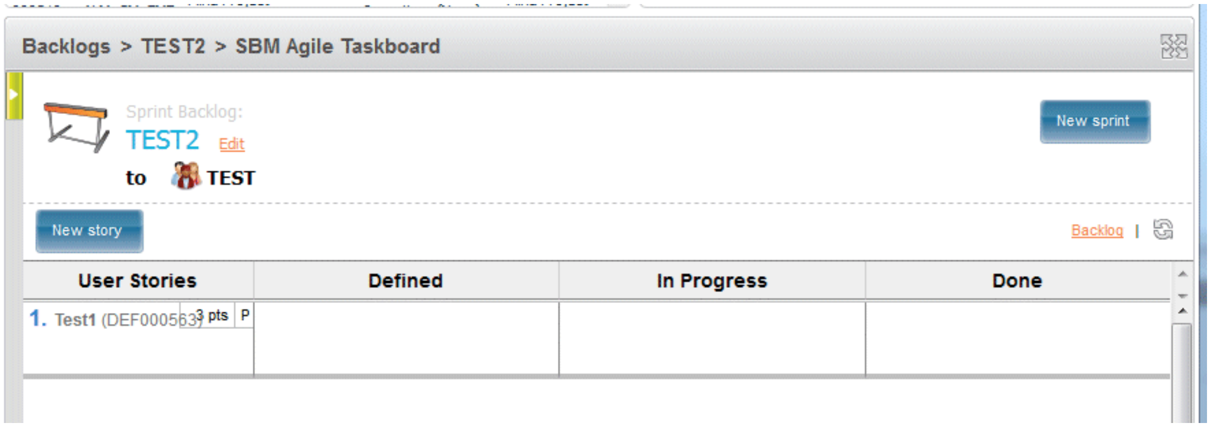
To show the burndown chart for this sprint:

Click the **Show Burndown** link at the top right.

To view the taskboard:

Click the **Taskboard** link in the heading.

This will show tasks listed in columns according to their status.



To switch to the backlog view:

On the taskboard, click the **Backlog** link in the heading.

To create a new sprint:

Click the **New sprint** button.

To create a new story:

Click the **New story** button.

To edit a story:

Click the name link for the story.

To create a new task:

1. Click the name link for the story.
2. Select the **Tasks** tab.
3. Click the **New Task** button.

   This page is similar to the Create Dev Task page described in Creating New Tasks [page 37], except that there is a header containing a **Priority** field.

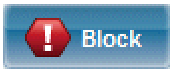
4. On the Create Dev Task page, select a **Priority**. If you want to show the task as blocked, click the Block button and enter some text for the description.

To edit a task:

Click on the name link.

To block a story or a task:

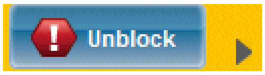
1. On the Change Request page or the Dev Task page, click the **Block** button.



2. Enter some text for the description.
3. Click **OK**.

To unblock a story or a task:

Click the **Unblock** button.



To view the details of a blocked story:

Click the arrow to the right of the **Unblock** button. To add a comment, click the **Reply** link and enter some text.

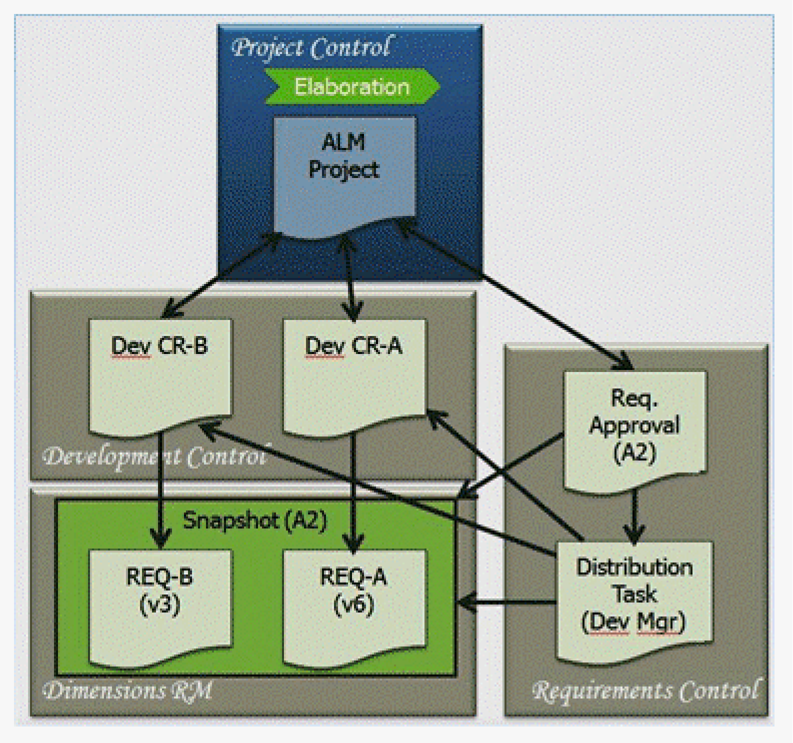
Using the Requirements Manager Integration

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 [page 68].
- 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 [page 34].
- 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 [page 76].
- View the requirements that are included in a selected dev package. See Viewing the Requirements for a Dev Package [page 77].

