U.S. Department of Education
Federal Student Aid

START HERE
GO FURTHER
FEDERAL STUDENT AID®

Federal Student Aid Enterprise Configuration Management User Guide

Version 01.03

01/24/2011

Document Version Control

Version	Date	Description
01.00	07/08/2010	Initial Release
01.01	08/13/2010	Incorporated RequisitePro baseline creation verbiage and the new report “Change Requests and Defect Summary”
01.02	09/30/2010	Added verbiage that describes the policies related to associating a CR with a Defect.
01.03	01/24/2011	Updated RequisitePro access/security matrix and requirement attribute matrix.

Table of Contents

Section 1. Introduction 1
1.1 Purpose 1
1.2 Scope 1
1.3 Intended Audience 1
1.4 Document Organization 1
1.5 References and Related Documents 2
Section 2. Rational Environment Configuration 3
2.1 Configuration of the Rational Environment 3
2.1.1 Requisite Pro Tailoring and Configuration 3
2.1.2 ClearQuest Project Configuration and Tailoring 4
2.1.3 ClearCase Configuration and Tailoring 5
2.1.4 Rational Quality Manager Tailoring and Configuration 6
2.2 Rational Tool Integration 10
2.3 Security 10
Section 3. Configuration Control 12
3.1 Change Control 12
3.1.1 ClearQuest Design Overview 12
3.1.2 Working with Change Requests 18
3.1.3 Email Notification 26
3.2 Version Control 28
3.2.1 Project Documentation VOB 30
3.2.2 Project VOB (PVOB) and Component VOBs 30
3.2.3 Stream Management 31
Section 4. Tool Integrations 37
4.1 ClearQuest to RequisitePro 37
4.2 ClearQuest to ClearCase 38
4.3 ClearQuest to Rational Quality Manager 39
Section 5. Baseline Management 41
5.1 Baseline Naming Conventions 41
5.2 Creating Baselines 41
5.2.1 Creating a base-ClearCase Label to Represent an FSA Baseline (Functional, Design, or Product) 42
5.2.2 Creating a ClearCase UCM-Baseline to Represent an FSA Baseline (Functional, Design, or Product) 42
5.3 Including Artifacts Managed in Other Tools 43
5.3.1 Including RequisitePro Artifacts in a Baseline 43
5.3.2 Including Rational Quality Manager Artifacts in a Baseline 43
5.4 Archival and Management 43
Section 6. Reporting 44
6.1 ClearQuest Queries 44
6.1.1 General Queries 44
6.1.2 Report Queries (Filters) 44
6.2 ClearQuest Reports 45
6.2.1 CR Reports 45

6.2.2 ChildCR Report Formats 46
Appendix A: Acronyms and Abbreviations A-1
Appendix B: Glossary B-1
Appendix C: ClearQuest Record Details C-1
Appendix D: ClearQuest Report Samples D-1
Appendix E: Security E-1
Appendix F: Job Aids F-1

List of Figures

Figure 2-1: Test Plan 7
Figure 2-2: Test Plan Categories 7
Figure 2-3: RQM Test Case 8
Figure 2-4: RQM Test Case Categories 9
Figure 3-1: Related ClearQuest Record Types 12
Figure 3-2: CR Record Workflow 14
Figure 3-3: ChildCR Record Workflow 15
Figure 3-4: Defect Record Workflow 16
Figure 3-5: Defect Resolution Flowchart 17
Figure 3-6: Opening a New Change Request 20
Figure 3-7: CR Submit Form – Attachments Tab 22
Figure 3-8: Attachment-Related Fields 22
Figure 3-9: ChildCR Submit – Main Tab 23
Figure 3-10: ChildCR Submit – Development Tab 24
Figure 3-11: Submitting a Defect from RQM 26
Figure 3-12: Stream Structure for a Single Release 33
Figure 3-13: Stream Structure for a New Release 34
Figure 3-14: Cross Project Delivery 35
Figure 4-1: Change Request – Requirements Tab 37
Figure 4-2: RequisitePro Requirements Linked to ClearQuest Records 38
Figure 4-3: CR Change Set 39
Figure 4-4: Defect Created from RQM 40

List of Tables

Table 1-1: Intended Audience and Uses 1
Table 2-1: Rational Tools Requiring Configuration 3
Table 2-2: Tailorable RequisitePro Attributes 4
Table 2-3: Dynamic Name Lists 5
Table 2-4: Stateless Record Types 5
Table 2-5: ClearCase Project Configuration Tasks 6
Table 2-6: RQM Test Case Categories 10
Table 2-7: Rational Operations Reserved for the CM Lead 11
Table 3-1: Table of CR States in the Standard CR Lifecycle 19
Table 3-2: CR Submit Form – Main Tab 21
Table 3-3: ChildCR Step Action Table 25

Table 3-4: Defect Step-Action Table 25
Table 3-5: CR Email Notification Rules 27
Table 3-6: ChildCR Email Notification Rules 28
Table 3-7: Defect Email Notification Rules 28
Table 3-8: VOBs Used in a System Development Effort 29
Table 3-9: Recommended VOBs and Directories 31

Section 1. Introduction

Automation and enforcement of software Configuration Management (CM) helps to ensure that project teams adhere to established CM processes and procedures. This, in turn, helps to ensure the integrity of the system throughout its development, implementation, and maintenance.

A suite of tools have been tailored and integrated in order to aid in the automation and enforcement of FSA Configuration Management related processes and procedures.

1.1 Purpose

The purpose of the FSA Enterprise Configuration Management User Guide is to provide guidance in the utilization of the FSA Enterprise Solution built on the IBM Rational tool suite to support software configuration management. It is the responsibility of the CM Lead to ensure that the guidelines, rules, and procedures defined in this plan are adhered to by all system support team members.

1.2 Scope

The information presented in this user guide is limited to the usage of the Rational tools. All users are expected to have prior hands-on experience using RequisitePro, Rational Quality Manager, ClearCase and ClearQuest.

1.3 Intended Audience

Table 1-1 lists the intended users and the purpose for which the users may utilize information in this document.

Users	Uses
Configuration Management Staff	As a guide to implementing standard configuration management practices using the Rational suite of tools.

Table 1-1: Intended Audience and Uses

1.4 Document Organization

This document is comprised of the following sections:
Section 1 – Introduction
Section 2 – Rational Environment Configuration
Section 3 – Configuration Control
Section 4 – Tool Integration

Section 5 – Baseline Management
Section 6 – Reporting
Appendix A – Acronyms and Abbreviations
Appendix B – Glossary
Appendix C – ClearQuest Record Details
Appendix D – Report Samples
Appendix E – Security
Appendix F – Job Aids

1.5 References and Related Documents

The following documents were referenced during the development of this requirements management user guide:

- ACS Directive OCIO:1-106, Lifecycle Management Framework
- FSA Enterprise Configuration Management Plan Template
- FSA Enterprise Requirements Management Plan Template
- FSA Enterprise Requirements Management User Guide
- FSA Enterprise Test Management Standard
- FSA Enterprise Test Management User Guide
- IBM Rational ClearCase Command Reference
- IBM Rational ClearCase Information Center (Web Based ClearCase Guidance): https://publib.boulder.ibm.com/infocenter/cchelp/v7r1m0

Section 2. Rational Environment Configuration

At the start of a system development effort, a system-specific instance of the FSA Enterprise Solution will need to be created and configured within the FSA Rational Environment. The FSA Rational Support Group (FSA RSG) will create the instance of the environment for the new system and the configuration management (CM) Lead will configure each tool in the environment based on system-specific requirements.

Table 2-1 lists the primary Rational tools used in every system development effort as well as the type of system-specific configuration performed by the CM Lead.

Rational Tool	Description of System-Specific Configuration
RequisitePro	No system-specific tailoring is supported.
ClearQuest	Choice list values will be updated for CR, ChildCR, and Defect Record types.
Rational Quality Manager	Test Plan and Test Case category choice lists will be tailored to system-specific requirements.
ClearCase	Directories will need to be created in the system documentation Version Object Base (VOB) and software VOBs. Over the life of the development effort, base-ClearCase labels and Unified Change Management (UCM) baselines will need to be created.

Table 2-1: Rational Tools Requiring Configuration

Unless otherwise indicated, the Rational tool related functions described in this document can be performed using any of the interfaces (native-client, thin-client, and web-client) provided by the tools.

NOTE: When printing from the web interface for any of the Rational tools, the tool-specific print function should be used rather than using the browser print function directly.

2.1 Configuration of the Rational Environment

2.1.1 Requisite Pro Tailoring and Configuration

Prior to the start of a system development effort, a system-specific RequisitePro project will be created by the FSA RSG. Values for the requirements attributes listed in Table 2-2 should be provided to the RSG team in order to tailor RequisitePro project to the system-specific needs.

NOTE: The maximum length for each list value is 32 characters.

Attribute Name	Description	Type	Choice List or Format
Product Release	The latest release that this requirement applies to. ***Defined during project configuration and updated throughout the life of the project.***	List	##.##.###
Requirements Baseline	The latest requirements baseline that modified this requirement. ***Defined during project configuration and updated throughout the life of the project.***	List	FB_##.##.###
Component	Component list for the system under development. ***Defined during project configuration and updated throughout the life of the project.***	List	Any text string

Table 2-2: Tailorable RequisitePro Attributes

RequisitePro cannot be tailored by anyone other than the FSA RSG. In the event that the system development team wishes to tailor RequisitePro, a formal change request should be submitted via the FSA Rational Environment Request System (FRERS).

2.1.2 ClearQuest Project Configuration and Tailoring

The FSA Enterprise Solution of ClearQuest has been tailored to allow two levels of tailoring of choice lists for the various record types (CR and ChildCR). When a system-specific instance of ClearQuest is first deployed, and throughout the life of the project, the CM Lead will be responsible for maintaining these choice lists. Each choice list has been implemented as either a ClearQuest Dynamic Name List or as a Reference List to a stateless record type. Table 2-3 describes Dynamic Name Lists.

Dynamic Name List	Used by Record Types	Form/Field
CR-Category	CR	Main/Category
CR-CancellationReason	CR	CCB/Cancellation Reason
CR-Complexity	CR	Impact Analysis/Complexity
CR-OnHoldReason	CR	CR/On Hold Reason
Defect-Type	Defect	Main/Defect Type
Defect-Phase	Defect	Main/Detected in Phase
Defect-CancellationReason	Defect	“Cancellation/Rejection”/Cancellation Reason

Table 2-3: Dynamic Name Lists

Table 2-4 lists all of the stateless records that will need to be created and maintained by the CM Lead during the life of a system development effort. In addition to creating instances of the stateless records, the CM lead will need to "mark" the records as either "Active" or "Inactive".

Stateless Record Type	Used by Record Types	Form/Field
Release	CR	Main/Found In Release Main/Assigned Release
	Defect	Target Release
Component	CR	Development/Components
	ChildCR	Development/Components

Table 2-4: Stateless Record Types

Release Records: Used to specify the release that the CR was found in and the targeted release for the CR. A release record is either active or inactive. An active release is one that is currently being developed or scheduled for development. An inactive release is one that is currently in production or that has, in the past, been in production.

- When a system release goes into production, the corresponding ClearQuest "Release" record should be updated to indicate that it is inactive.
- When a new system release is identified, a new ClearQuest "Release" record should be created and made active.

Component Records: Each CR and ChildCR will be linked to one or more system components.

- When a new system component is identified, a new ClearQuest component record should be created and made active.
- When existing system components become obsolete, the corresponding ClearQuest component record should be made inactive.

2.1.3 ClearCase Configuration and Tailoring

The FSA Enterprise Solution of ClearCase has been configured to allow the project and CM Team to manage the elements in the VOBs, the stream, delivery and rebase operations, and the creation of label types, labels, and baselines. The recommended initial set of VOBs and their sub directories are elaborated in Section 3.2. If the project requires additional policy enforcement mechanisms via ClearCase triggers, a FRERS change request should be submitted by the FSA Point of Contact or Technical Point of contact. A summary of initial and ongoing ClearCase related project configuration tasks are listed in 2-5.

Task	Who Performs Task	When Task is Performed
Create Initial UCM Project for the first major release being worked on.	FSA Rational Support Group	At the start of each new major release of the project. A FRERS ticket should be created in order to request that a new UCM Project be created.
Create release-specific integration stream.	System CM Lead	At the start of each new major release of the project.
Create shared development stream.	System CM Lead	At the start of each new major release of the project.
Create baselines.	System CM Lead	Any time.
Create and manage directory structure in all VOBs.	System CM Team	At the start of each new system development effort. The directory structure will most likely change over the life of the project.
Manage which users have permission to deliver to the release-specific integration stream.	System CM Lead	At the start of each new project. Initially, only the CM team will have permission to checkout/in/deliver on the release-specific integration stream. However, the CM Lead may permit other users to deliver to the project integration stream. See Appendix F-Job Aids for a description of how to restrict/allow write access to the release-specific integration stream.
Maintain list of users that are permitted to deliver to the project integration stream. This is typically the CM Lead and perhaps an alternate.	System CM Lead	At the start of each new project. Initially, only the CM team will have permission to checkout/in/deliver on the project integration stream. However, the CM Lead may permit other users to deliver to the project integration stream from the release-specific integration stream. See appendix F-Job Aids for a description of how to restrict/allow write access to the project integration stream.

Table 2-5: ClearCase Project Configuration Tasks

2.1.4 Rational Quality Manager Tailoring and Configuration

The CM Lead may update choice lists for both Test Plans and Test Cases.

Test Plan: Summary Section

Each test plan can be categorized based on the release of the system being tested and the phase of testing being performed. Therefore, the category choice lists for “Release” and “Testing Phase” will need to be populated.

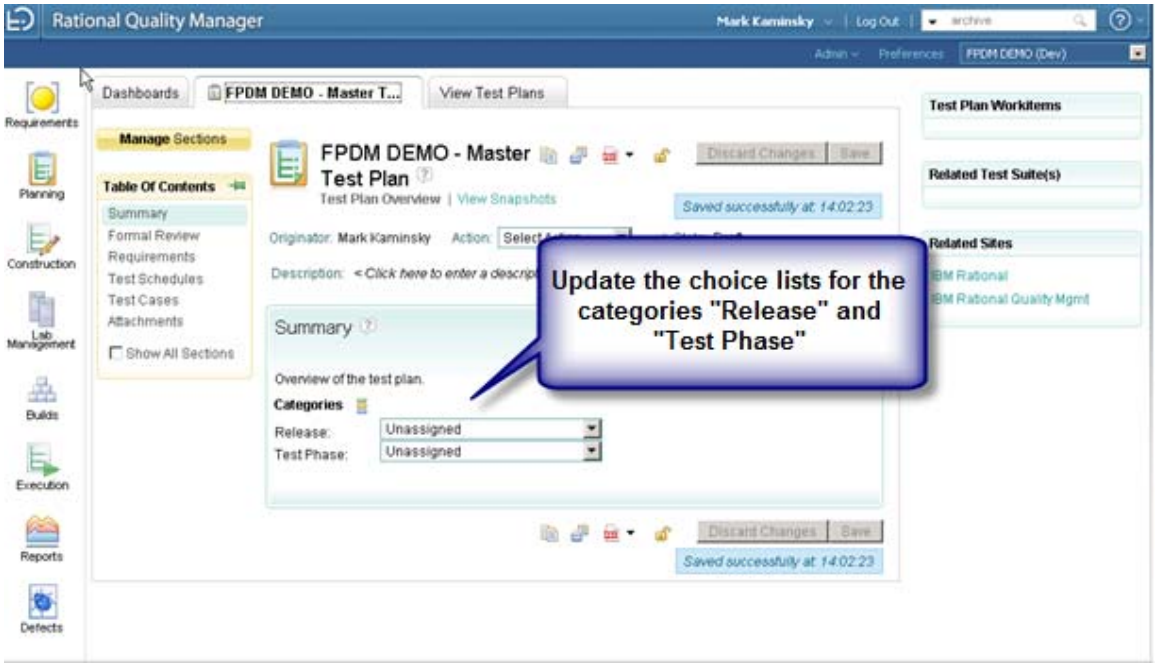


Figure 2-1: Test Plan

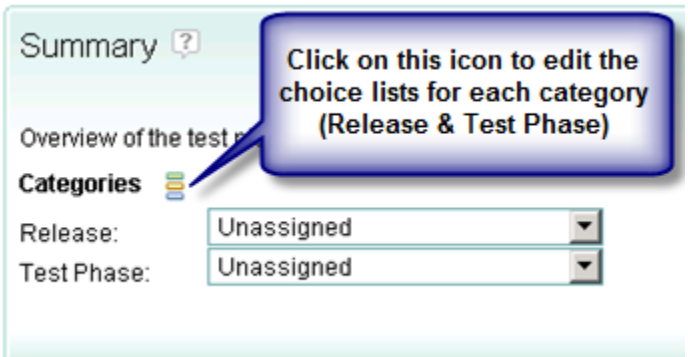


Figure 2-2: Test Plan Categories

Test Plan: Summary Section

Each test case can be categorized based on the types of categories in the table below.

Category Name	Category Description	Initial Values (values preloaded)
Category	Generic list that can be tailored based on the system needs.	To be loaded by the CM Lead at project startup.
Function	A means of categorizing the test case based on the function being tested.	To be loaded by the CM Lead at project startup.