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.

Related Topics

Viewing a Distribution Task [page 69]

Creating a Single Dev Change Request from One or More Requirements [page 70]

Creating one or More Dev Change Requests From Requirements [page 30]

Viewing a Distribution Task

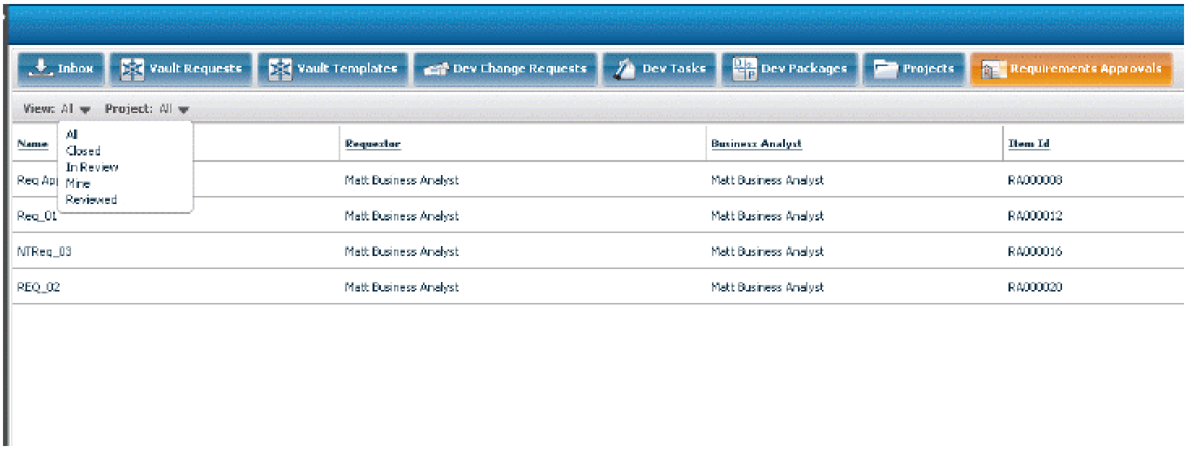
You select a distribution task by first selecting the requirement approval to which it belongs.

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.

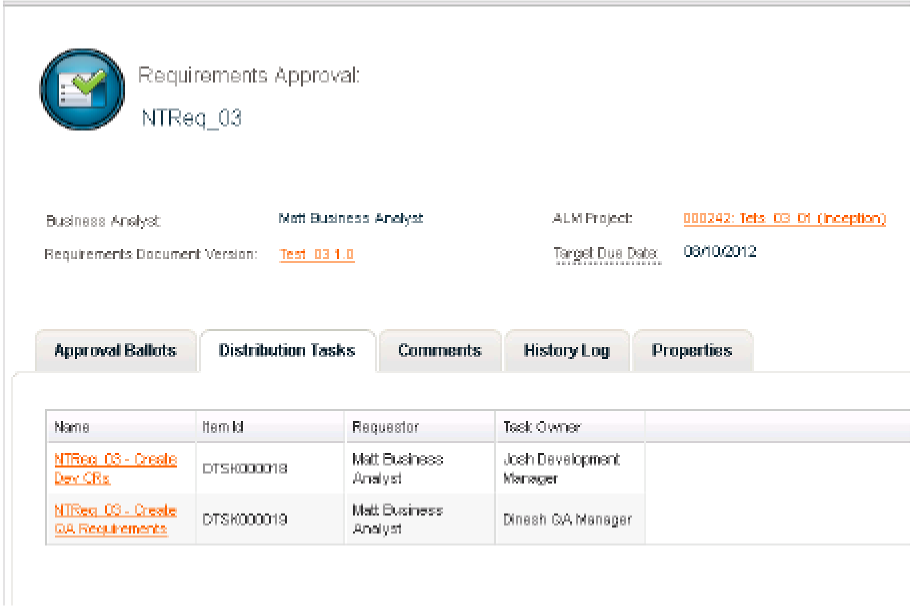
By clicking the arrow for the **Project** option, you can select dev change requests that are associated to a specific project.