Category Name	Category Description	Initial Values (values preloaded)
Test Phase	The test phase to which the test case applies.	System Testing User Acceptance Testing

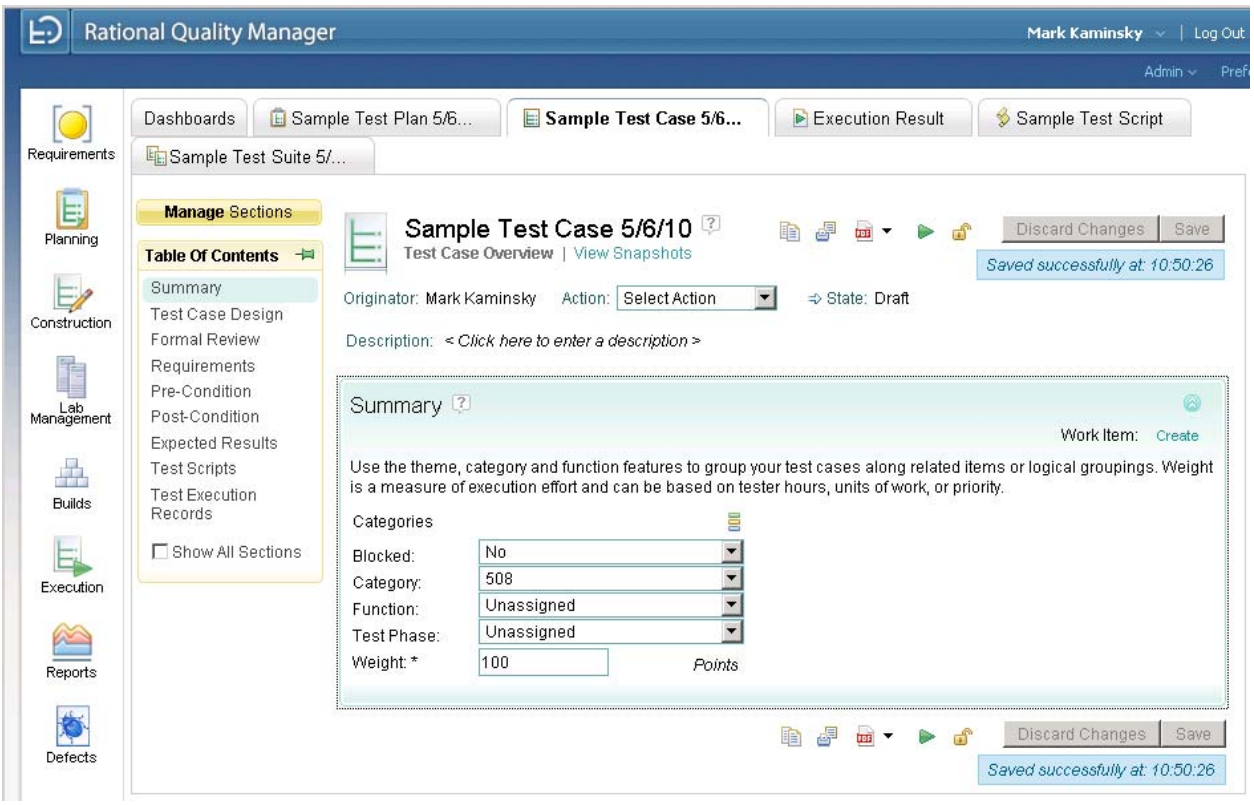


Figure 2-3: RQM Test Case

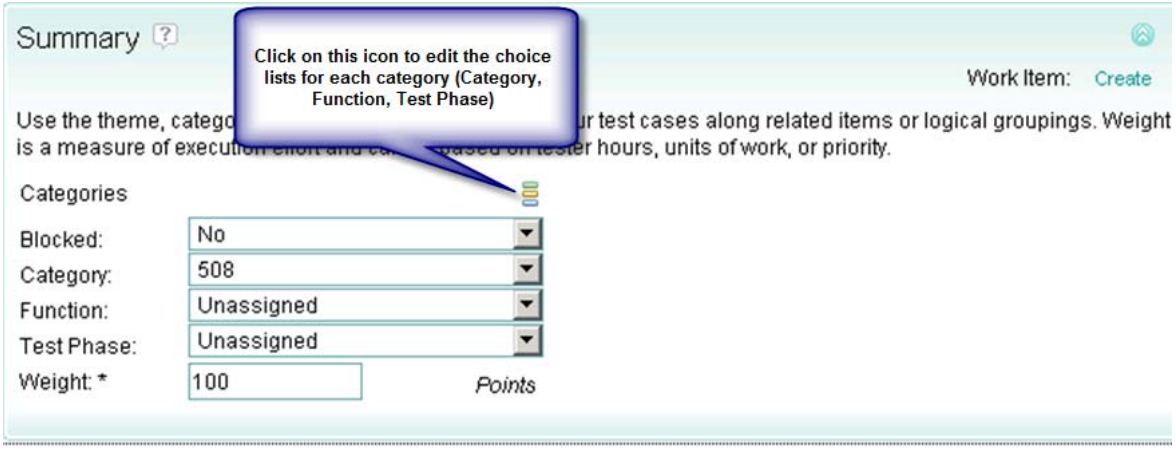


Figure 2-4: RQM Test Case Categories

Table 2-6 lists the predefined test case categories.

Display Name	Description	Initial Values
Category	Generic means of categorizing the test case.	Login Test Security Test Data Migration Test 508 Test Performance Test (Part of a full, formal Performance Test Suite) Performance Tune Test (Running one or two processes against the code) Inter-System Test Smoke Test Documentation Test Regression Test First Live Batch Test Post Implementation Verification Support Period Test Usability Test Functional Test Sub-System Test
Function	Means of categorizing the test plan based on the type of function being tested.	To be loaded by the CM Lead at project startup.

Display Name	Description	Initial Values
Test Phase	The type of testing being performed.	System Testing User Acceptance Testing

Table 2-6: RQM Test Case Categories

2.2 Rational Tool Integration

Central to configuration management is the ability to manage change requests and control changes to system development artifacts. Two Rational tools will be used to support these goals:

- ClearQuest: Change Control
- ClearCase: Version Control

The Rational tools have been tailored such that CRs and ChildCRs are linked to defects managed by ClearQuest, requirements managed by RequisitePro, file-based artifacts managed by ClearCase, and test cases managed by Rational Quality Manager. Section 4 provides additional details on the available integrations between the Rational tools.

2.3 Security

Security protocols for each of the Rational tools is primarily based on a user's membership in one of the following groups:

- CMTeam
- CMLead
- DevTeam
- DevLead
- FSA_Mgmt
- OperationsLead
- OperationsTeam
- RequirementsLead
- RequirementsTeam
- SecurityTeam
- TestLead
- TestTeam

The Leads for each group will be members of both the "Lead" group and the "Team" group. For instance: If the user Bob is the CM Lead for the project WidgetDev, he will be a member of the group CM Lead and CM Team.

Rational Quality Manager, RequisitePro, and ClearQuest each have built in group definitions which can be used to control access to artifacts managed by the tools. ClearCase security is

based primarily on membership in active directory groups and custom triggers that specify which users have permission to perform specific actions.

Table 2-7 summarizes the operations in each Rational tool that are reserved for the system's CM Lead.

Rational Tool/Component	Permitted Operation
RequisitePro	** No special permissions are reserved for the CM Lead. The CM Lead and CM Team are restricted to "read-only" access to RequisitePro requirements artifacts.
Rational Quality Manager	Manage categories associated with Test Plans and Test Cases. Delete various testing artifacts such as test plans, cases, and scripts.
ClearQuest	Manage Dynamic Name Lists and Stateless Records in order to control the content of CR, ChildCR, and Defect choice lists. See Tables 2-3 and 2-4.
ClearQuest/CRs	Manage CR Workflow: See Table 3-4.
ClearQuest/ChildCRs	Manage ChildCR Workflow: See Table 3-5.
ClearCase	Deliver changes to the project Integration Stream. Manage list of users that are permitted to deliver changes to the release-specific integration stream. Manage content of the system documentation VOB. Create ClearCase Labels in the system documentation VOB during the process of creating a baseline. Create ClearCase UCM Baselines on the UCM Project Integration stream.

Table 2-7: Rational Operations Reserved for the CM Lead

Additional details on role/group based permission are described in Appendix E.

Section 3. Configuration Control

Modification of system components is controlled through a formal configuration control process. The FSA Enterprise Solution has been tailored to support this process through automation and enforcement of standard FSA configuration management policies and procedures.

3.1 Change Control

A standard change control process will be used to manage and control software product creation and revisions. Rational ClearQuest has been tailored to automate and enforce FSA guidelines related to change request (CR) management. For each system under development, a ClearQuest database will be created and used to manage all change requests related to that system. The system-specific ClearQuest database will be based on a standard schema and tailored to system-specific needs.

3.1.1 ClearQuest Design Overview

Figure 3-1 shows the pertinent ClearQuest records used in the FSA configuration control record types and their relationship to one another.

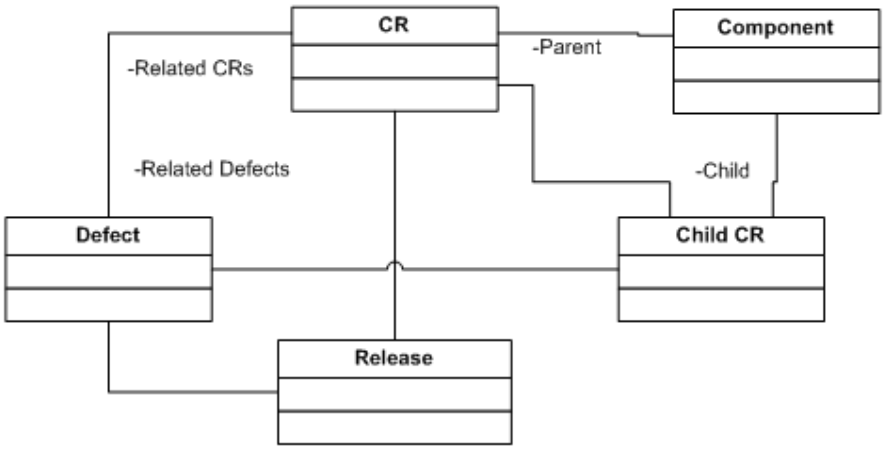


Figure 3-1: Related ClearQuest Record Types

While this section is primarily focused on Change Requests (CR) and Child Change Requests (ChildCR), these record types depend on Defects, Components, and Releases. All of these ClearQuest record types are as follows:

- **Change Request (CR) Record:** The CR is the primary record used to document and manage required changes to a production release of a system. A CR is opened in ClearQuest in order to document production level enhancements, corrections, and maintenance activities. Each CR will be linked to two "Release" records and at least one "Component" record.

- **Defect Record:** The defect record is used by testers to track all defects found during a phase of testing. Defect records can only be created by members of the test team and should only be created while executing a test case from within Rational Quality Manager.
- **ChildCR Record:** The ChildCR record is used to subdivide the work associated with a CR into more granular units which can be assigned to developers to implement.
- **Release Record:** The release record is used to categorize Change Requests and Defects.

The following choice lists associated with a CR record type are automatically populated based on the instances of the Release record type:

- **Found in Release**: A reference to a "Release" record which indicates the production release of the system that the Change Request pertains to. This choice list is populated based on inactive release records.
- **Targeted Release**: A reference to a "Release" record which indicates the future release of the system that the change is targeted to be included in. This choice list is populated based on active release records.

The following choice lists associated with a ChildCR record type are automatically populated based on the instances of the Release record type:

- **Targeted Release**: A reference to a "Release" record which indicates the future release of the system that the Defect is targeted to be included in. This choice list is populated based on active release records.

Both "Found in Release" and "Targeted Release" are references to records that must be created and maintained by the CM Lead for the system. For every release of an FSA system, there will be an equivalent release record stored in the ClearQuest database. It is the CM Lead's responsibility to maintain the list of active release records in ClearQuest. As new system releases are planned out, new ClearQuest release records are created that correspond to the releases.

Active releases are those releases that are in development or planned for development. Inactive releases include the current production release of the system as well as all previous releases that are no longer in production.

Component Record: The component record type is used to subdivide the system into functional areas. It is the CM Lead's responsibility to create component record types in the system's ClearQuest database that correspond to meaningful functional areas for the system under development. CRs and ChildCRs written against the system will be categorized by the component(s) to which they apply. Both CR and ChildCR record types contain the multi-select list field called Component which is a list of components of the system that will likely need to be changed as part of the implementation of the CR.

In addition to the fields listed above, the CR record type includes many fields that describe the characteristics of the change request; all of which are documented in detail in Appendix C: ClearQuest Record Details.

Figure 3-2 shows the workflow (state transition model) for a CR. The primary flow is a straight path from the "Submitted" state to the "Closed" state. Alternate transitions are possible including cancelling a CR, placing a CR on hold or transitioning the CR back to previous states. The image shows the possible state transitions as well as which user groups are allowed to initiate the transition. Each arrow is labeled as follows: <Action Name [groups allowed to perform the action]>. The off page connectors CQ-CR1 and CQ-CR2 reference the defect processing flowchart in Figure 3-5.

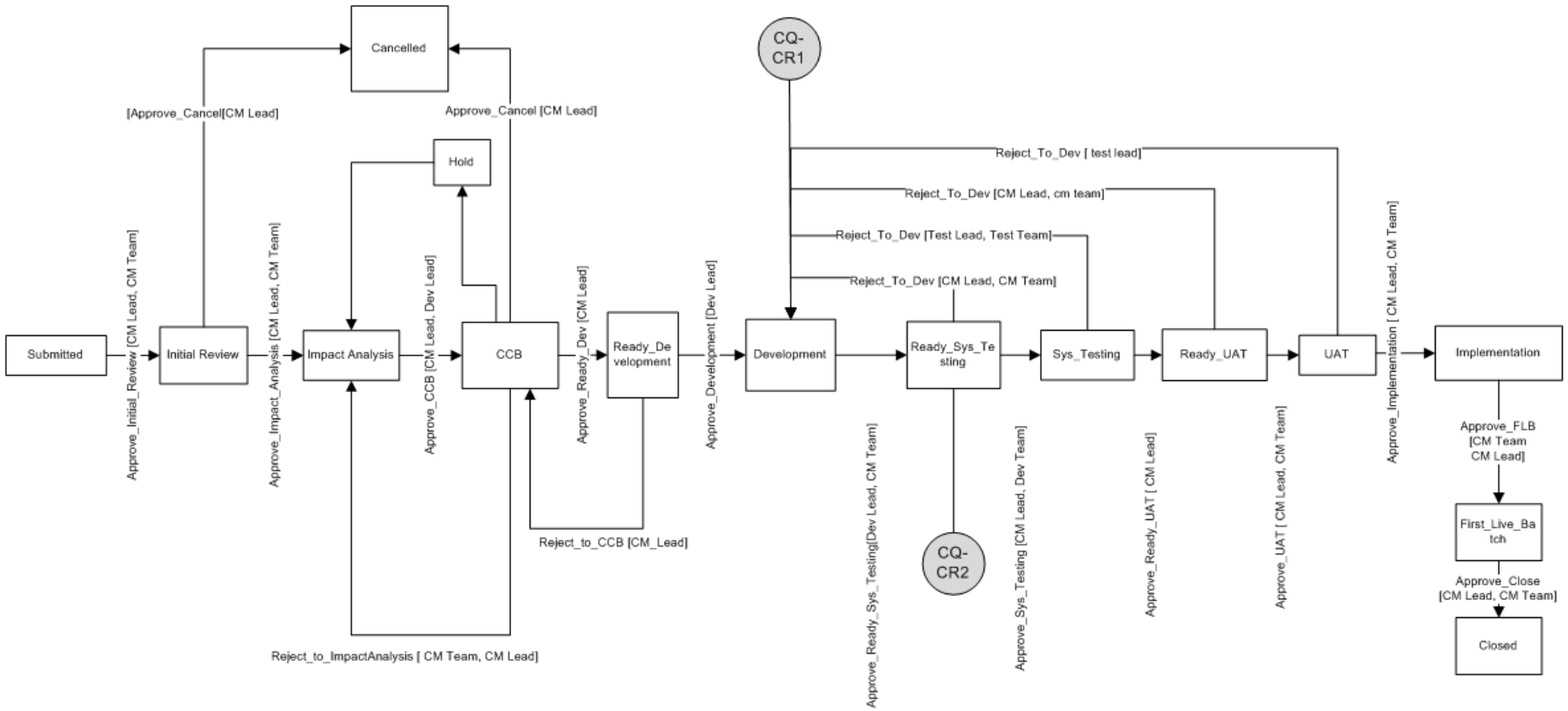


Figure 3-2: CR Record Workflow

When the effort involved in implementing a CR is likely to be large enough to require multiple developers, ChildCRs are created and linked to the "parent" CR. ChildCRs are used to subdivide work among multiple developers. Each CR will contain a list of zero or more childCRs. The state model shown in Figure 3-3 shows the life of a ChildCR from the point where it is created from a parent CR through closure. The off page connectors CQ-CR1 and CQ-CR2 reference the defect processing flowchart in Figure 3-5.

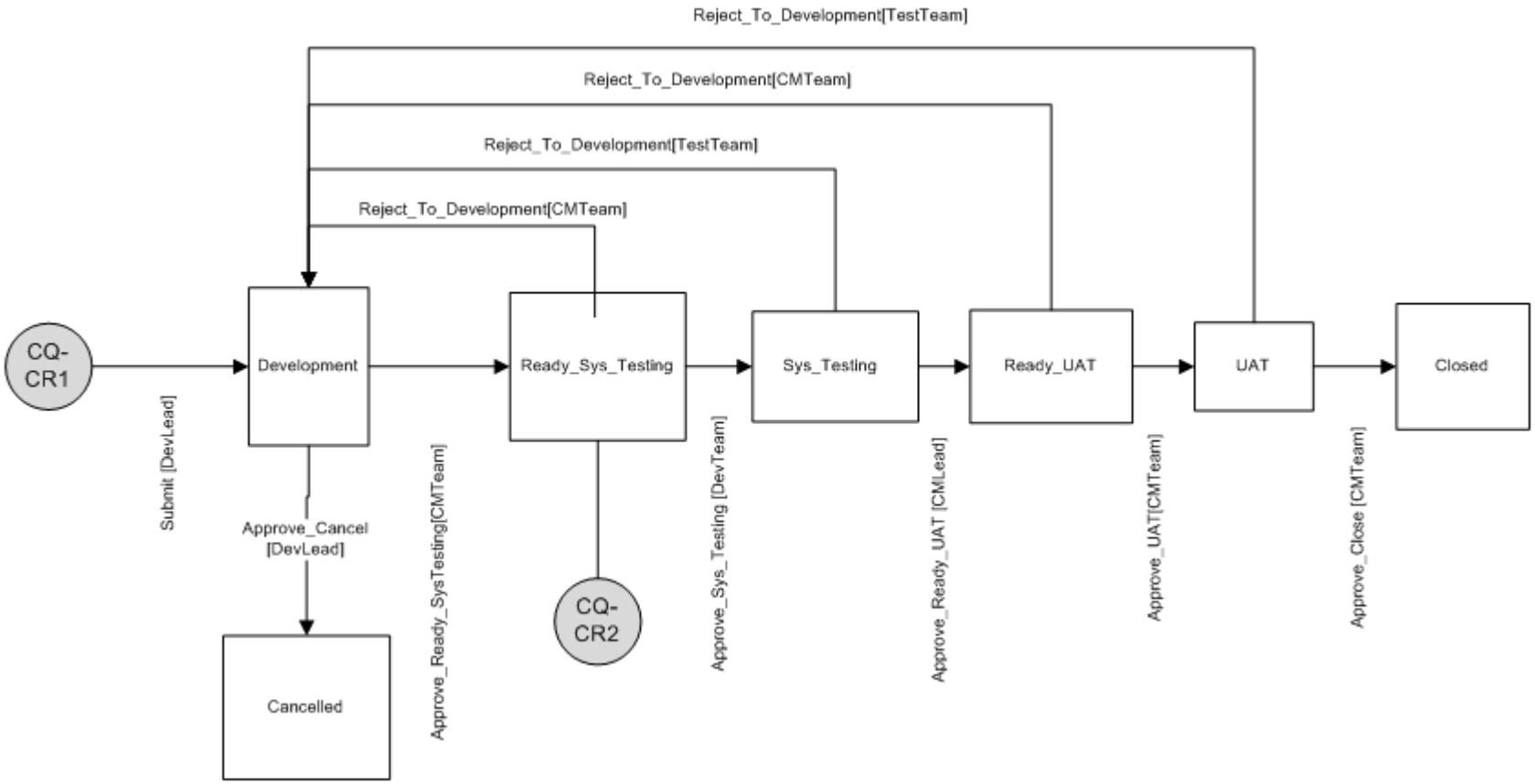


Figure 3-3: ChildCR Record Workflow

Figure 3-4 shows the workflow (state transition model) for a Defect. The primary flow is a straight path from the Submitted state to the Closed state. Alternate transitions are possible including cancelling a defect, or transitioning the defect back to dev-review states from the re-test state. The image shows the possible state transitions as well as which user groups are allowed to initiate the transition. Each arrow is labeled as follows: <Action Name [groups allowed to perform the action]>. The off page connectors CQ-CR1 and CQ-CR2 reference the defect processing flowchart in Figure 3-5.