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:

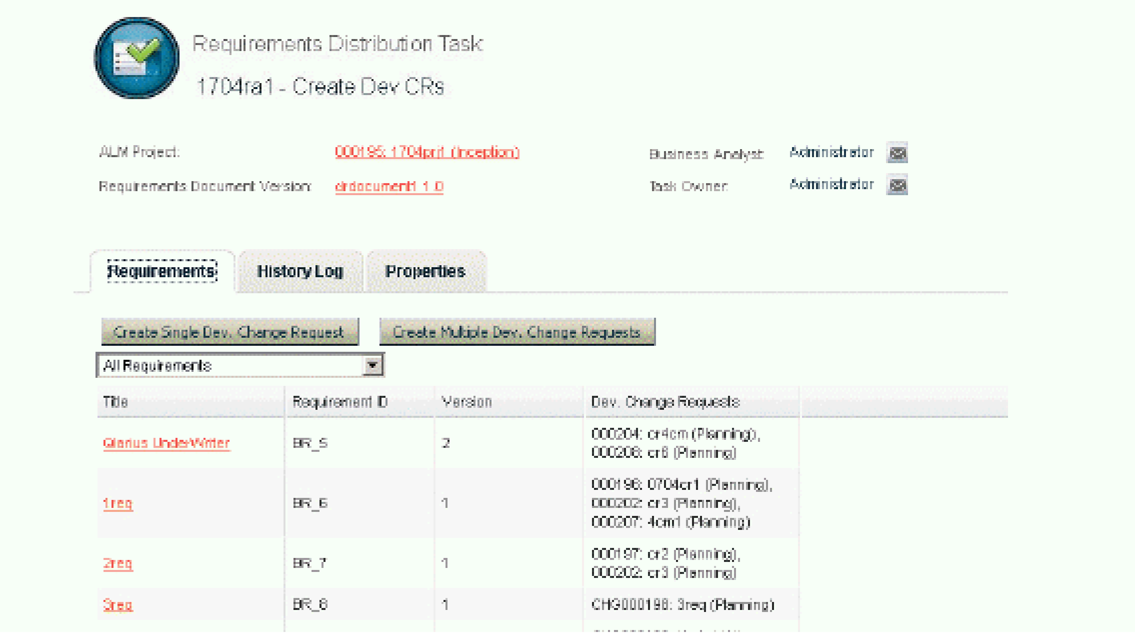
1. click on the corresponding row.


Requirements Approval:
NTReq_03

Business Analyst: Matt Business Analyst
ALM Project: 000242: Tets_03_01 (Inception)
Requirements Document Version: Test_03.1.0
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 Distribution Task

1704ra1 - Create Dev CRs

ALM Project: 000195: 1704prj (Inception)

Requirements Document Version: drdocument1 1.0

Business Analyst: Administrator

Task Owner: Administrator

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

Title	Requirement ID	Version	Dev. Change Requests
Glanus UnderWriter	BR_5	2	000204: cr4cm (Planning), 000208: cr8 (Planning)
1req	BR_6	1	000196: 0704cr1 (Planning), 000202: cr3 (Planning), 000207: 4cm1 (Planning)
2req	BR_7	1	000197: cr2 (Planning), 000202: cr3 (Planning)
3req	BR_8	1	CHG000198: 3req (Planning)

4. Select the **Requirements** tab

You can select:

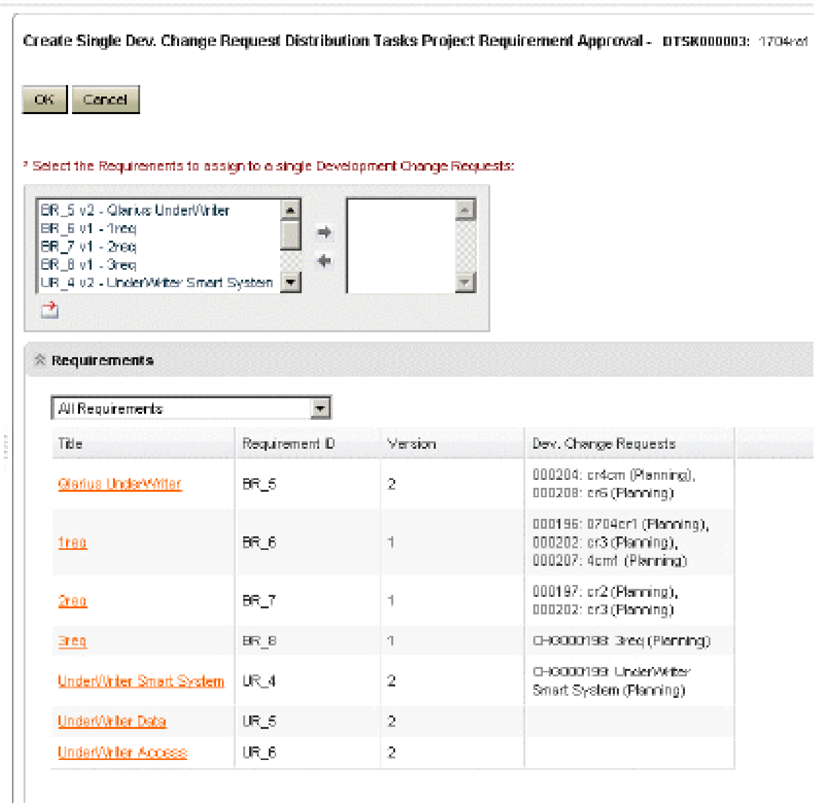
- **All Requirements** to view all the requirements belonging to the distribution task
- **With Development Change Requests** to view only requirements that have a dev change request assigned to them.
- **Without Development Change Requests** to view only requirements that do not have a dev change request 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: 1704rel