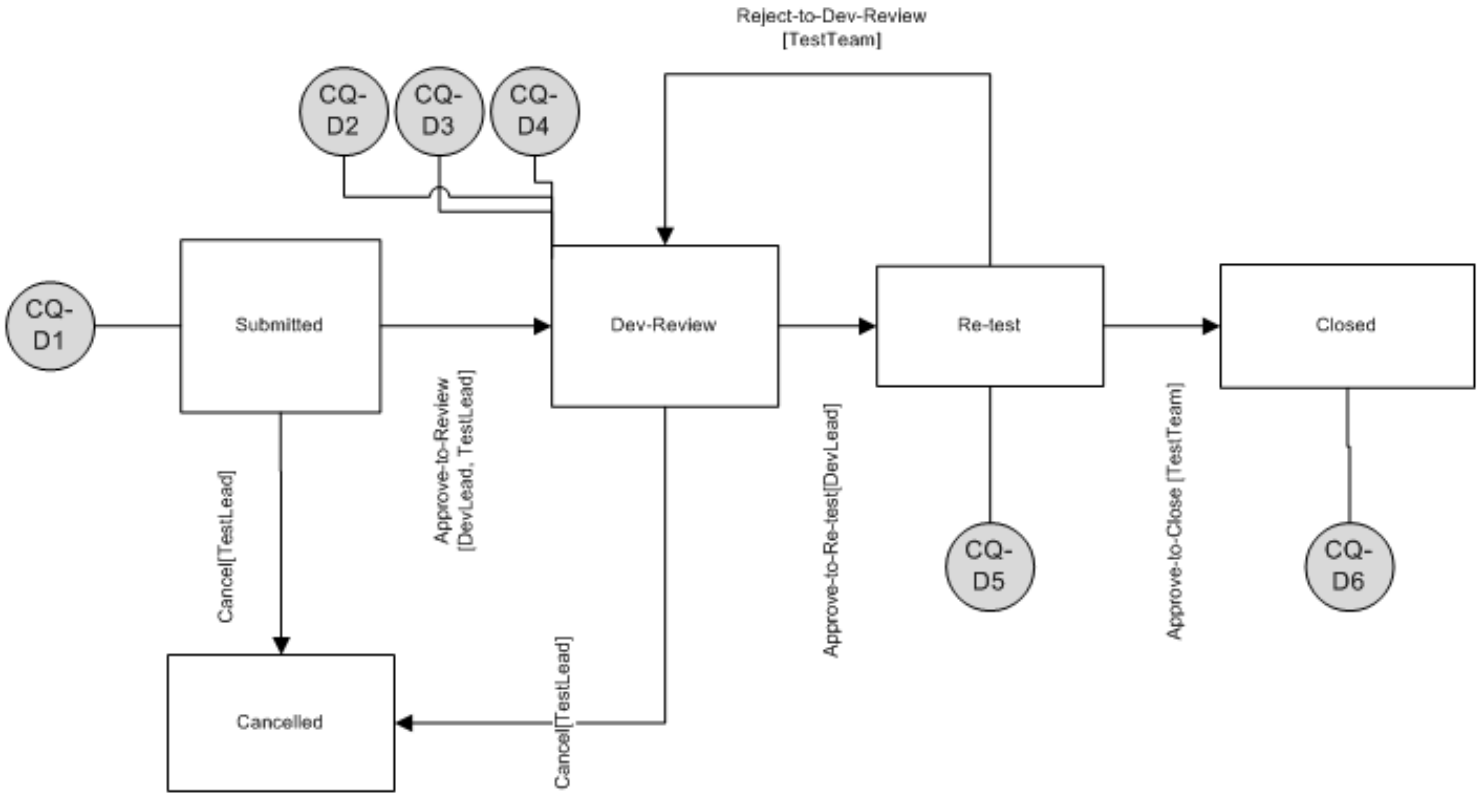


Figure 3-4: Defect Record Workflow

Change Requests and Defects do not exist independently of one another. The flowchart shown in Figure 3-5 shows how defects and change requests (CRs and ChildCRs) are intertwined when a defect is discovered, analyzed, corrected, and ultimately closed. The off page connectors CQ-CR1, CQ-CR2, CQ-D1, CQ-D2, CQ-D3, CQ-D4, CQ-D5 and CQ-D6 show how the processes depicted in the flowchart relate to the state transition models/workflows for CRs, ChildCRs, and Defects shown in Figures 3-4, 3-3, and 3-2.

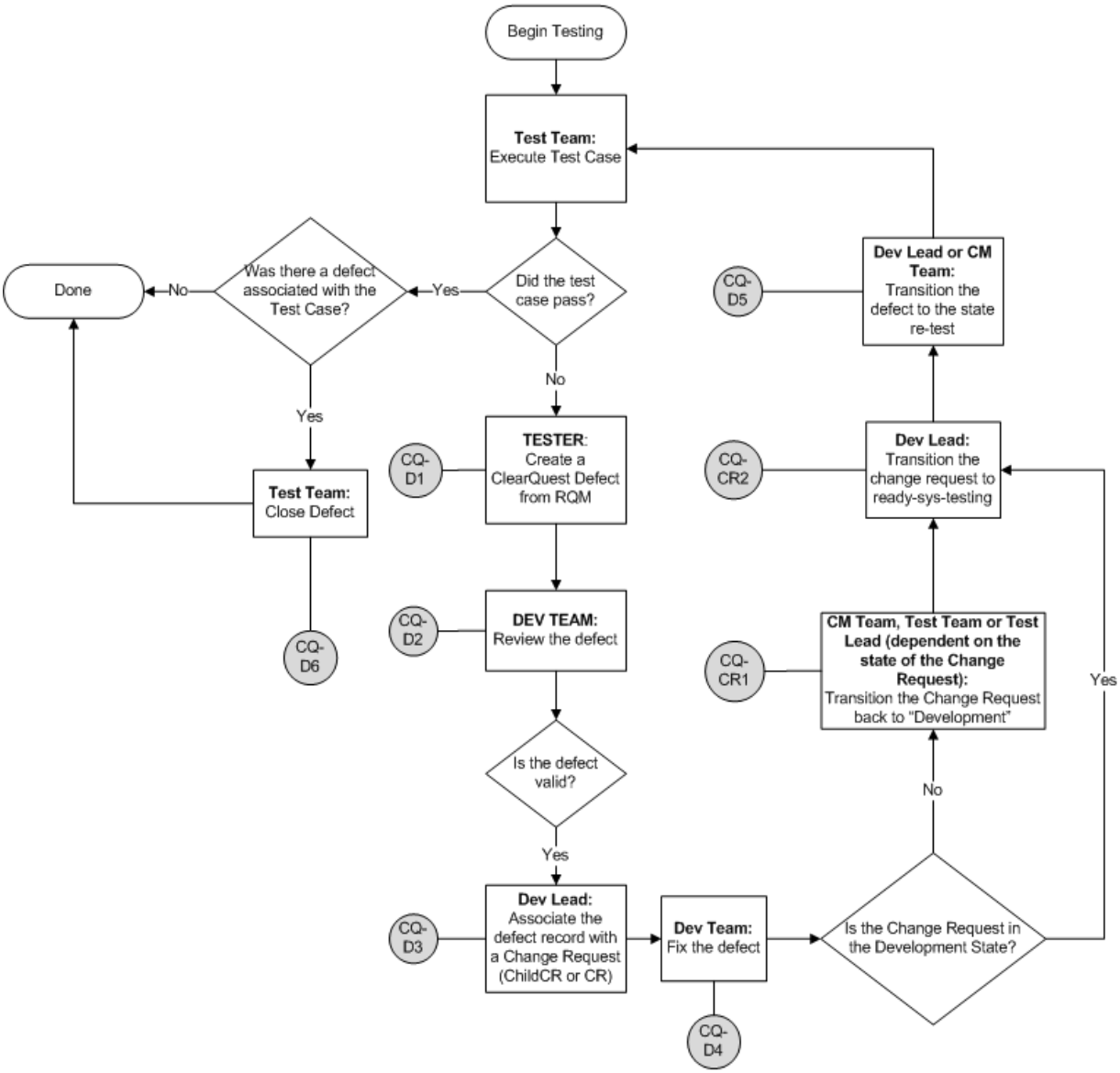


Figure 3-5: Defect Resolution Flowchart

3.1.2 Working with Change Requests

Processing a CR entails transitioning the CR through various states where each state represents the work that is being performed relative to the CR. For example: A CR that has been transitioned to the "Impact_Analysis" state means that the CR is being evaluated by the requirements team, the development team, and the security team in order to determine the impact of implementing the CR.

By the time that the approved CR has progressed through the workflow from submission to closure, the requested change will have been incorporated into a baselined release and deployed into "First Live Batch".

Table 3-1 describes the states associated with the primary workflow.

CR State	Description
Submitted	The change requested in this CR has been submitted but not yet reviewed.
Initial_Review	The change requested in this CR is being briefly reviewed for being appropriately submitted. Impact analysis assignments should be made while in this state.
Impact_Analysis	The change requested in this CR is being reviewed for impact to requirements, development, and security by the assigned team members.
CCB	The impact analysis information has been completed and the CR is ready for consideration by the CCB. The CCB will determine if the CR will be processed.
Development	The CCB has approved the change request and the relevant development work should be underway. Note that the requirements changes must precede or accompany the software development effort because functionality is developed according to requirements.
Ready_Sys_Testing	The Development Lead has determined that the software changes are complete, have passed unit tests, and should be included in the next release to system test so that system tests may begin for this CR. If this CR represents a change to Requirements only, the CM Lead will determine how to process this CR through the remainder of the lifecycle. If this CR represents a change to a document only, the CM Lead will determine how to process this CR through the remainder of the lifecycle.
Sys_Testing	The new baseline release that includes the software changes that implement this CR have been deployed to the system testing environment. The test team should begin system testing of this CR.
Ready_For_UAT	The system testing has been successful. The Test Lead has determined that software changes that implement this CR should be included in the next release to User Acceptance Test (UAT).

CR State	Description
UAT	The new baseline release that includes the software changes that implement this CR have been deployed to the UAT environment. The test team should begin UAT testing of this CR.
Implementation	The UAT testing has been successful. The Test Lead has determined that software changes that implement this CR should be included in the next release to production First Live Batch (FLB).
FLB	The new baseline release that includes the software changes that implement this CR have been deployed to the production FLB environment. Standard production verification procedures should now occur.
Closed	The CR has been verified in production FLB as correct, or it has been reported as an issue and a new CR has been submitted to resolve the issue.

Table 3-1: Table of CR States in the Standard CR Lifecycle

Each state in the lifecycle of a CR relates to work being performed by one or more individuals. Refer to the system CM plan for additional details the CR process model.

As mentioned previously, any change to a baselined artifact will require submission, processing, and approval of a formal Change Request managed by Rational ClearQuest.

Opening a Change Request

Any user that has been granted access to the system-specific instance of ClearQuest is permitted to open a CR. CR records are opened in order to request a change to the system in production. Change Requests are **NOT** used to track defects identified during testing. Change requests are categorized as enhancements, corrections, or maintenance requests.

In order to open a change request, specific fields must be completed by the submitter. Figure 3-6 shows a new change request and fields that are mandatory in order to commit the CR. All red italicized fields must be entered in order to complete the submission process. After all fields have been entered, the user will click on the “OK” button to commit the record.

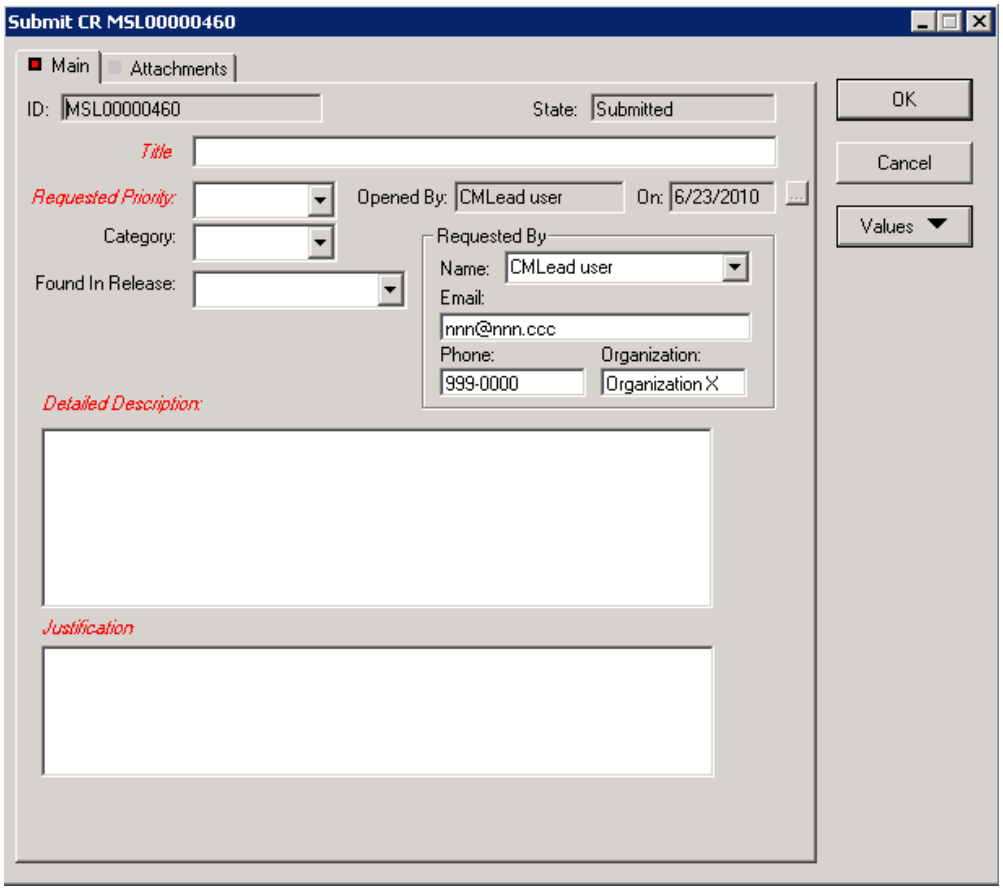


Figure 3-6: Opening a New Change Request

Table 3-2 shows the fields that are required to be entered by the person opening the CR in order to commit the record to the database.

Display Name	Description	Required on Submit	Field Value Choices
ID	CR's identifier	n/a	Auto-generated by ClearQuest
State	CR's state/status	n/a	Auto-populated and updated by ClearQuest
Title	Short description of the CR.	Yes	Name
Requested Priority	Requested priority of the CR.	Yes	Routine Urgent Emergency

Display Name	Description	Required on Submit	Field Value Choices
Category	A means of categorizing the CR.	No	Correction Enhancement Maintenance
Found In Release	The release of the application that the CR is being written against.	No	List of inactive release records. NOTE: Active release records are used to populate the "Assigned Release" field. Inactive release records are used to populate this field.
Detailed Description	Multi-line, detailed description of the change being requested.	Yes	Free-form text
Justification	Reason(s) why the implementation of the CR will improve the FSA environment.	Yes	Free-form text
Opened By	Person that entered the CR into ClearQuest.	n/a	Auto-populated by ClearQuest
Opened On	Date that the CR was entered into ClearQuest.	n/a	Auto-populated by ClearQuest
Name (Requested By)	Name of the person requesting the CR.	Yes	Free-form text (can be populated using the drop down list of existing ClearQuest users)
Phone (Requested By)	Phone number of the person requesting the CR.	Yes	Free-form text
Organization (Requested By)	Organization that the person belongs to requesting the CR.	Yes	Free-form text
Email (Requested By)	E-Mail address of the person requesting the CR.	Yes	Free-form text

Table 3-2: CR Submit Form – Main Tab

The person opening the CR may, optionally, attach one or more files to the CR record via the "Attachments" tab as shown in Figure 3-7.

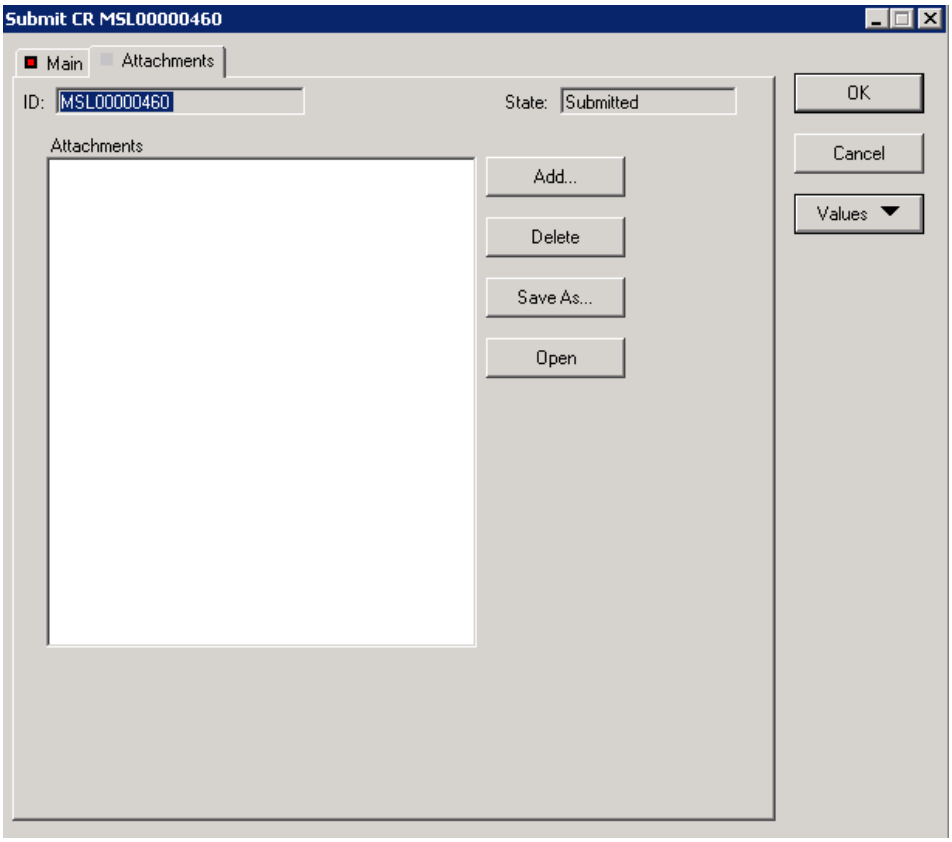


Figure 3-7: CR Submit Form – Attachments Tab

Display Name	Description	Required on Submit	Field Value Choices
ID	CR's identifier	n/a	Auto generated by ClearQuest
State	CR's state/status	n/a	Auto populated and updated by ClearQuest
Attachments	External files that can be attached to the CR. NOTE: The files will be stored directly in the DB.	No	n/a

Figure 3-8: Attachment-Related Fields

3.1.2.1 Working with ChildCR Records

A ChildCR record is created by the Development Lead in the event that a CR needs to be divided into smaller units of work. This is typically done when the effort involved to implement the CR requires more than a single developer. A ChildCR could also be created simply because it makes sense to divide the CR into smaller, more manageable units of work even if the work will not be assigned multiple developers.

Opening a ChildCR

Figure 3-9 and Figure 3-10 show the tabs on the ChildCR submit form which require user input when a new ChildCR is created.

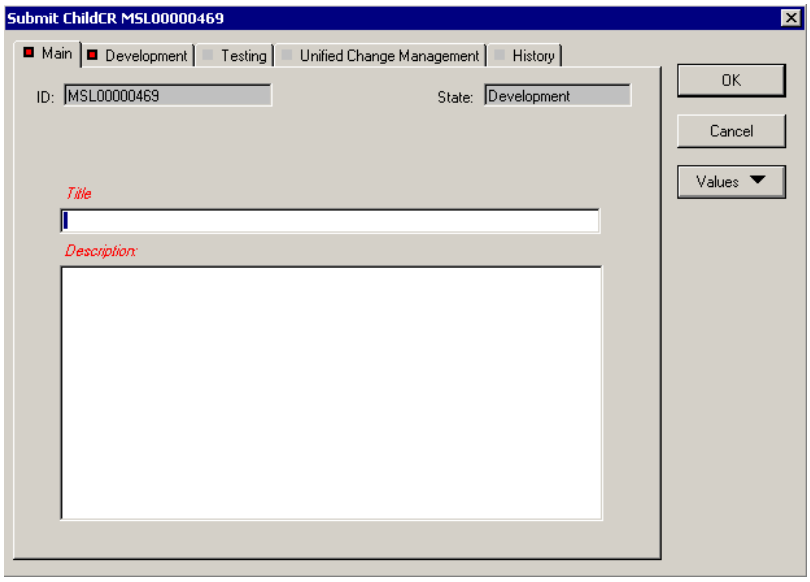


Figure 3-9: ChildCR Submit – Main Tab

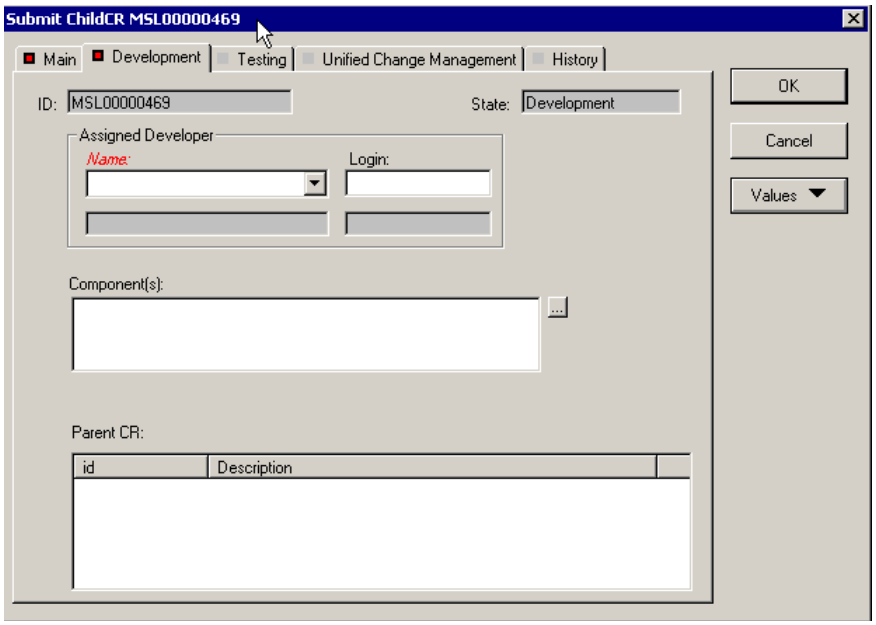


Figure 3-10: ChildCR Submit – Development Tab

Each state in the lifecycle of a ChildCR relates to work being performed by one or more project team members. Table 3-3 describes the various states that a ChildCR can be in, the individual that is primarily responsible for the ChildCR while in that state, what work is performed while the ChildCR is in the state, and the output that is produced before the ChildCR is transitioned to the next state.

State	Who Leads	What Happens	Output
Development	Dev Lead	Creates the ChildCR from the parent CR and assigns it to a developer to work on.	An assigned ChildCR that can be used to work on changes to the product baseline.
Ready for System Testing	CM Team	After the developer has completed working on the ChildCR, the CM Team member is informed. Once the CM Team member validates the changes to the ChildCR, the ChildCR is turned over to the testing team.	The system has been updated based on the ChildCR. The updated system is ready to be tested.
System Testing	Dev Team	After the CM Lead validates the changes related to the ChildCR, the CR can be transitioned to System Testing by the Dev Team where the test team will conduct formal system testing.	Successfully executed system test cases.

State	Who Leads	What Happens	Output
Ready for UAT	CM Team	The ChildCR is held in a staging area until the Dev Team can move it to the next state (UAT).	The ChildCR is ready to be included in User Acceptance Testing.
UAT	CM Team	The ChildCR is included in User Acceptance Testing.	Successfully executed User Acceptance Test Cases.
Closed	CM Team	The ChildCR is complete.	

Table 3-3: ChildCR Step Action Table

3.1.2.2 Working with Defect Records

Table 3-4 shows the series of steps taken when a defect is identified and corrected.

Person	Action
Tester	Creates a defect record in ClearQuest in response to a failed test case.
Test Lead	Transitions the defect to Dev-Review.
Dev Lead	Reviews the Defect and associates a CR with the defect or contacts the Test Lead with reasons why the issue identified is NOT a defect so it can be cancelled. NOTE: In order for a CR to be associated with a defect, the following conditions must be true: 1) The target release of the CR and the defect must be the same. 2) The defect must be in the “Dev-Review” state. 3) The CR must be in the “Development” state.
Test Lead	Based on Dev Lead review, either rejects the related CR back to Development or agrees to cancel the defect.
Developer	Corrects the defect, documents the changes, and promotes the CR and Defect back to testing, and retest, respectively.
CM Team	Promotes the CR to Ready-for-system-testing after completing a build and creating a new version of the product baseline
Tester	Retests and closes the defect after verifying that it has been corrected. The tester may also need to promote the CR to Ready-for-UAT if the defect was the result of a failed test case during UAT. The CM team would then need to provide a build of the system and a new version of the baseline for UAT for another round of UAT testing.

Table 3-4: Defect Step-Action Table

Opening a Defect Record

Defects should only be created from RQM when a verification point in a test case fails or when the test case fails completely. Figure 3-11 shows the submit screen for a defect record created from inside RQM.

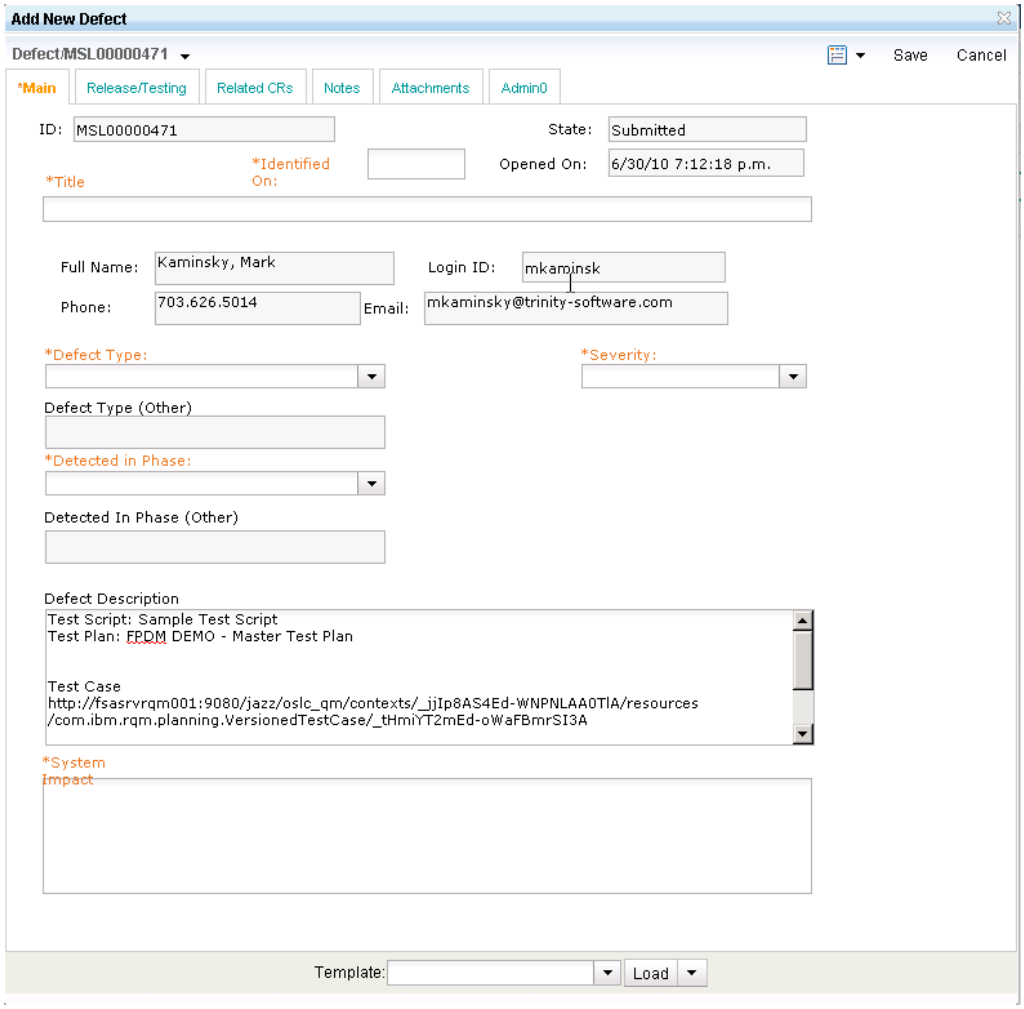


Figure 3-11: Submitting a Defect from RQM

3.1.3 Email Notification

Email rules have been established in order to automatically notify various groups and users when the CR transitions from state to state. An email will be automatically sent to each user who is associated with a role that is relevant to the state that a CR is changed to. For example, when a CR is changed to the Impact Analysis state, an email will be sent to those users in the roles of Development Lead, Requirements Lead, and Security Lead because they will need to provide an impact analysis for this CR.

The format of the email notification will be as follows:

Subject Line [Request State Transitioning To] – [Request ID] – [Request Title]
Message Body:
Created/Modified By: [User ID of Individual Taking Action]

This request may require your attention: you may be assigned to work on this request, assigned to approve someone else's work, or you are just being notified of a status change.

[hyperlink to the ClearQuest record]

3.1.3.1 Email Rules for CRs

Table 3-5 shows which CR actions initiate an email notification and which roles receive the email.

Actions	Roles that Email is Sent To
Approve_Initial_Review	CM Team
Approve_Impact_ Analysis	Development Lead, Test Lead, Requirements Lead, Security Lead
Approve_CCB	CM Team
Approve_Development	Development Lead, Test Lead, Requirements Lead, Security Lead
Approve_Ready_SysTesting	CM Team, Test Team
Approve_Sys_Testing	Test Team
Approve_Ready_for_UAT	CM Team
Approve_UAT	Test Team
Approve_FLB	Development Lead, Test Lead, Requirements Lead, Operations Lead
Approve_Cancel	Development Lead, Test Lead, Requirements Lead, Security Lead, Submitter
Reject_to_Development	Development Lead, Development Team
Reject_to_ImpactAnalysis	Development Lead, Test Lead, Requirements Lead, Security Lead

Table 3-5: CR Email Notification Rules

3.1.3.2 Email Rules for ChildCRs

Table 3-6 shows which ChildCR actions initiate an email notification and which roles receive the email.

Actions	Roles that Email is Sent To
Submit	Developer, Test-Lead
Approve_Ready_Sys_Testing	CM Team, Test Team
Approve_Sys_Testing	Test Team
Approve_Ready_for_UAT	CM Team
Approve_UAT	Test Team
Approve_Cancel	Dev Lead, Test Lead
Reject_to_Development	Dev Lead, Dev Team

Table 3-6: ChildCR Email Notification Rules

3.1.3.3 Email Rules for Defects

Table 3-7 shows which Defect actions initiate an email notification and which roles receive the email.

Actions	Roles that Email is Sent To
Submit	Developer, Test-Lead
Approve-to-Review	Dev-Lead
Approve-to-Re-Test	Test-Team

Table 3-7: Defect Email Notification Rules

3.2 Version Control

Version control is the processes and procedures used to manage changes to the actual configuration items (CIs) as they are being developed and tested. The CM repository allows a CI to be checked out and checked in once work on that item has been completed. To ensure only authorized changes are made, CIs can only be modified using a CR that is in the development state of the CR workflow. This approach integrates the change control process with the version control process to ensure the integrity of CIs throughout the life of the system.

All systems development artifacts will be managed and controlled by a system-specific instance of ClearCase. One or more ClearCase Version Object Bases (VOBs) will be created in order to store the system development artifacts. ClearCase will be used for the version control of systems development artifacts, including documentation. Examples of systems development artifacts under version control:

- systems development documents such as design, architecture, data conversion, etc

- source code
- script files
- configuration files
- COTS custom code files
- bundled COTS file packages
- system baselines (Functional, Design, and Product)

Each project will utilize one or more ClearCase VOBs in order to store the system-specific system development artifacts. Table 3-8 lists the types of VOBs that will be created for each system development effort.

VOB Name	Type of VOB	Purpose
<System>_Documentation	Base ClearCase VOB	Storage of system-related documentation including functional and design baselines.
<System>_PVOB	UCM Project VOB	ClearCase system VOB used for each system that will utilize UCM component VOBs. A Project VOB (PVOB) is used by ClearCase to store UCM project-specific metadata.
<System>	UCM Component VOB	Storage of source code. A single UCM component VOB will be used when the system being developed does not warrant the use of multiple component VOBs.
<System>_<Component >	UCM Component VOB	Storage of source code. Multiple component VOBs will be utilized when a system needs to be subdivided into ClearCase components. This is typically necessary only when the components need to be baselined and delivered independently from the rest of the system. In this case, a component could be viewed as a sub-system.

Table 3-8: VOBs Used in a System Development Effort

An example of the use of VOBs are as follows: Given the fictitious system named "Unified Widget Framework" composed of components "GUI" and "DB", we would have the following VOBs:

- System's Documentation VOB: \UWF_Documentation
- System's UCM Project VOB: \UWF_PVOB
- System's Component VOBs: \UWF_GUI and \UWF_DB

3.2.1 Project Documentation VOB

The project's documentation VOB will be created by the FSA Rational Support Group (RSG) at the initiation of the system's development effort. This VOB will be created as a "base ClearCase" VOB with a single "main" branch and no sub branches.

All work performed in the system's Documentation VOB will be performed on the Main branch using the standard/default ClearCase configuration specification as follows:

element * CHECKEDOUT
element * /main/LATEST

Standard base ClearCase label types and label instances will be utilized when creating functional, design, and product baselines in the project's Documentation VOB. The CM Lead will be responsible for creating baseline label types and applying that label to elements in the project's Documentation VOB which correspond to the elements that are included in the baseline. The baseline naming conventions specified in the system-specific Configuration Management Plan will be adhered to when creating the label types.

Refer to the ClearCase Reference Guide for detailed instructions on the use of the commands "mklbtype" and "mklabel".

3.2.2 Project VOB (PVOB) and Component VOBs

For any system consisting of one or more system components, a ClearCase UCM Project (PVOB, UCM Project, and Component VOBs) will be established at the initiation of the system-development effort. The basic structure of the project will be as follows:

- One FSA system-specific ClearCase UCM Project VOB will be created.
- One or more ClearCase UCM component VOBs will be created in order to manage the project's code base.

If only a single component VOB is required then the VOB will be named based on the system's identifier <system>. For example: "\UWF". If multiple component VOBs are required or if there is a possibility that additional component VOBs may be required in the future, the VOB will be named "<System>_<Component>". For example: "\UWF_GUI".

Table 3-9 defines the ClearCase VOBs and recommended first level directories below the VOB level directory. The CM Lead will be responsible for creating and managing the initial set of directories within each VOB.

VOB Name	VOB Type	Directories	Description
<system>_Documentation	Base ClearCase VOB	ConfigurationManagement Design Operations ProjectManagement	Contains all documentation for the system under development.

VOB Name	VOB Type	Directories	Description
		Requirements SystemSecurity Testing	
<system>_PVOB	ClearCase UCM Project VOB	None	This VOB is managed by ClearCase and is used exclusively for storing UCM Project-related metadata.
<system>_[purpose designator]	ClearCase UCM Component VOB	<determined by CM Lead>	Contains all source code for a system under development. In the event that more than a single UCM VOB is required to support the system development effort, an additional designator will be added to the VOB name representing the purpose of the VOB.

Table 3-9: Recommended VOBs and Directories

3.2.3 Stream Management

Every UCM Project will be created as a standard "Parallel Development" project. This means that there will be a project level integration stream (UWF_Integration) and one or more child streams.

The stream hierarchy will be as follows:

- **<system>_<release>_Integration:** Official product baselines will reside on this stream. NOTE: Only the CM Team has permission to checkout files on this stream. The only time that files should be modified on this stream is when a delivery from the release-specific stream (see below) is being performed.
- **rel-<release ID>:** Changes made on the shared development stream will be delivered to this release-specific integration stream in order to create interim baselines used for system testing. The CM Team has permission to checkout, checkin, and perform delivery operations on this stream. The CM Lead may provide the ability to deliver to this stream. The only time that files should be modified on this stream is when a delivery from the shared development stream (see below) is being performed.
- **rel-<release ID>-dev**: Based on assigned change requests, the developers will checkout, edit, and checkin code on this shared development stream.

A UCM Baseline Label will be created on the Project Integration stream (see PB_00.00 in the example below). It is this baseline that represents the starting point for developing the next major release of the system.

A release-specific stream is created off of the UCM project integration stream in order to start development of a new release.

The release-specific integration stream (see rel-01.00 below) will be created as a child of the UCM project's integration stream at the point designated by an existing product baseline (PB_00.00 in the example below).

Developers will NOT work directly on this release-specific integration stream. Rather, a child stream will be created off of the release-specific integration stream at the point of the initial release-specific baseline (see PB_01.00.000 below).