OK Cancel

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

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

Requirements

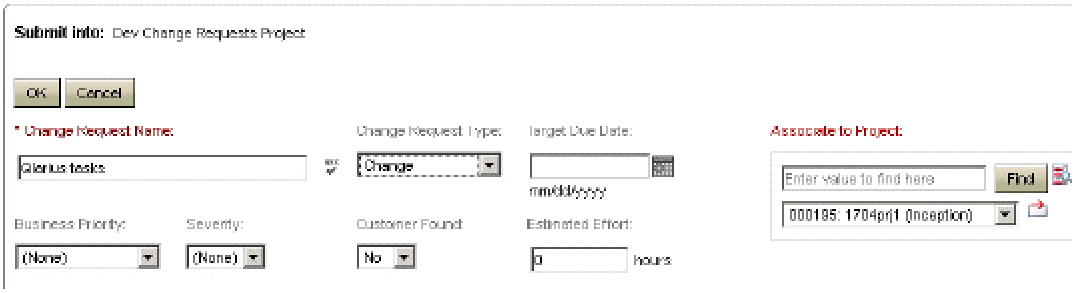
All Requirements

Title	Requirement ID	Version	Dev. Change Requests
Glarius UnderWriter	BR_5	2	000204: cr4cm (Planning), 000208: cr6 (Planning)
1req	BR_6	1	000196: 0704cr1 (Planning), 000202: cr3 (Planning), 000207: 4cm1 (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	
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 info: Dev Change Requests Project

OK Cancel

* Change Request Name: Glarius tasks

Change Request Type: Change

Target Due Date: mm/dd/yyyy

Associate to Project: Enter value to find here Find

000195: 1704prj1 (Inception)

Business Priority: (None)

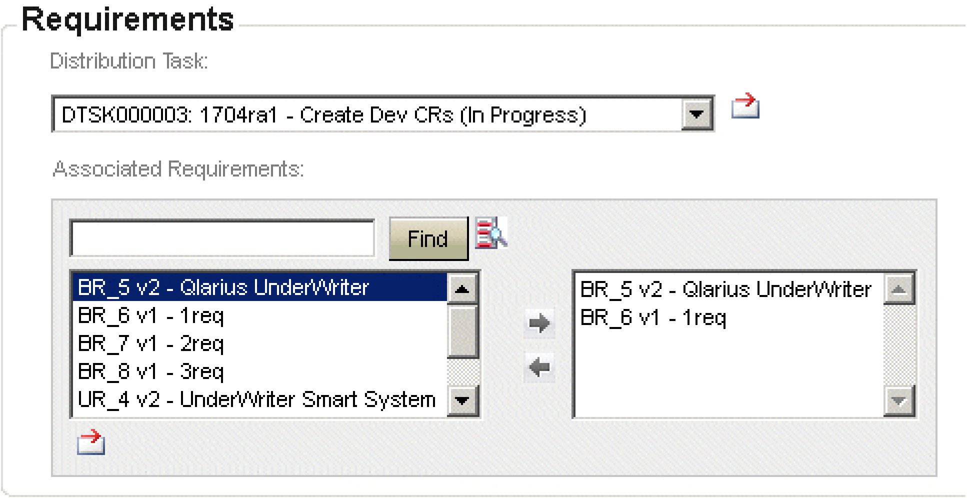
Severity: (None)

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.

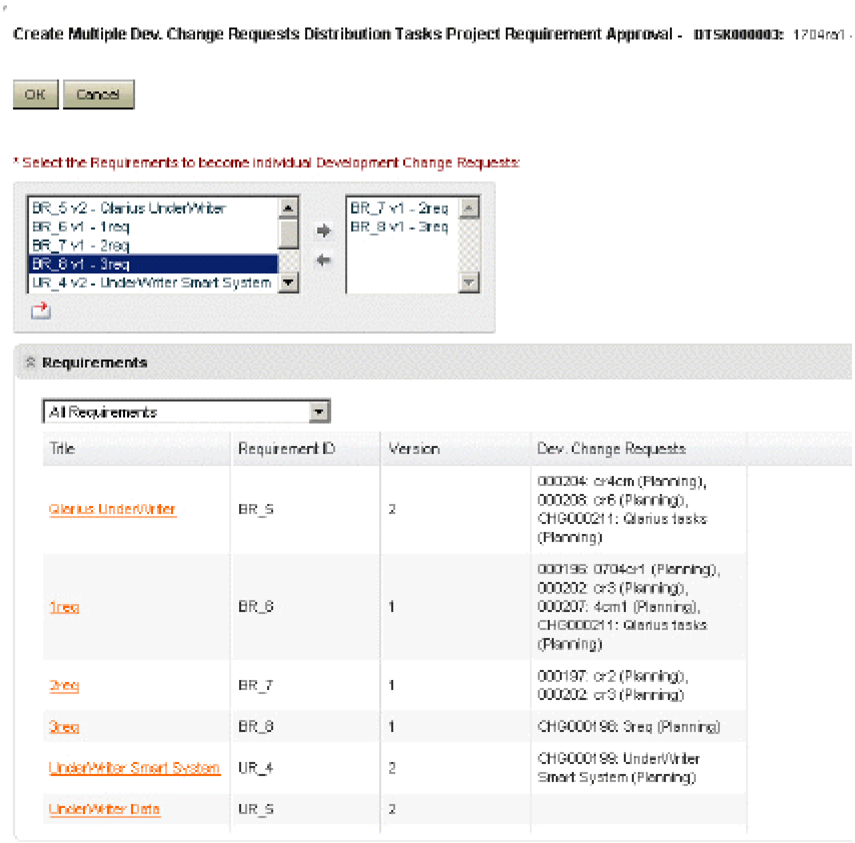

Requirements
Distribution Task:
DTSK000003: 1704ra1 - Create Dev CRs (In Progress)
Associated Requirements:
Find
BR_5 v2 - Qlarius UnderWriter
BR_6 v1 - 1req
BR_7 v1 - 2req
BR_8 v1 - 3req
UR_4 v2 - UnderWriter Smart System
BR_5 v2 - Qlarius UnderWriter
BR_6 v1 - 1req

8. Click **OK**.

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

Selected:

- BR_7 v1 - 2req
- BR_8 v1 - 3req

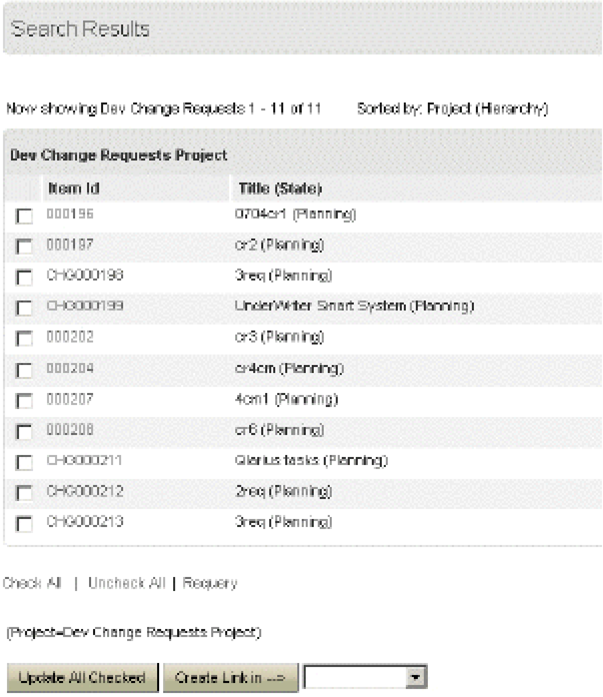
Requirements

All Requirements

Title	Requirement ID	Version	Dev. Change Requests
Clarius UnderWriter	BR_5	2	000204: cr4cm (Planning), 000208: cr6 (Planning), CHG000211: Clarius tasks (Planning)
1req	BR_6	1	000196: 0704cr1 (Planning), 000202: cr3 (Planning), 000207: 4cm1 (Planning), CHG000211: Clarius 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	Glarius tasks (Planning)
□	CHG000212	2req (Planning)
□	CHG000213	3req (Planning)

(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.