Developers will work on the shared release-specific development stream. As changes on the shared release-specific development stream are completed they will be delivered to the release-specific integration stream.

A release-specific baseline will be created in order to support system testing.

When system testing completes successfully, the changes will be delivered to the project integration stream where a Product Baseline will be created. It is this product baseline that is used for user acceptance testing.

When a new major release of the system is to begin, a new ClearCase UCM project will be created off of one of the product baselines located on the previous project's UCM Integration stream.

Figure 3-12 shows a graphical representation of this stream strategy given a single release of a system.

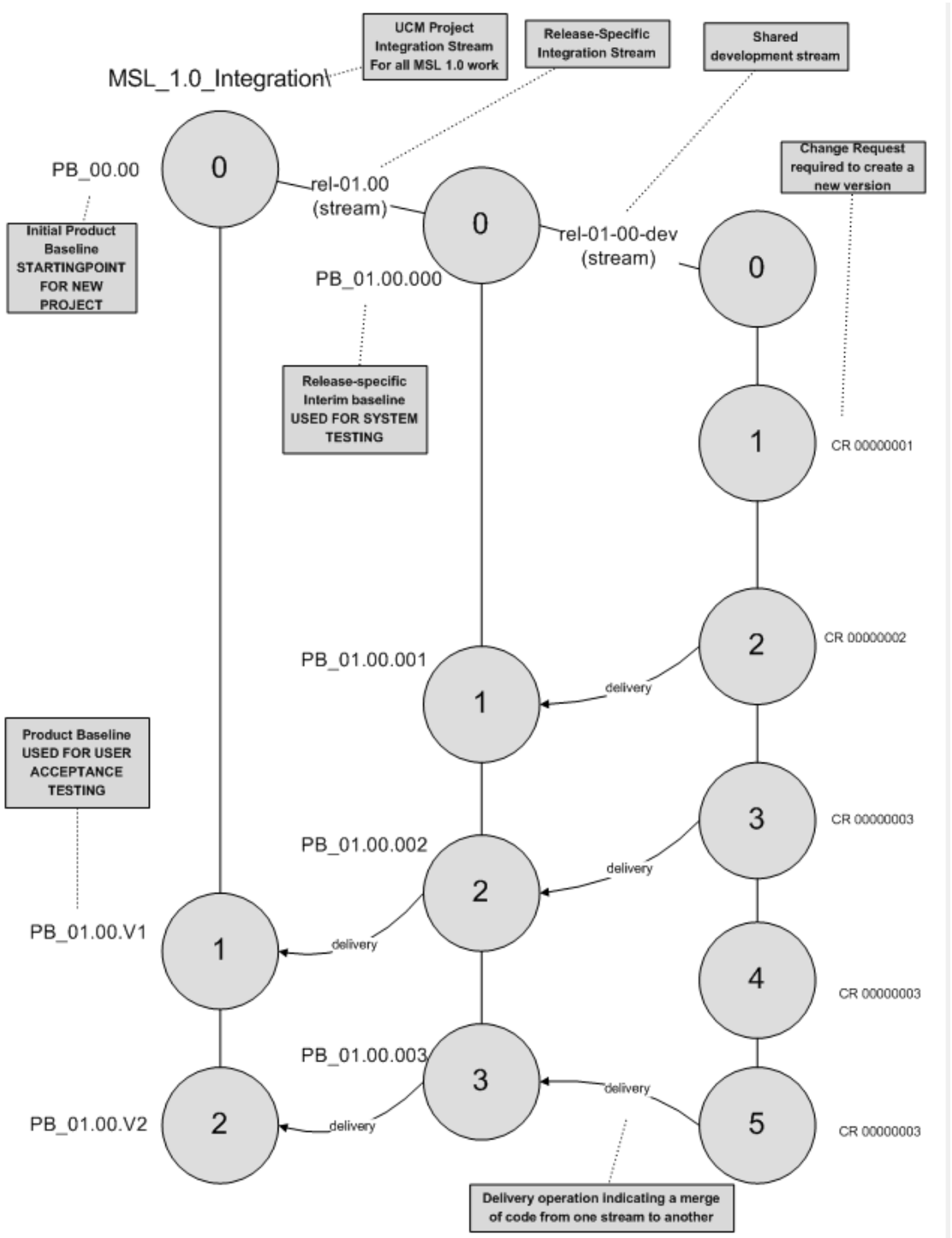


Figure 3-12: Stream Structure for a Single Release

Figure 3-13 shows how the stream structure will change when a new release of the system is started. Notice that the release-specific integration stream (Rel-02.00) starts at the product baseline PB_01.00.V2.

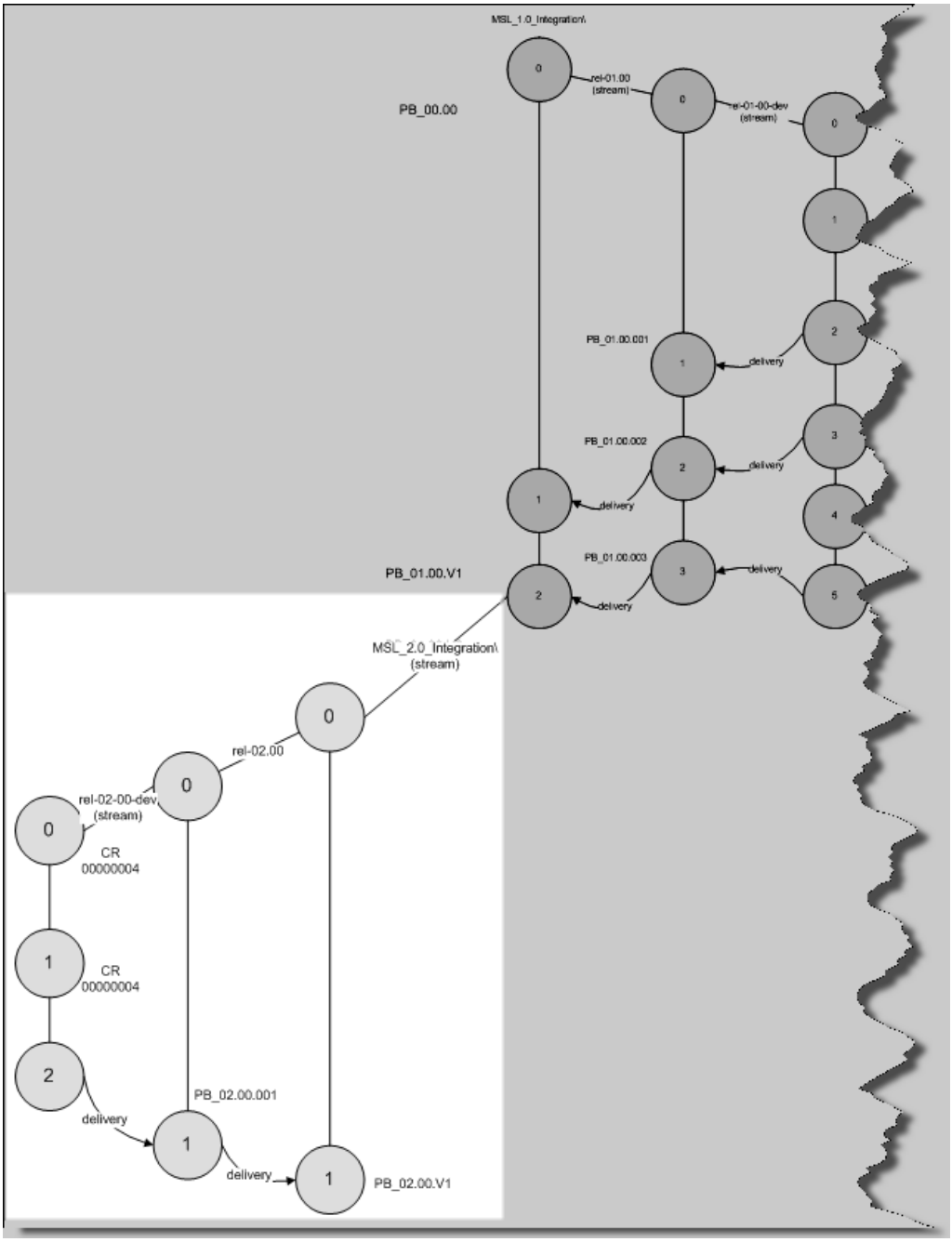


Figure 3-13: Stream Structure for a New Release

Figure 3-14 shows how to handle the case where there are two releases of a system being developed simultaneously. This scenario could occur when it is necessary to begin working on a new release of a system prior to the completion of a prior release. In the example below, development for MSL 2.0 was started off of product baseline PB_01.00.v2 while changes were still being made to MSL 1.0.

Once MSL Baseline PB_01.00.V3 get's created, it can be propagated to MSL 2.0 development. The bold black arrow shows how a baseline from MSL 1.0 can be merged into the MSL 2.0 shared development stream using the ClearCase UCM Cross Project Delivery capability.

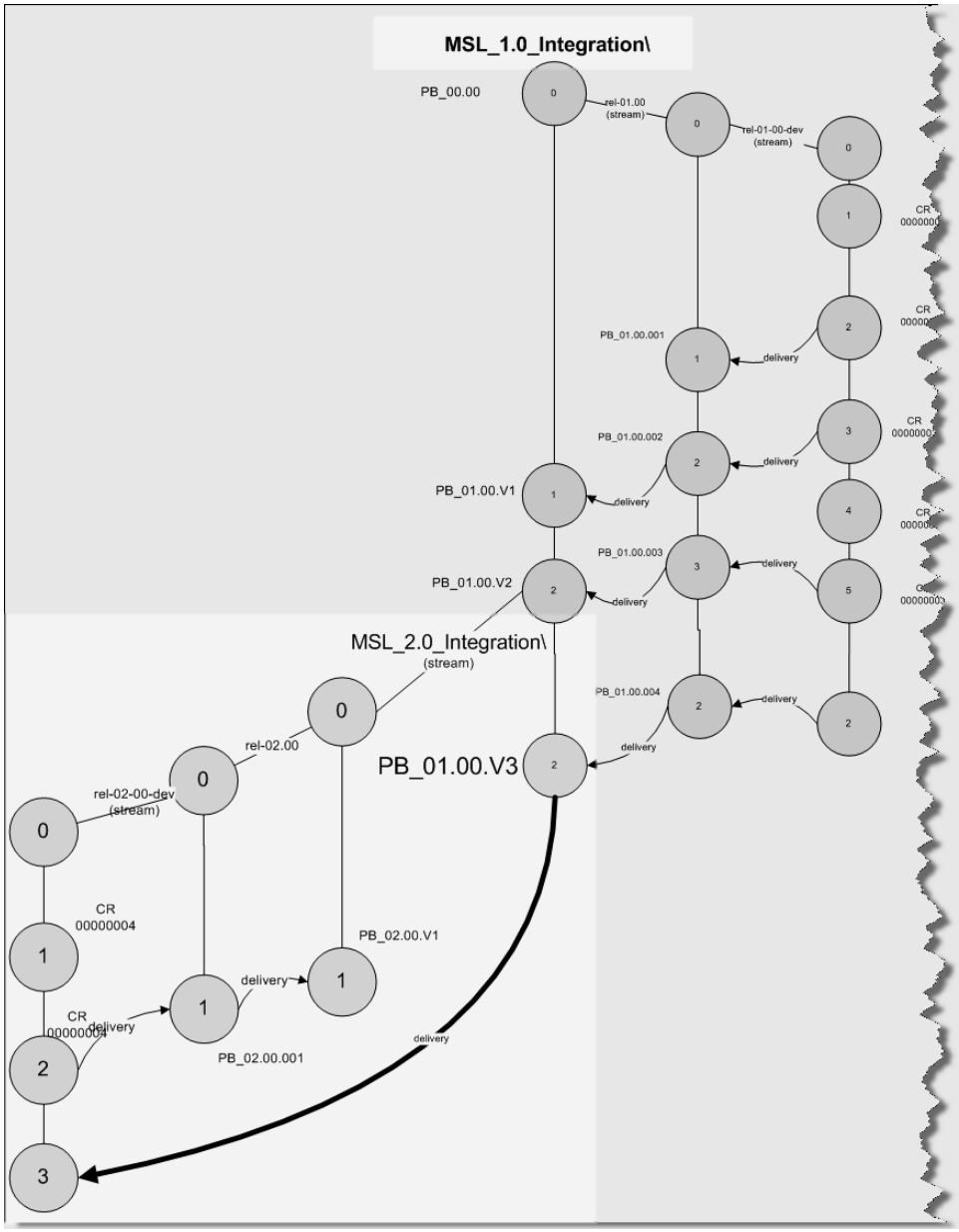


Figure 3-14: Cross Project Delivery

Stream and Baseline Related Policies:
UCM policies and procedures that are being enforced are described in detail in Appendix E: Security.

Section 4. Tool Integrations

Change Requests do not exist in isolation. The Rational tools have been tailored such that CRs and ChildCRs are linked to defects managed by ClearQuest, requirements managed by RequisitePro, file-based artifacts managed by ClearCase, and test cases managed by Rational Quality Manager.

4.1 ClearQuest to RequisitePro

A ClearQuest CR can be linked to one or more requirements managed by RequisitePro from the "Requirements" tab on the CR form. Requirements are associated with a CR under one of the following circumstances:

a) A change request has been submitted against one or more requirements.
b) A change request has been written as a request for new functionality (enhancement) to the system under development. In this case, the details of the request are elaborated during a requirements effort and linked back to the CR for reference.

Figure 4-1 shows the "Requirements" tab from a CR. If this CR was linked to one or more requirements managed by RequisitePro, the requirements would appear in the "Associated Requirements" field.

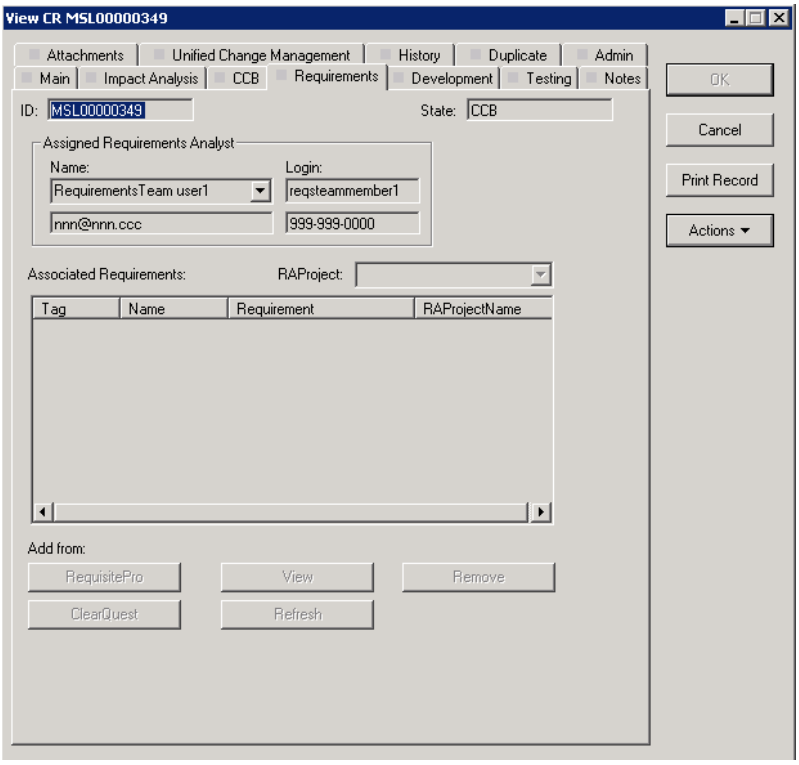


Figure 4-1: Change Request – Requirements Tab

The link between the CR and Requirements can also be viewed from RequisitePro. As shown in Figure 4-2, each requirement has an attribute labeled "Change Request". If the requirements listed below were linked to CRs then the CR ID number would appear in the "Change Request" column.

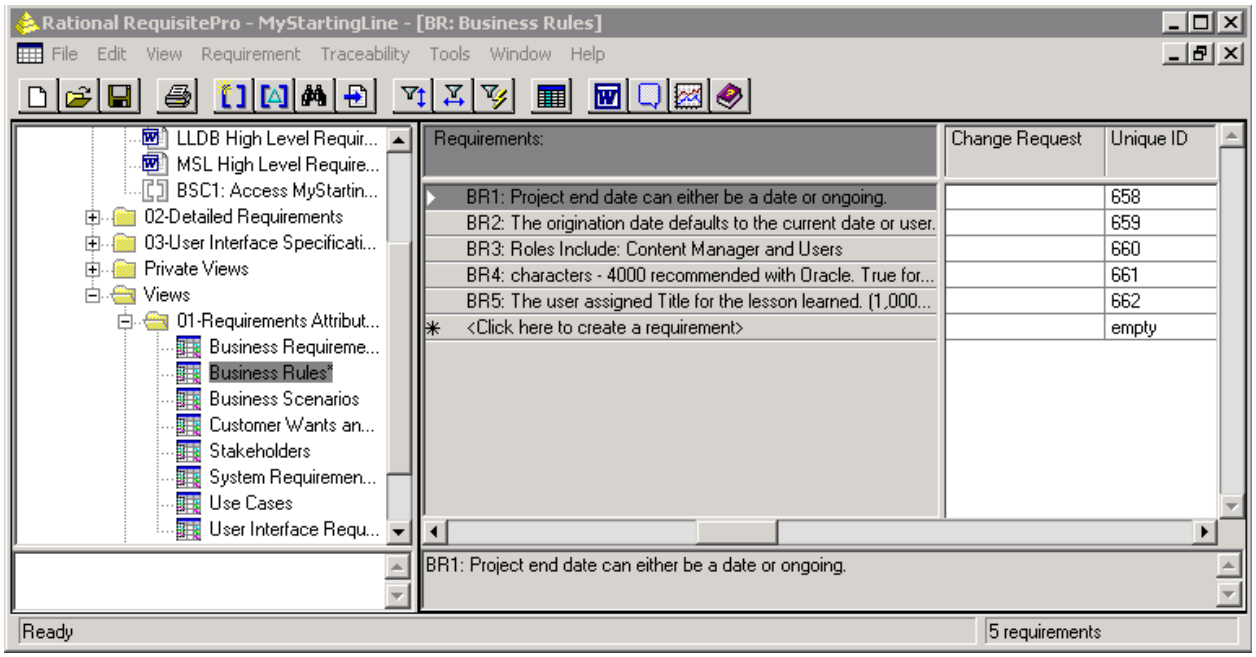


Figure 4-2: RequisitePro Requirements Linked to ClearQuest Records

4.2 ClearQuest to ClearCase

Any change to a file controlled in one of the ClearCase UCM Component VOBs will require that either a CR or a ChildCR be associated with the checkout and checkin operations. That is, when a developer attempts to modify a file stored in a ClearCase Component VOB, s/he will be required to specify either a CR or a ChildCR stored in ClearQuest. Furthermore, that developer must have been assigned to work on the CR for the checkout/checkin operation to succeed.