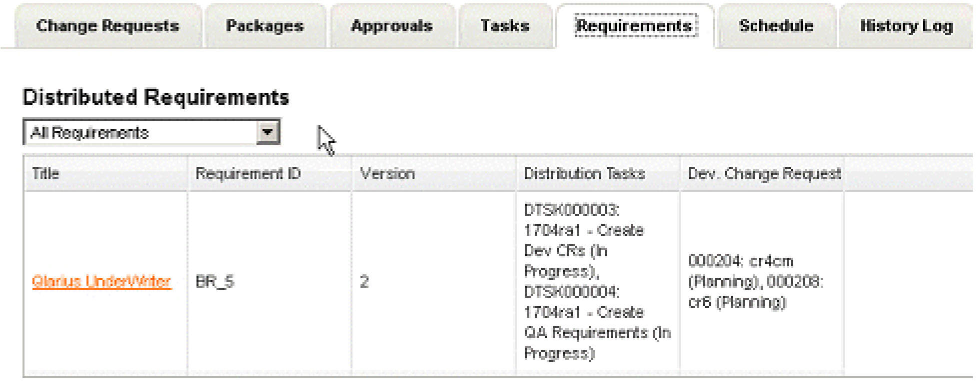
Related Topics

Creating a Single Dev Change Request from One or More Requirements [page 70]

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)

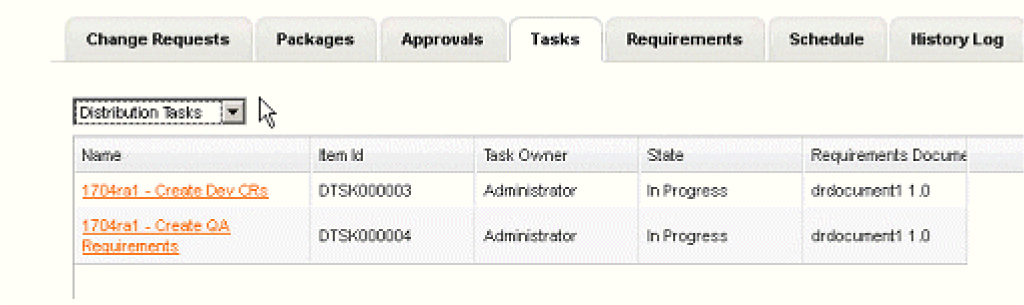
To view the requirements:

- 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



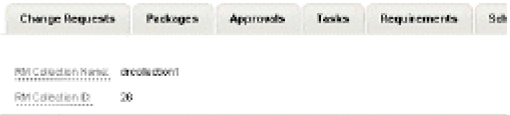
Distribution Tasks

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:

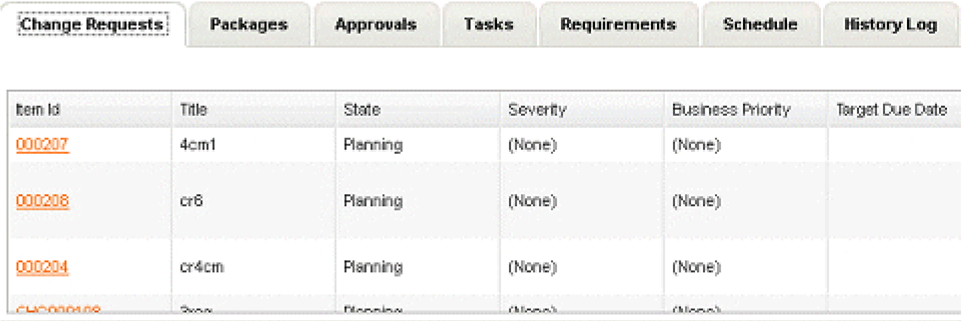
Click the Dimensions RM tab.



RM Collection Name: drcollection1
RM Collection ID: 28

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

Click the Change Requests tab.

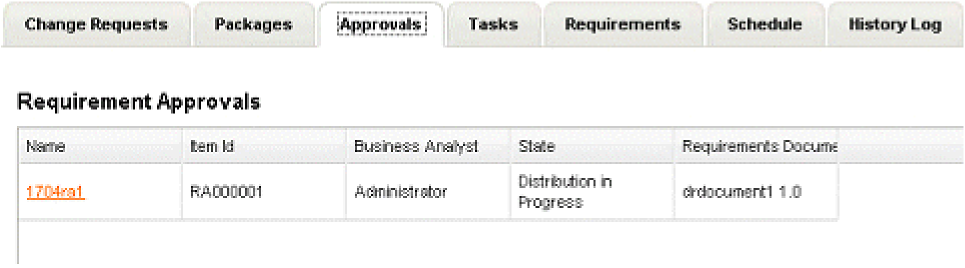


Item Id	Title	State	Severity	Business Priority	Target Due Date
000207	4cm1	Planning	(None)	(None)	
000208	cr6	Planning	(None)	(None)	
000204	cr4cm	Planning	(None)	(None)	

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.

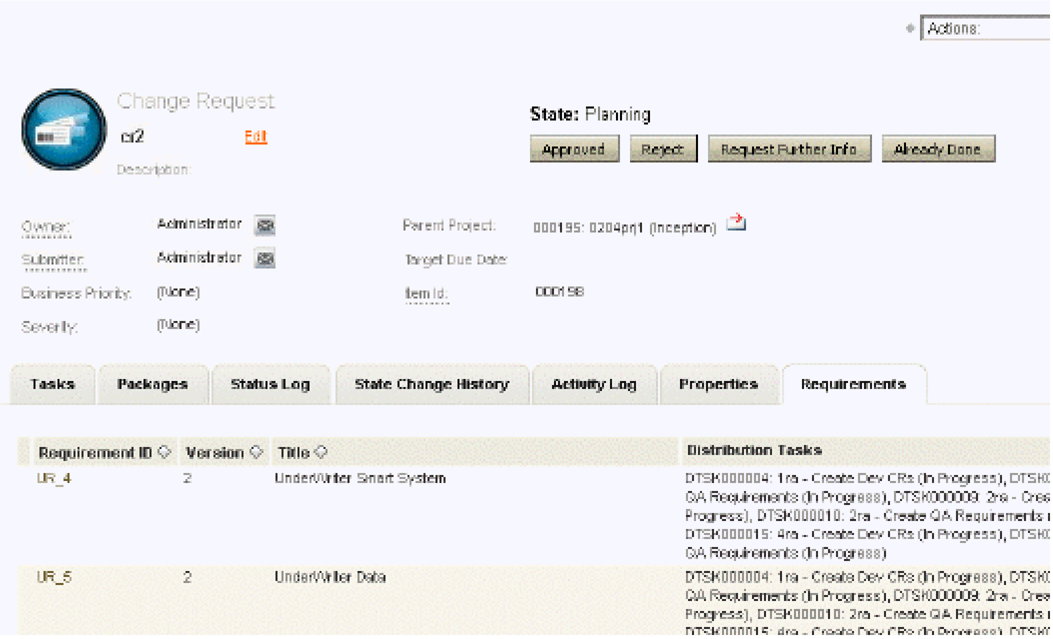
Viewing and Associating Requirements for a Dev Change Request

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

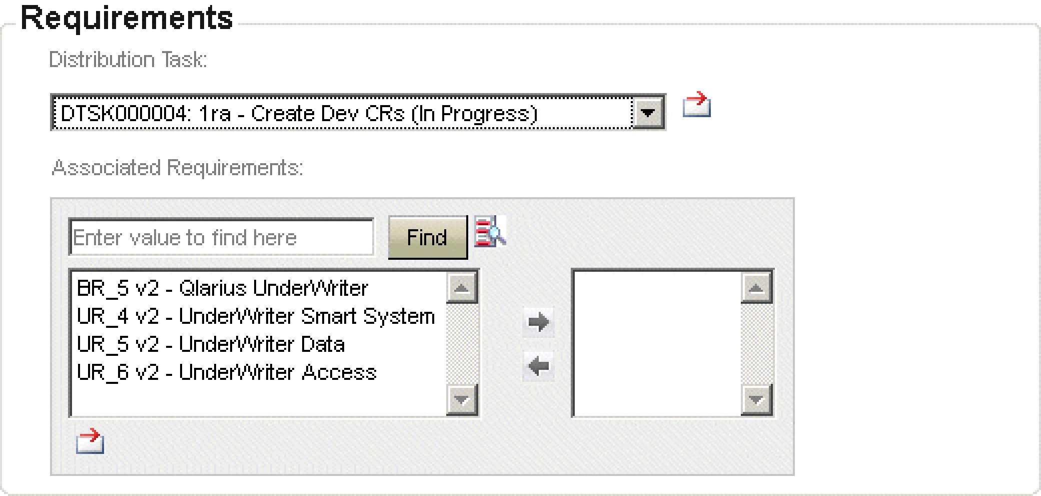
Requirement ID	Version	Title	Distribution Tasks
UR_4	2	UnderWriter Smart System	DTSK000004: 1ra - Create Dev CRs (In Progress), DTSK(QA Requirements (In Progress), DTSK000009: 2ra - Crea Progress), DTSK000010: 2ra - Create QA Requirements (DTSK000015: 4ra - Create Dev CRs (In Progress), DTSK(QA Requirements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