Upon a successful checkin, the CR will be linked to the version of the file that was created. This approach ensures that all files changed in order to implement a CR are linked back to the CR. By cross referencing the files associated with each CR that was implemented for a given release, it is possible to identify all of the files that were created or changed in order to create that release. This is the standard approach to Activity Management used in the ClearCase UCM Model for software development described in the IBM Rational ClearCase Information Center: https://publib.boulder.ibm.com/infocenter/cchelp/v7r1m0.

Figure 4-3 shows a ClearQuest CR linked to two versioned elements. You will notice that the CR shows not only what files and directories were changed while working on the CR, but also which specific versions of those files and directories were created.

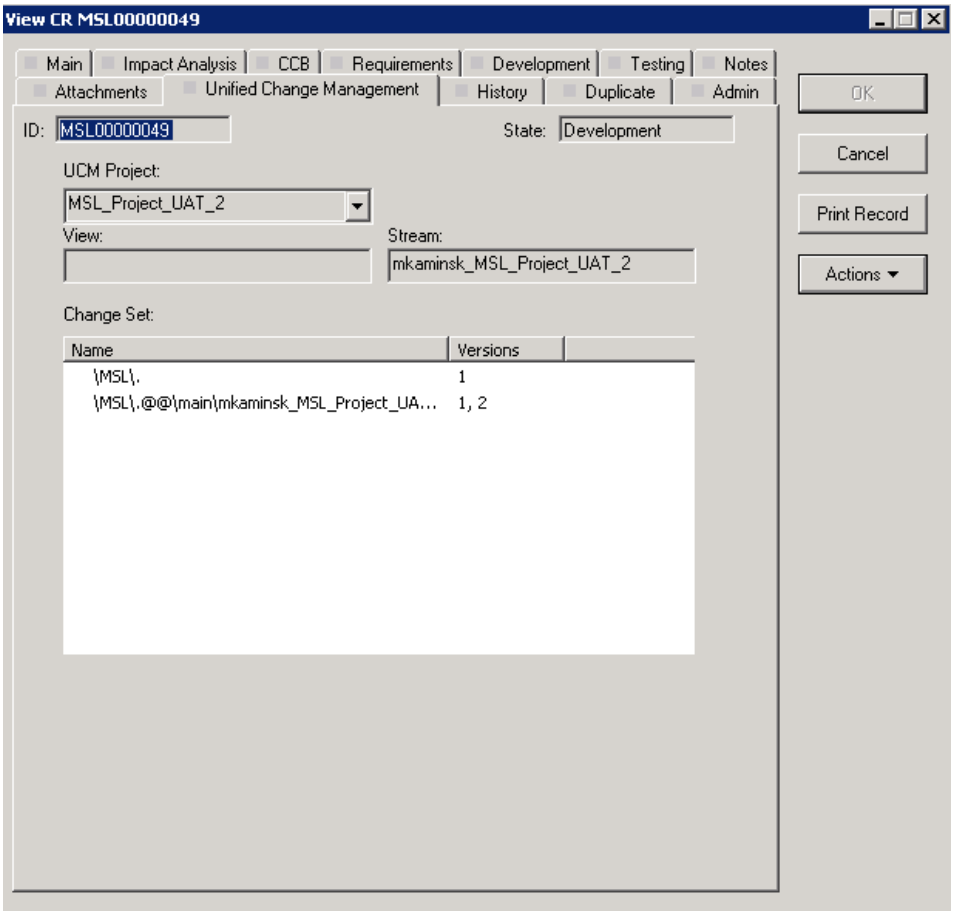


Figure 4-3: CR Change Set

4.3 ClearQuest to Rational Quality Manager

The ClearQuest defect record will be used to document all defects identified during the execution of test cases managed by Rational Quality Manager. When a tester identifies a defect during the execution of a test case, the tester will generate a new ClearQuest based defect record from inside of RQM. RQM will communicate with ClearQuest and present the tester with a Defect submission form via a web interface. The details about the test case being executed will be automatically populated in the detailed description field of the defect. Figure 4-4 shows the submission form for a defect created from RQM.

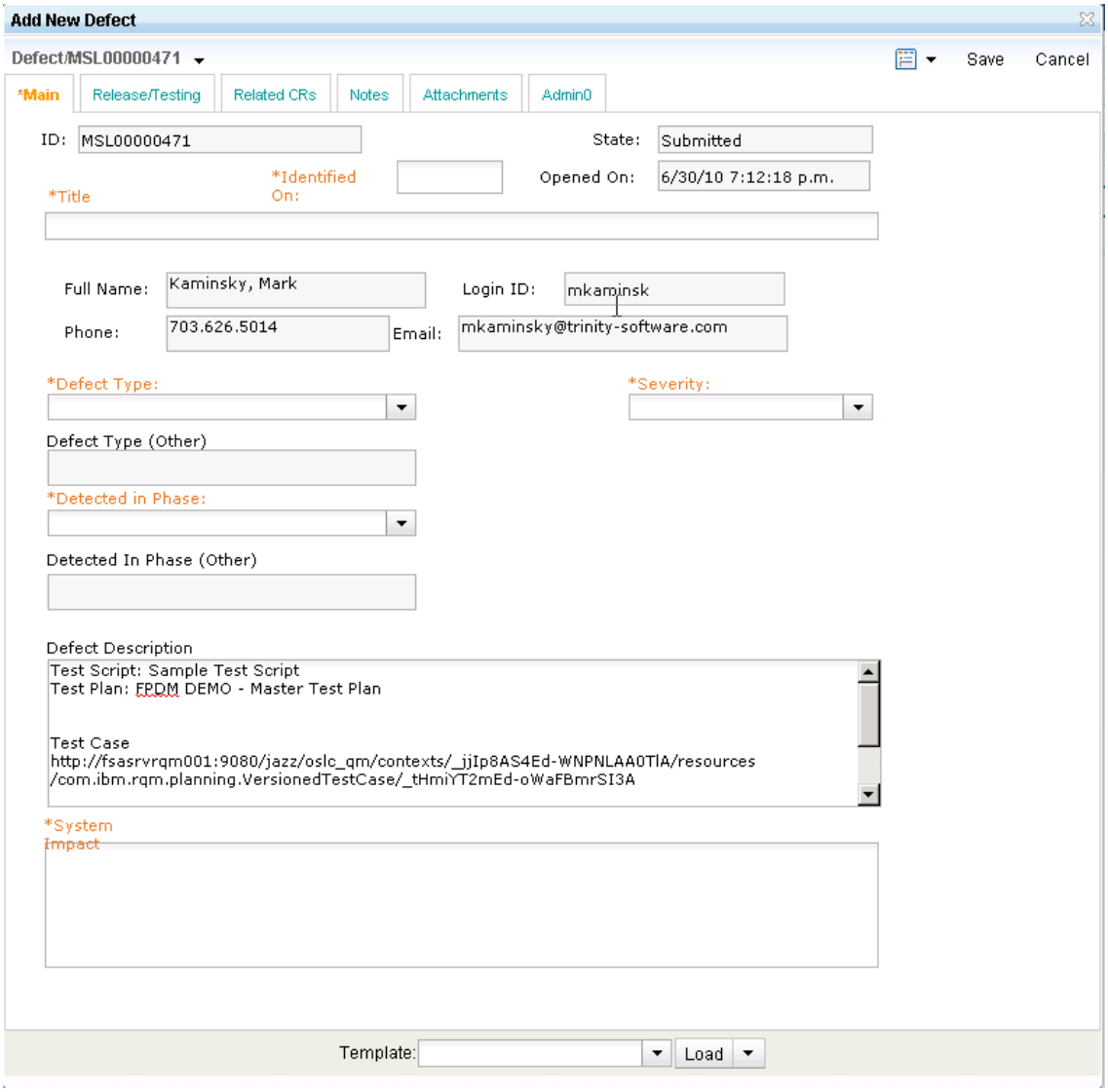


Figure 4-4: Defect Created from RQM

Section 5. Baseline Management

The three types of system baselines defined in the system CM Plan (Functional, Design and, Product) are implemented using a combination of ClearCase labels and ClearCase UCM Baselines.

A base-ClearCase label must be used when a system baseline refers to files stored in a base-ClearCase VOB such as the system's documentation VOB. A ClearCase UCM Baseline is used when the system baseline refers to file elements stored in one or more of the system's software VOBs (ClearCase UCM Component VOBs).

In the case of a Functional Baseline, it is highly unlikely that any file referred to by the baseline would be stored in the system's software VOB. Therefore, a base-ClearCase label would be used to represent the entire Functional Baseline for the system.

On the other hand, a Product Baseline is likely to include files stored in both the system's software VOB and the system's documentation VOB. In this case a base-ClearCase Label will be used to catalog the files in the system's documentation VOB that belong to the product baseline and a ClearCase UCM Baseline will be used to catalog the files contained in the system's software VOB that belong to the product baseline

5.1 Baseline Naming Conventions

Baselines will be named using two letters followed by the numbering of the system release and the version of the baseline:

- Functional Baseline: FB_##.##_v#
- Design Baseline: DB_##.##_v#
- Product Baseline: PB_##.##_v#

5.2 Creating Baselines

A baseline adhering to the conventions above will be created by the CM Lead. While the specific details of creating a baseline may differ from one type of baseline to another based on the artifacts being incorporated into a baseline, the high level process is consistent across all baseline types:

1. Collect the artifacts that are to be included in the baseline.
2. Add those artifacts to ClearCase control.
3. Apply a base-ClearCase label and/or a ClearCase UCM baseline to the artifacts.

The types of artifacts included in functional baselines, design baselines, and product baselines are described in the system's CM Plan.

The majority of the artifacts included in the project's functional baseline will be document-based artifacts managed directly in the system's ClearCase "Documentation" VOB. Since the

"Documentation" VOB will be a "base-ClearCase" VOB rather than a UCM Component VOB, base-ClearCase labels will be used predominantly to represent functional baseline.

5.2.1 Creating a base-ClearCase Label to Represent an FSA Baseline (Functional, Design, or Product)

After all relevant files have been added to the project's ClearCase Documentation VOB, a ClearCase label representing the baseline for the system will be created and applied to the VOB.

The steps required to create a functional baseline in a base-ClearCase VOB such as the system documentation VOB are as follows:

1. The CM Manager will create a label type using either the ClearCase graphical user interface or the command "cleartool mklbtype".
2. The CM Manager will apply the label to all elements in the VOB to which the functional baseline applies using either the "Apply Label Wizard" or the command "cleartool mklabel".

For detailed instructions on creating base-ClearCase labels, refer to either online ClearCase help via the Rational ClearCase Information Center or to the "mklbtype" and "mkbl" commands in the ClearCase Command Reference Guide.

Accessing the Contents of a Baseline in the Documentation VOB (base-ClearCase VOB)

A previously created baseline can be retrieved simply by configuring a "base" ClearCase view's Configuration Specification as follows:

```
element * <baseline name> -nocheckout
element * /main/LATEST -nocheckout
```

A view configured using the single rule above will provide the user with access to the artifacts included in the baseline.

5.2.2 Creating a ClearCase UCM-Baseline to Represent an FSA Baseline (Functional, Design, or Product)

Creating a UCM Baseline in the FSA Enterprise Solution is NO different than creating a UCM baseline anywhere else. From the ClearCase UCM project explorer, the CM Team member will select the stream to which the UCM baseline is to be applied. They will then right-click on the stream and select "create baseline". They will supply the name of the new baseline and click "OK". ClearCase will then create the new UCM baseline and apply it to the elements in the VOB(s).

Refer to Appendix E: Job Aids for an example of how to create a ClearCase baseline.
For detailed instructions on creating UCM baseline, refer to either online ClearCase help via the Rational ClearCase Information Center or to the "mkbl" command in the ClearCase Command Reference Guide.

5.3 Including Artifacts Managed in Other Tools

Several of the artifacts listed above will be managed by Rational tools rather than file-based artifacts stored directly in ClearCase. These artifacts will need to be extracted from the Rational tools and converted to file-based artifacts prior to being incorporated into a baseline.

5.3.1 Including RequisitePro Artifacts in a Baseline

Many of the Requirements documents and records will be managed directly in RequisitePro and will, therefore, need to be extracted, packaged, and added to ClearCase control prior to creating a ClearCase Baseline. This is accomplished using a two-stage approach:

1. Each MS Word-based document stored in the RequisitePro project will be exported to PDF format and checked into the project's "Documentation" VOB.
2. The data stored in the RequisitePro project and associated Oracle database will be extracted using the RequisitePro Baseline Manager and added to ClearCase control. Since the RequisitePro Baseline Manager may only be executed by an administrator, a request must be submitted to the FSA Rational Software Group (RSG) via a FSA Rational Environment Request System (FRERS). The name of the baseline (see section 5.1 of this document) and the System should be included in the FRERS request.

5.3.2 Including Rational Quality Manager Artifacts in a Baseline

While the test plans will be created and managed outside of Rational Quality Manager (RQM), test cases will be managed in RQM and will need to be exported from RQM in order to be added to the project's "Documentation" VOB. All test cases for all phases of testing will be exported to a PDF format using RQM's build in PDF print capability and added to the system's Documentation VOB.

5.4 Archival and Management

Only the last 2 major releases of the system will be maintained. These releases will include any minor releases or patches to the system that may have occurred in between the major releases.

Once a third major release of the system has been implemented, the oldest release of the system will be archived and removed from electronic or manual libraries.

For detailed information about Phasing, Baselines, and Milestones, refer to the FSA Enterprise Configuration Management Plan template and the system-specific Configuration Management Plan.

Section 6. Reporting

The following reports and queries are available in public folders for all users. Examples of each report are provided in Appendix D.

6.1 ClearQuest Queries

The public ClearQuest queries are divided into two high level categories:

- General queries that users will run directly.
- Report based queries which will act as filters for the reports that are run.

6.1.1 General Queries

The queries listed below are made available to all users to run directly or as a basis for creating personal queries. These queries are not linked to any public reports.

- **All CRs**: Lists all CRs sorted by CR ID from most recent CR to oldest CR. The fields presented in the results grid are as follows: ID, State Headline.
- **All ChildCRs**: Lists all ChildCRs sorted by ChildCR ID from most recent CR to oldest CR. The fields presented in the results grid are as follows: ID, State Headline.
- **All Components**: Lists all Component Records sorted by Component Name. The fields presented in the results grid are as follows: Name, Description.
- **All Releases:** Lists all Release Records sorted by Release ID. The only field presented in the result grid is ReleaseID.
- **All Email Rules:** Lists all Email Rules sorted by Email Rule Name. The fields presented in the results grid are as follows: Email Rule Name, Active.

6.1.2 Report Queries (Filters)

The queries listed were developed in order to support the filter criteria associated with the public reports that are provided.

CR Queries

- **CR-CCB Review:** Lists all CRs that need to be reviewed by the CCB.
  Prompt: none
  Filter: CRs in state (Submitted, Initial Review, Impact_Analysis, Hold or CCB).
- **CR-CCB Target Release:** Lists CRs that have been targeted for a release but have not yet been completed.
  Filter: CRs in any state other than Initial-Review, Impact-Analysis and CCB.
  Prompt: Target Release
- **CR-Query for ID:** Prompt the user for the CR ID and filter on it.
  Prompt: CR ID
  Filter: CR matching the ID specified. If no ID is specified, all CRs are listed.
- **CR-Summary:** General purpose CR query based on multiple prompted values.
  Prompt: Found In Release, Target Release, State, Assigned Developer
  Filter: Filters on all prompted values.

- **CR-and-Defect-Summary:** Prompt the user for the the target release and filter's on it.
  Prompt: Release Number
  Filter: Filter's on all prompted values.

ChildCR Queries

- **ChildCR-Query for ID**: Prompt the user for the ChildCR ID and filter on it
  Prompt: ChildCR ID
  Filter: ChildCR matching the ID specified. If no ID is specified then all ChildCRs are listed.
- **ChildCR-Summary:** General purpose ChildCR query based on multiple prompted values.
  Prompt: Parent CRs->Found In Release, Parent CR=>Target Release, State, Assigned Developer.
  Filter: Filters on all prompted values.

6.2 ClearQuest Reports

ClearQuest reports have been developed in order to aid in reviewing CRs and ChildCRs.