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

Find

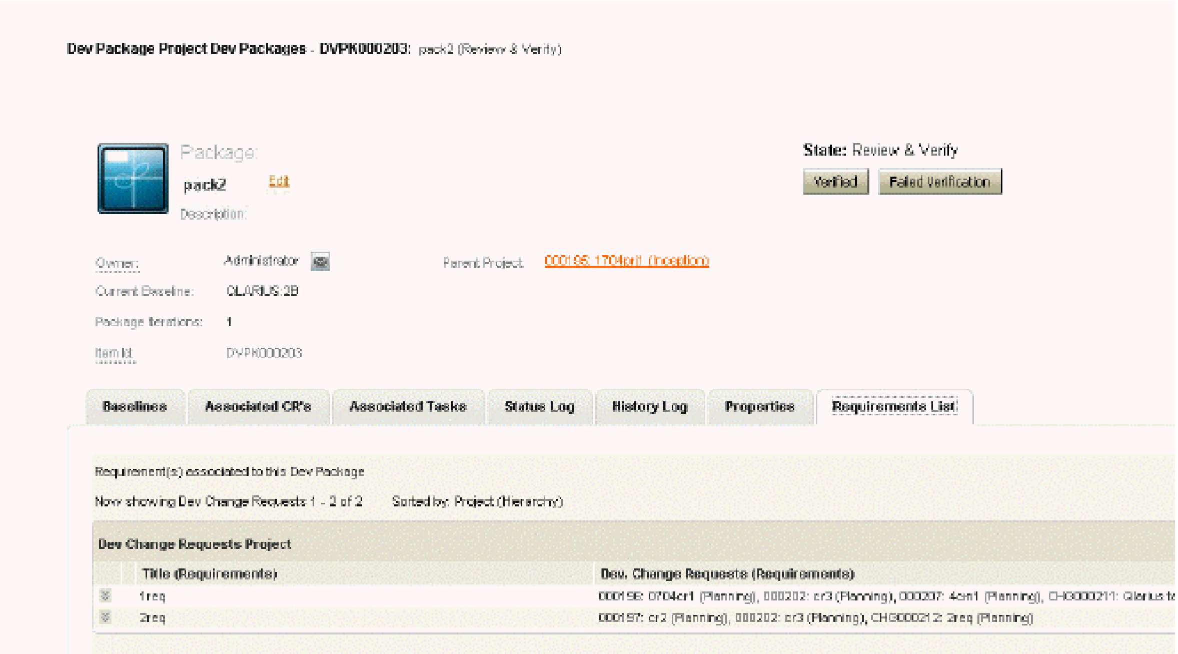
BR_5 v2 - Qlarius UnderWriter
UR_4 v2 - UnderWriter Smart System
UR_5 v2 - UnderWriter Data
UR_6 v2 - UnderWriter Access

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 Edit
Description:

State: Review & Verify

Verified | Failed Verification

Owner: Administrator

Parent Project: 000196: 1704prj1 (Inception)

Current Baseline: QLARIUS:2B

Package Iterations: 1

Item Id: DVPK000203

Baselines | Associated CR's | Associated Tasks | Status Log | History Log | Properties | Requirements List

Requirement(s) associated to this Dev Package

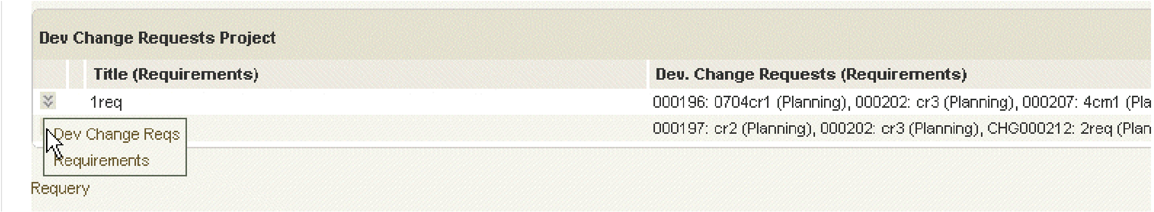
Now showing Dev Change Requests 1 - 2 of 2   Sorted by: Project (Hierarchy)

Dev Change Requests Project

	Title (Requirements)	Dev. Change Requests (Requirements)
︾	1req	000196: 0704cr1 (Planning), 000202: cr3 (Planning), 000207: 4cm1 (Planning), CHG000211: Qlarius te
︾	2req	000197: cr2 (Planning), 000202: cr3 (Planning), CHG000212: 2req (Planning)

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

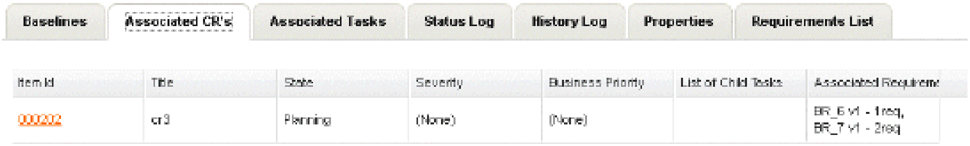
- **Requirements** to view the requirement.
- **Dev Change Reqs** to view the dev change requests.



Dev Change Requests Project

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

- 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_7 v1 - 2req