6.2.1 CR Reports

- **Configuration Control Board Review**
  Summary: Provides a list of CRs to be reviewed by the CCB
  Filter: Uses query "CR-CCB Review"
  ClearQuest Fields Displayed: ID, Title, Description, Opened On, State, Opened By, Requested By, Requested Priority, Assigned Priority, Category, Found in Release, Target Release, General Impact Analysis, Development Impact, Requirements Impact, and Security Impact.
  Sort Order: State (Submitted, Initial_Review, Impact_Analysis, CCB), Submit Date (oldest first).
- **Configuration Control Board Targeted Release**
  Summary: Provides a list of CRs that have been reviewed by the CCB.
  Filter: Uses query: "CR-CCB Target Release"
  Fields to Display: ID, Title, Description, Opened On, State, Opened By, Requested By, Assigned Priority, Found in Release, Target Release, Complexity, Assigned Developer, Associated Requirements, Associated Components, and Related Change Requests.
  Sort Order: Release number (oldest first)
- **Change Request Summary Report**
  Summary: Short summary of each CR
  Filter: Uses query "CR-Summary"
  Fields to Display: ID, Related CRs, Title, Found In Release, Target Release, Opened On, Assigned Developer, and State.
  Sort Order: Target Release, State, Developer
- **Change Request Detailed Report**
  Summary: Detailed listing of all visible fields in a CR
  Filter: Uses query "CR-Query for ID"

Fields to Display: All visible fields in the CR. Order the fields by Tab on the CR record form.
Sort Order: Target Release (oldest first), Date Submitted (oldest first)

- **Change Requests and Defect Summary**
  Summary: Displays the defects related to change requests for a specified target release.
  Filter: Uses query "CR-and-Defect Summary"
  Fields to Display: CR ID, Title, Component, Developer, Related Defects (Defect ID, Defect Title, Defect State)
  Sort Order: Target Release (oldest first), CR ID (oldest first)

6.2.2 ChildCR Report Formats

- **Child Change Request Summary Report**
  Summary: Short summary of each ChildCR
  Filter: Uses Query "ChildCR-Summary"
  Fields to Display: ID, Related CRs, Title, Found In Release, Target Release, Opened On, Assigned Developer, and State.
  Sort Order: Target Release, State, Developer
- **Child Change Request Detailed Report**
  Summary: Detailed listing of all visible fields in a ChildCR.
  Filter: Uses query "ChildCR-Query for ID"
  Fields to Display: All visible fields in the ChildCR ordered by tab order on the ChildCR record form.
  Sort Order: Target Release (oldest first), Date Submitted (oldest first)

Appendix A: Acronyms and Abbreviations

Appendix A: Acronyms and Abbreviations

Acronym	Definition
ATP	Authorization To Proceed
CCB	Change Control Board
CI	Configuration Identification/Item
CM	Configuration Management
CMRB	Configuration Management Review Board
COTS	Commercial Off The Shelf
CR	Change Request
ECCB	Enterprise Change Control Board
ECM	Enterprise Configuration Management
ECMM	Enterprise Configuration Management Methodology
EOCM	Enterprise Operational Change Management
FCA	Functional Configuration Audit
FSA	Federal Student Aid
GFE	Government Furnished Equipment
ITIL	Information Technology Infrastructure Library
NIST	National Institute of Standards and Technology
PA	Process Audit
PCA	Physical Configuration Audit
PM	Program Manager
QA	Quality Assurance
SCC	Standard Change Control
SE	Systems Engineering
SEI	System Engineering Institute
SR	Service Request
UAT	User Acceptance Test
V&V	Verification and Validation
VDC	Virtual Data Center

Appendix B: Glossary

Appendix B: Glossary

Term	Definition
Baseline	A Benchmark used as a reference point. For example: An ITSM Baseline can be used as a starting point to measure the effect of a Service Improvement Plan. A Performance Baseline can be used to measure changes in Performance over the lifetime of an IT Service. A Configuration Management Baseline can be used to enable the IT Infrastructure to be restored to a known Configuration if a Change or Release fails.
Capability Maturity Model Integration (CMMi)	Capability Maturity Model® Integration (CMMi) is a process improvement approach developed by the Software Engineering Institute (SEI) of Carnegie Melon University. CMMI provides organizations with the essential elements of effective processes. It can be used to guide process improvement across a project, a division, or an entire organization. CMMI helps integrate traditionally separate organizational functions, set process improvement goals and priorities, provide guidance for quality processes, and provide a point of reference for appraising current processes.
Change	The addition, modification or removal of anything that could have an effect on IT Services. The Scope should include all IT Services, Configuration Items, Processes, Documentation etc.
Change Management	The Process responsible for controlling the Lifecycle of all Changes. The primary objective of Change Management is to enable beneficial Changes to be made, with minimum disruption to IT Services.
Change Control Board (CCB)	Board that makes decisions regarding whether or not proposed changes to a software project should be implemented. The change control board is constituted of project stakeholders or their representatives. The authority of the change control board may vary from project to project, but decisions reached by the change control board are often accepted as final and binding.
Configuration	A generic term, used to describe a group of Configuration Items that work together to deliver an IT Service, or a recognizable part of an IT Service. Configuration is also used to describe the parameter settings for one or more CIs.
Configuration Identification (CI)	The Activity responsible for collecting information about Configuration Items and their Relationships, and loading this information into the CMDB. Configuration Identification is also responsible for labeling the CIs themselves, so that the corresponding Configuration Records can be found.
Configuration Item (CI)	Any Component that needs to be managed in order to deliver an IT Service. Information about each CI is recorded in a Configuration Record within the Configuration Management System and is maintained throughout its Lifecycle by Configuration Management. CIs are under the control of Change Management. CIs typically include IT Services, hardware, software, buildings, people, and formal documentation such as Process documentation and SLAs
CI Type	A Category that is used to Classify CIs. The CI Type identifies the required Attributes and Relationships for a Configuration Record. Common CI Types include: hardware, Document, User etc.

Term	Definition
Configuration management (CM)	The Process responsible for maintaining information about Configuration Items required delivering an IT Service, including their Relationships. This information is managed throughout the Lifecycle of the CI. Configuration Management is part of an overall Service Asset and Configuration Management Process. Configuration management (CM) is a field of management that focuses on establishing and maintaining consistency of a product's performance and its functional and physical attributes with its requirements, design, and operational information throughout its life. For information assurance, CM can be defined as the management of security features and assurances through control of changes made to hardware, software, firmware, documentation, test, test fixtures, and test documentation throughout the life cycle of an information system.
Configuration Management Data Base (CMDB)	A database used to store Configuration Records throughout their Lifecycle. The Configuration Management System maintains one or more CMDBs, and each CMDB stores Attributes of CIs, and Relationships with other CIs. A configuration management database (CMDB) is a repository of information related to all the components of an information system. Although repositories similar to CMDBs have been used by IT departments for many years, the term CMDB stems from ITIL (Information Technology Infrastructure Library). In the ITIL context, a CMDB represents the authorized configuration of the significant components of the IT environment. A key goal of a CMDB is to help an organization understand the relationships between these components and track their configuration. The CMDB is a fundamental component of the ITIL framework's Configuration Management process. CMDB implementations often involve integration with other systems, such as Asset Management Systems. These integrations may make use of either a real-time, federated design or an ETL (extract, transform, and load) solution.
Custom Off The Shelf (COTS)	Application software or Middleware that can be purchased from a Third Party.
Enterprise Change Control Board (ECCB)	The official Federal Student Aid committee authorized to review, approve or reject enterprise operational changes. The ECCB oversees the Enterprise Operational Change Management (EOCM) process and supporting tools and reviews the Enterprise Master Release Schedule.
Enterprise Operational Change Management (EOCM)	The EOCM process is an enterprise-level process that operates in parallel to system/application specific CCBs within Federal Student Aid. This process includes clearly defined intersection points with a generic life cycle to ensure that communication and coordination of Enterprise Events occur across Federal Student Aid
Functional Configuration Audit (FCA)	A functional configuration audit ensures that functional and performance attributes of a configuration item are achieved
Information Technology Infrastructure Library (ITIL)	The Information Technology Infrastructure Library (ITIL) is a set of concepts and policies for managing information technology (IT) infrastructure, development and operations. ITIL is published in a series of books, each of which covers an IT management topic. The names ITIL and IT Infrastructure Library are registered trademarks of the United Kingdom's Office of Government Commerce (OGC). ITIL gives a detailed description of a number of important IT practices with comprehensive checklists, tasks and procedures that can be tailored to any IT organization.

Term	Definition
National Information Standards Technology (NIST)	A measurement standards laboratory which is a non-regulatory agency of the United States Department of Commerce. The institute's mission is to promote U.S. innovation and industrial competitiveness by advancing measurement science, standards, and technology in ways that enhance economic security and improve quality of life.
Physical Configuration Audit (PCA)	A physical configuration audit ensures that a configuration item is installed in accordance with the requirements of its detailed design documentation.
Process Audit (PA)	Ensures that all required configuration management processes were correctly executed and appropriate process documentation has been maintained.
Program Management (PM)	The process of managing multiple interdependent projects that lead towards an improvement in an organization's performance.
Quality Assurance (QA)	The Process responsible for ensuring that the Quality of a product, Service or Process will provide its intended Value.
Service Management (SM)	Service Management is a set of specialized organizational capabilities for providing value to customers in the form of services.
Service Request (SR)	A request from a User for information, or advice, or for a Standard Change or for Access to an IT Service. For example to reset a password, or to provide standard IT Services for a new User. Service Requests are usually handled by a Service Desk, and do not require an RFC to be submitted. See Request Fulfillment.
Standard Change Control (SCC)	Formal process used to ensure that changes to a product or system are introduced in a controlled and coordinated manner. It reduces the possibility that unnecessary changes will be introduced to a system without forethought, introducing faults into the system or undoing changes made by other users of software.
System Engineering (SE)	An interdisciplinary field of engineering that focuses on how complex engineering projects should be designed and managed.
System Engineering Institute (SEI)	The Carnegie Mellon Software Engineering Institute (SEI) is a federally funded research and development center headquartered on the campus of Carnegie Mellon University in Pittsburgh, Pennsylvania, United States. SEI also has offices in Arlington, Virginia, and Frankfurt, Germany. The SEI operates with major funding from the U.S. Department of Defense. The SEI also works closely with industry and academia through research collaborations. The SEI program of work is conducted in three principal areas: management practices, engineering practices, and acquisition practices.
User Acceptance Test (UAT)	Process to obtain confirmation by a Subject Matter Expert (SME), preferably the owner or client of the object under test, through trial or review, that the modification or addition meets mutually agreed-upon requirements. In software development, UAT is one of the final stages of a project and often occurs before a client or customer accepts the new system.

Term	Definition
Verification and Validation (V&V)	Process of checking that a software system meets specifications and that it fulfills its intended purpose. It is normally part of the software testing process of a project.

Appendix C: ClearQuest Record Details

Appendix C: Change Request Record Details

CR Record Type Forms

This section describes the various forms, tabs, and fields associated with the Change Request (CR) record type.

There are two distinct forms for CRs. The C-Submit form provides a condensed set of tabs and fields required to open a new CR. The CR-Record form contains the expanded set of tabs and fields used throughout the rest of the CR workflow.

The CR Submit Form contains the following tabs:

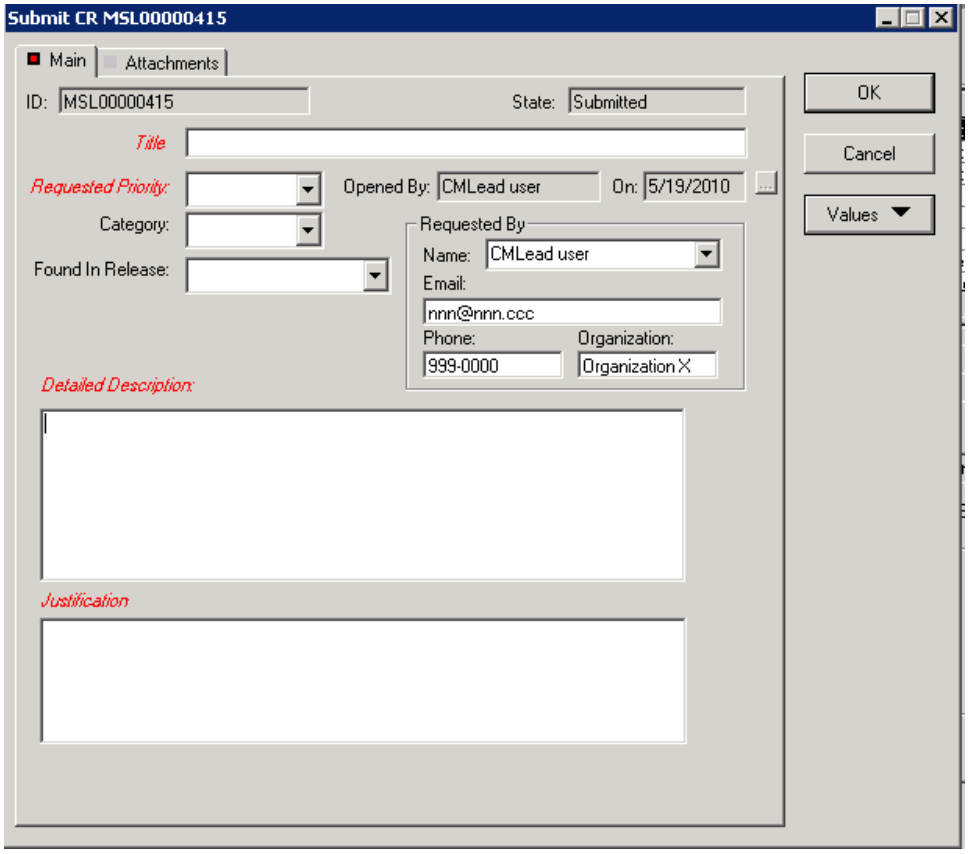
- Main: The information required to submit a new CR is visible on this tab.
- Attachments: Files may be attached to the new CR via the tab.

The CR Record Form contains the following tabs:

- Main: Contains most of the information related to the original submittal of the CR
- Impact Analysis: Contains the impact analysis documentation from the requirements team, development team and security team
- CCB: Contains the CCB disposition for the CR
- Requirements: Contains the list of requirements that relate to the CR
- Development: Contains development related information
- Testing: Contains testing related information
- Notes: General note entry and date/time stamped notes log
- Attachments: Contains references to files that have been attached to the CR
- Unified Change Management: Contains references to files that have been modified in response to the CR
- History: Contains a list of actions and state changes for the CR

The following images represent the as-built ClearQuest Change Request record type.

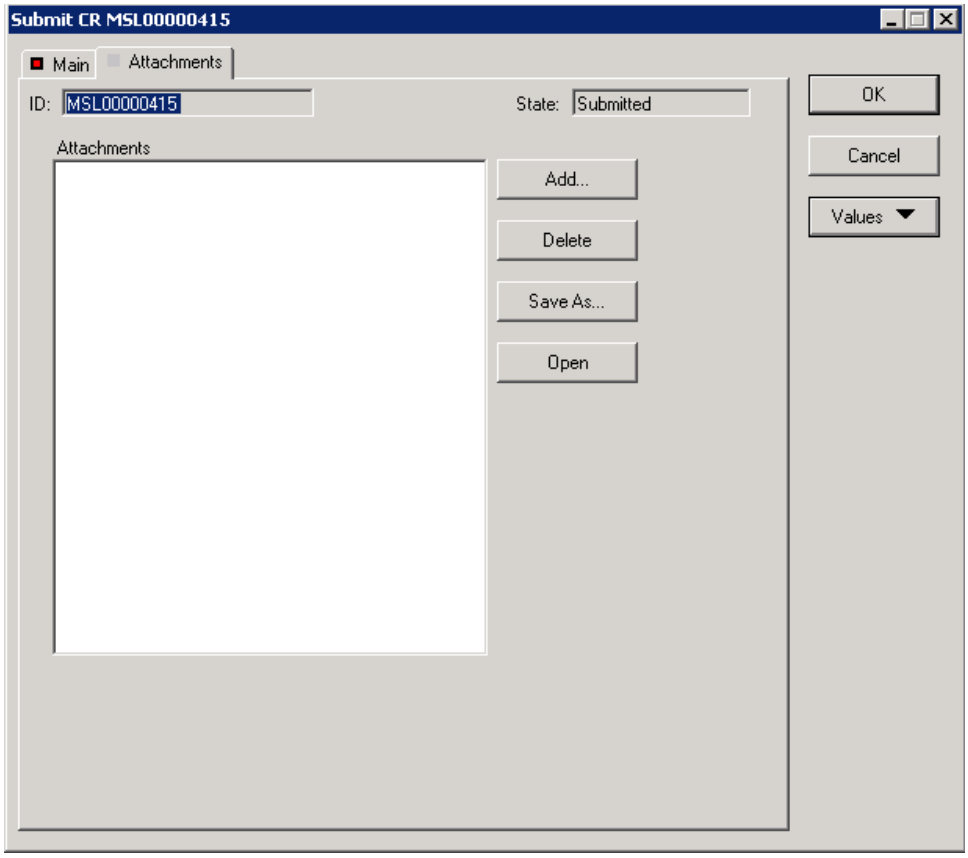
CR Submit (Main Tab)



Display Name	Description	Field Value Choices
ID	CR's identifier	Auto-generated by ClearQuest
State	CR's state/status	Auto-populated and updated by ClearQuest
Title	Short description of the CR	Name
Requested Priority	Requested priority of the CR	Routine Urgent Emergency
Category	A means of categorizing the CR	Correction Enhancement Maintenance
Found In Release	The release of the application that the CR is being written against.	Selected from a list of in-active release records. NOTE: an inactive release is one that is NOT

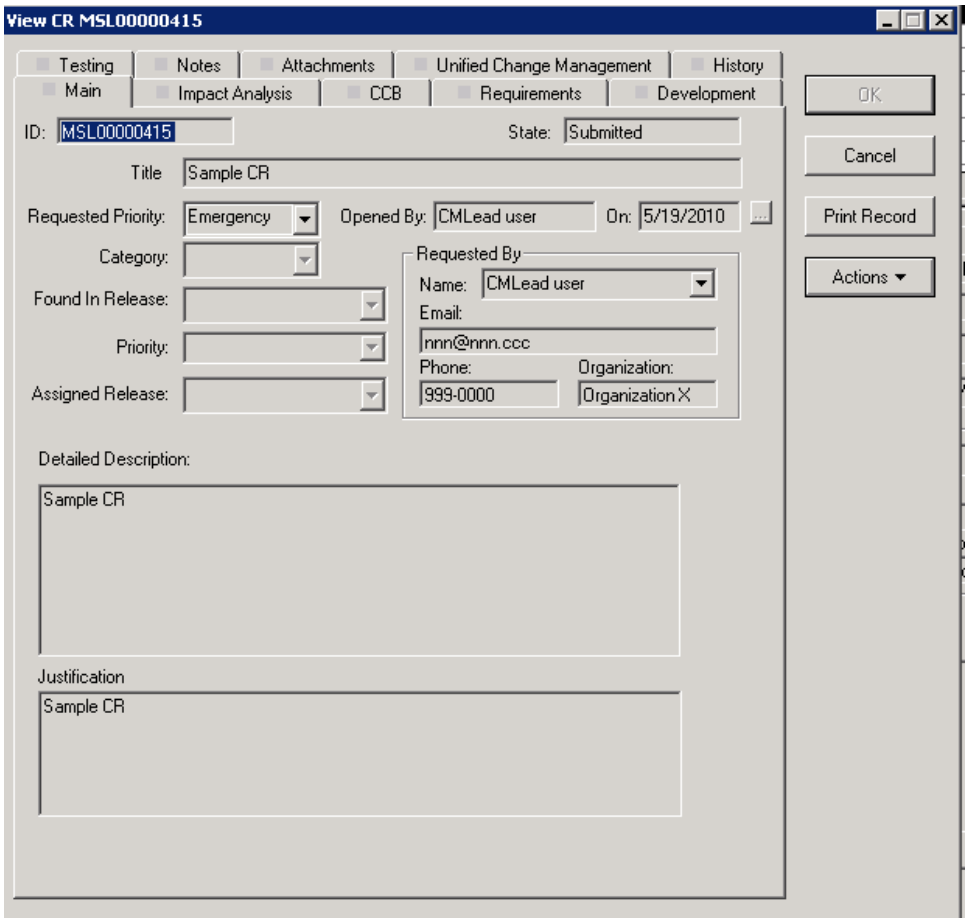
Display Name	Description	Field Value Choices
		currently under development.
Detailed Description	Multi-line, detailed description of the change being requested.	Free-form text
Justification	Reason(s) why the implementation of the CR will improve the FSA environment	Free-form text
Opened By	Person that entered the CR into ClearQuest	Auto-populated by ClearQuest
Opened On	Date that the CR was entered into ClearQuest	Auto-populated by ClearQuest
Name (Requested By)	Name of the person requesting the CR	Free-form text
Phone (Requested By)	Phone number of the person requesting the CR	Free-form text auto-populated based on Requested By Name
Organization (Requested By)	Organization that the person belongs to requesting the CR	Free-form text
Email (Requested By)	Email address of the person requesting the CR	Free-form text

CR Submit (Attachment tab)



Display Name	Description	Field Value Choices
ID	CR's identifier	Auto-generated by ClearQuest
State	CR's state/status	Auto-populated and updated by ClearQuest
Attachments	External files that can be attached to the CR. NOTE: The files will be stored directly in the DB.	n/a

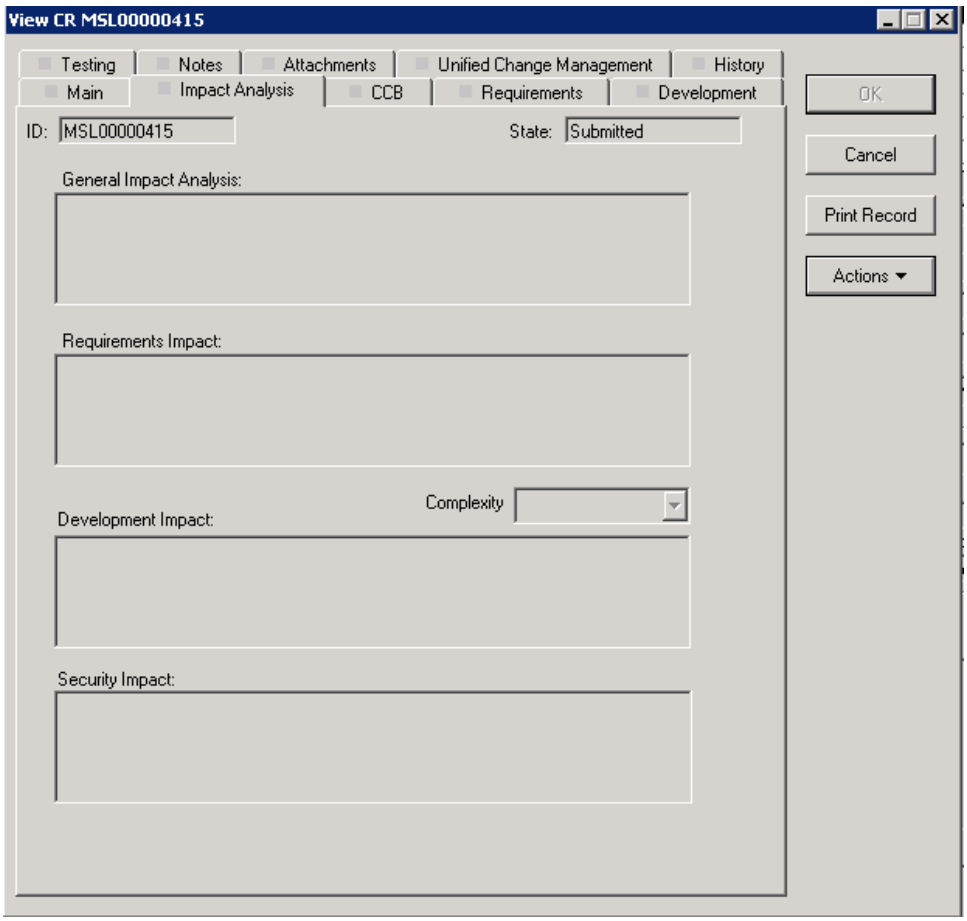
6.2.2.1 CR (Main Tab)



Display Name	Description	Field Value Choices
ID	CR's identifier	Auto-generated by ClearQuest
State	CR's state/status	Auto-populated and updated by ClearQuest
Title	Short description of the CR	Free-form text
Requested Priority	The priority of the CR from the requester's perspective	Routine Urgent Emergency
Category	A means of categorizing the CR	Correction Enhancement Maintenance
Found In Release	The release of the application that the CR is being written against.	Selected from a list of in-active release records.

Display Name	Description	Field Value Choices
		NOTE: an inactive release is one that is NOT currently under development
Priority	The assigned priority of the CR	Routine Urgent Emergency
Assigned Release	The release that the CR is targeted to be implemented in.	Selected from a list of active release records. NOTE: an active release is one that IS currently under development
Detailed Description	Multi-line, detailed description of the change being requested.	Free-form text
Justification	Reason(s) why the implementation of the CR will improve the FSA environment	Free-form text
Opened By	Person that entered the CR into ClearQuest	Auto-populated by ClearQuest
Opened On	Date that the CR was entered into ClearQuest	Auto-populated by ClearQuest
Name (Requested By)	Name of the person requesting the CR	Free-form text
Phone (Requested By)	Phone number of the person requesting the CR	Free-form text auto-populated based on Requested By Name
Organization (Requested By)	Organization that the person belongs to requesting the CR	Free-form text
Email (Requested By)	Email address of the person requesting the CR	Free-form text auto-populated based on Requested By Name

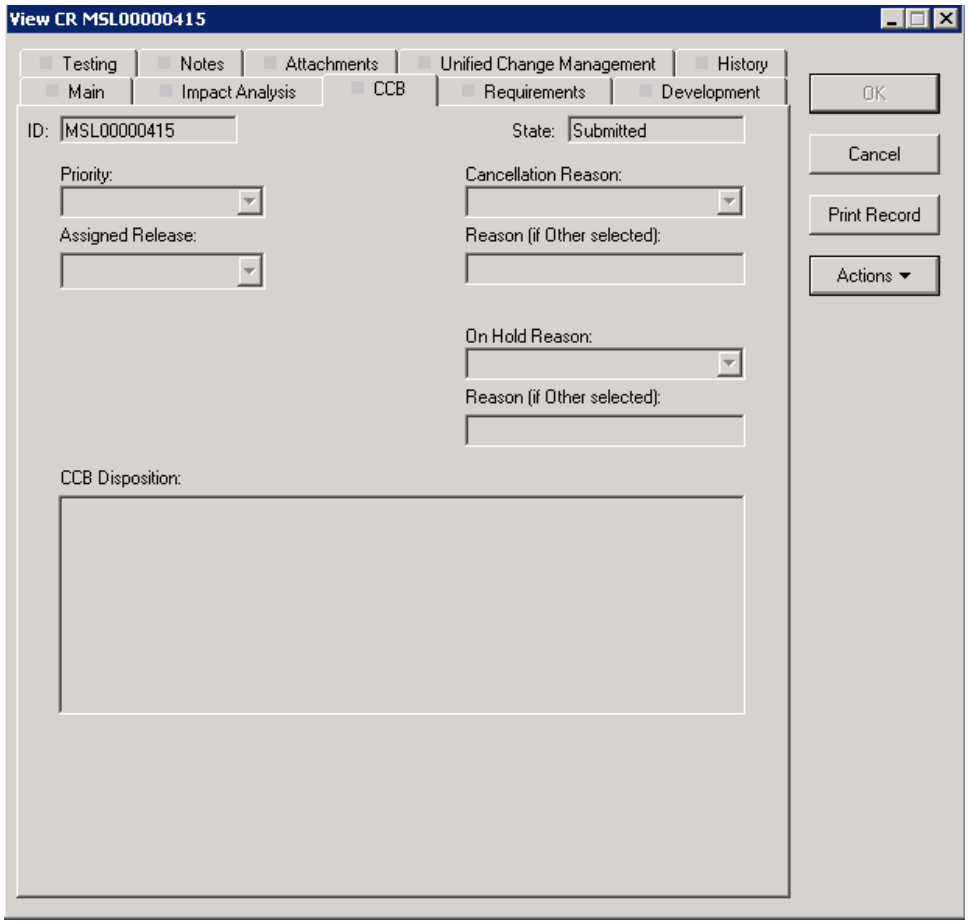
CR (Impact Analysis tab)



Display Name	Description	Field Value Choices
ID	CR's identifier	Auto-generated by ClearQuest
State	CR's state/status	Auto-populated and updated by ClearQuest
General Impact Analysis	General comments related to the impact of the CR on the system	Free-form text
Requirements Impact	Impact of implementing the CR to the system from the perspective of the requirements team	Free-form text
Development Impact	Impact of implementing the CR to the system from the perspective of the development team.	Free-form text
Security Impact	Impact of implementing the CR to the system from the perspective of the security team.	Free-form text

Display Name	Description	Field Value Choices
Complexity	Estimated complexity of the CR.	Unknown Low Medium High

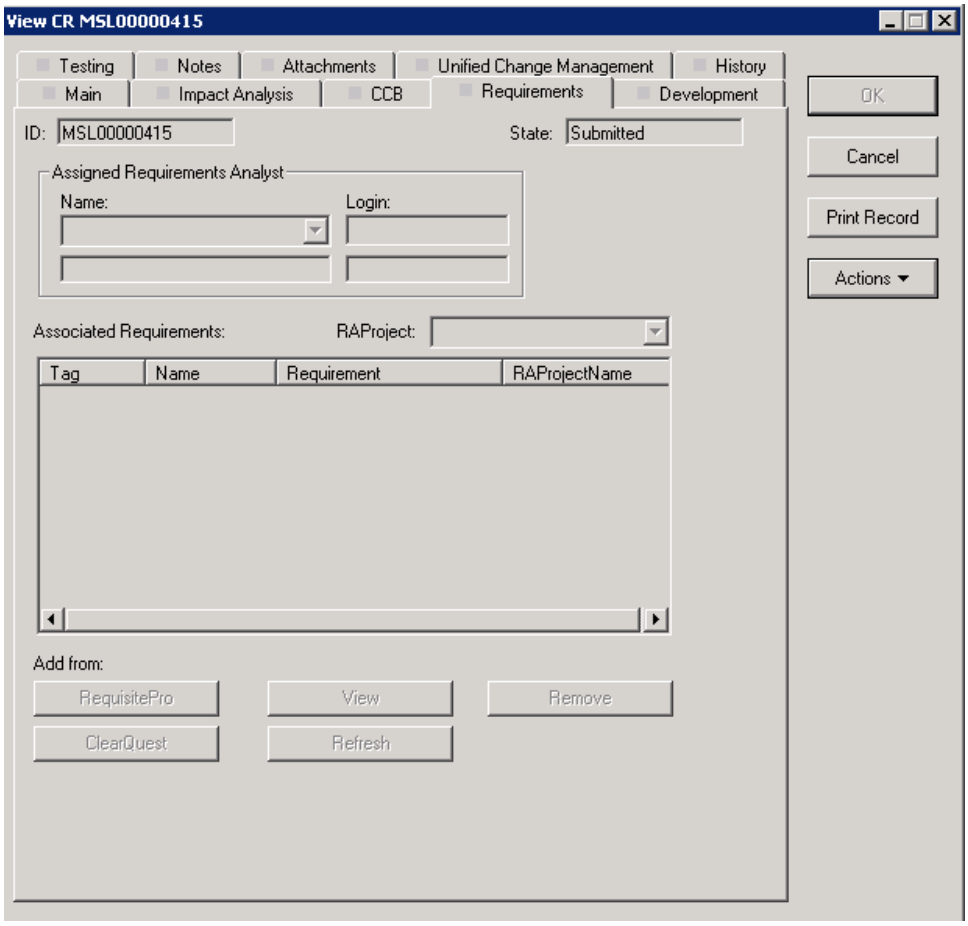
CR (CCB tab)



Display Name	Description	Field Value Choices
ID	CR's identifier	Auto-generated by ClearQuest
State	CR's state/status	Auto-populated and updated by ClearQuest

Display Name	Description	Field Value Choices
Cancellation Reason	A short reason for why the CR was cancelled	Duplicate Not Technically Feasible Incompatible with System's Mission Out of Scope for System Other
Reason (if Other selected)	If the cancellation reason chosen was "Other" then this field will be required	Free from text
Priority	Priority of the CR specified by the CCB. NOTE: This may be different than the requested priority entered when the CR was first submitted	Routine Urgent Emergency
Assigned Release	The release of the system that the CR will be included/implemented in	A list of all active release records for the system.
On Hold Reason	Reason that the CR has been placed on-hold	Requires Funding Requires Contract Changes Requires Targeted Release Other
Reason (if Other selected)	If the on-hold reason chosen was "Other" then this field will be required	Free-form text
CCB Disposition	The disposition of the CR based on a review by the CCB	Free-form text

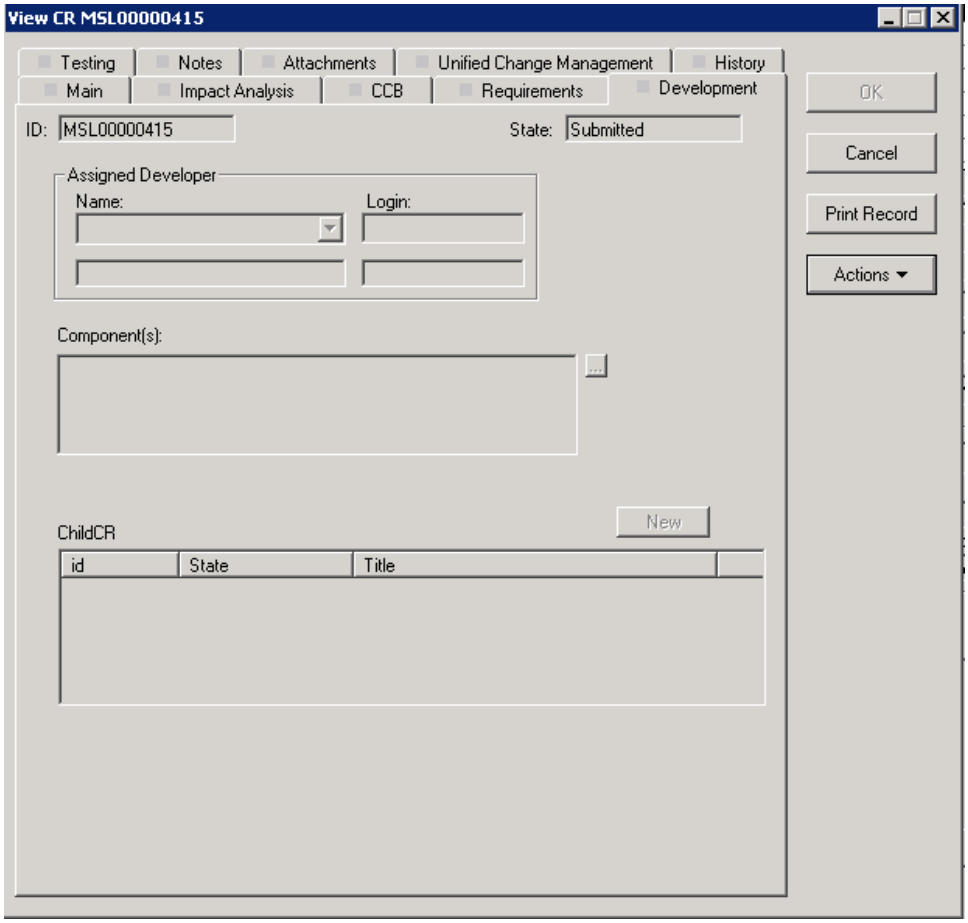
CR (Requirements Tab)



Display Name	Description	Field Value Choices
ID	CR's identifier	Auto-generated by ClearQuest
State	CR's state/status	Auto-populated and updated by ClearQuest
Name (Assigned Requirements Analyst)	The name of the requirements analyst assigned to the CR	List of all project team members in the Requirements-Team group
Email (Assigned Requirements Analyst)	The email address of the requirements analyst chosen	Auto-populated based on the name chosen for the assigned requirements analyst
Phone (Assigned Requirements Analyst)	The phone number of the requirements analyst chosen	Auto-populated based on the name chosen for the assigned requirements analyst
Login (Assigned	The login id of the requirements analyst	Auto-populated based on

Display Name	Description	Field Value Choices
Requirements Analyst)	chosen	the name chosen for the assigned requirements analyst
RAProject	The “Rational Administrator Project” where the requirements will be pulled from. NOTE: This field is required in order to associate the record with a Requisite Pro database. It may be possible to configure ClearQuest such that this field is populated automatically.	List of available Rational Administrator Projects that have been configured to be usable by the ClearQuest database. NOTE: Typically, only a single Rational Administrator Project would be associated with a ClearQuest database.
Associated Requirements	0 or more requirements located in the Requisite Pro project that is associated with the “Rational Administrator” project specified in the “RAProject” field.	All requirements in the related Requisite Pro project.

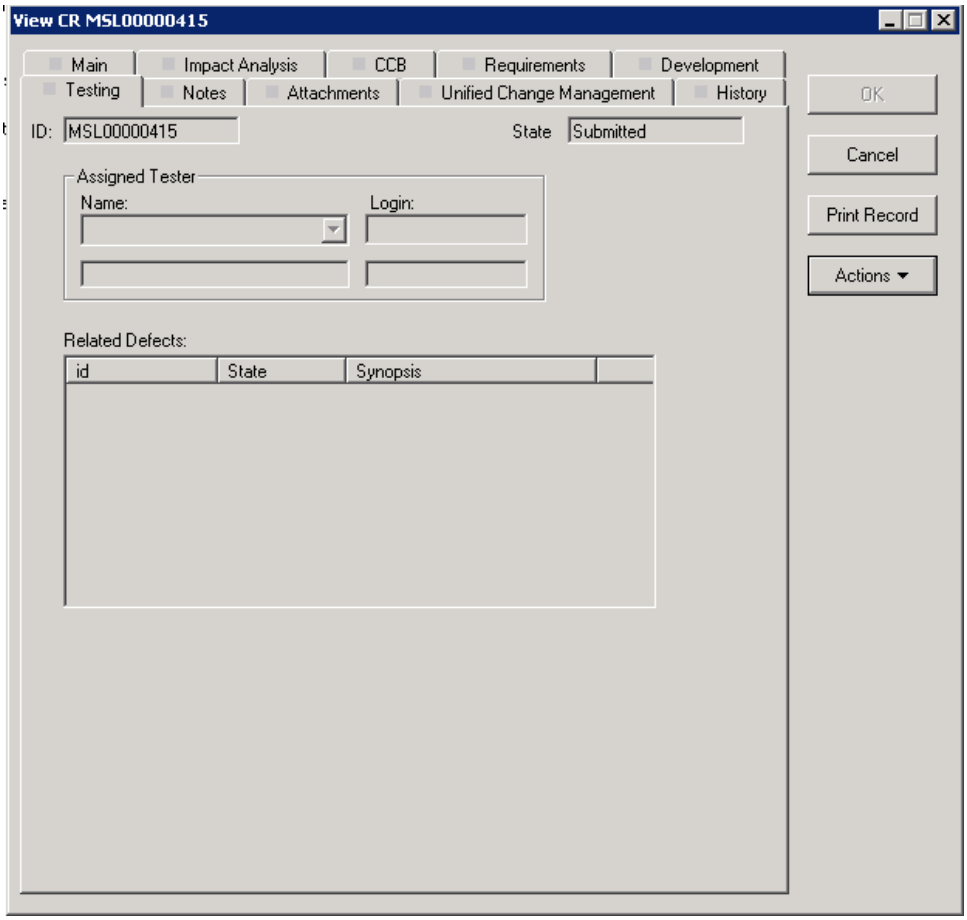
CR (Development Tab)



Display Name	Description	Field Value Choices
ID	CR's identifier	Auto-generated by ClearQuest
State	CR's state/status	Auto-populated and updated by ClearQuest
Name (Assigned Developer)	The name of the developer assigned to the CR	List of all project team members in the Requirements-Team group
Email (Assigned Developer)	The email address of the developer chosen	Auto-populated based on the name chosen for the assigned requirements analyst
Phone (Assigned Developer)	The phone number of the developer chosen	Auto-populated based on the name chosen for the

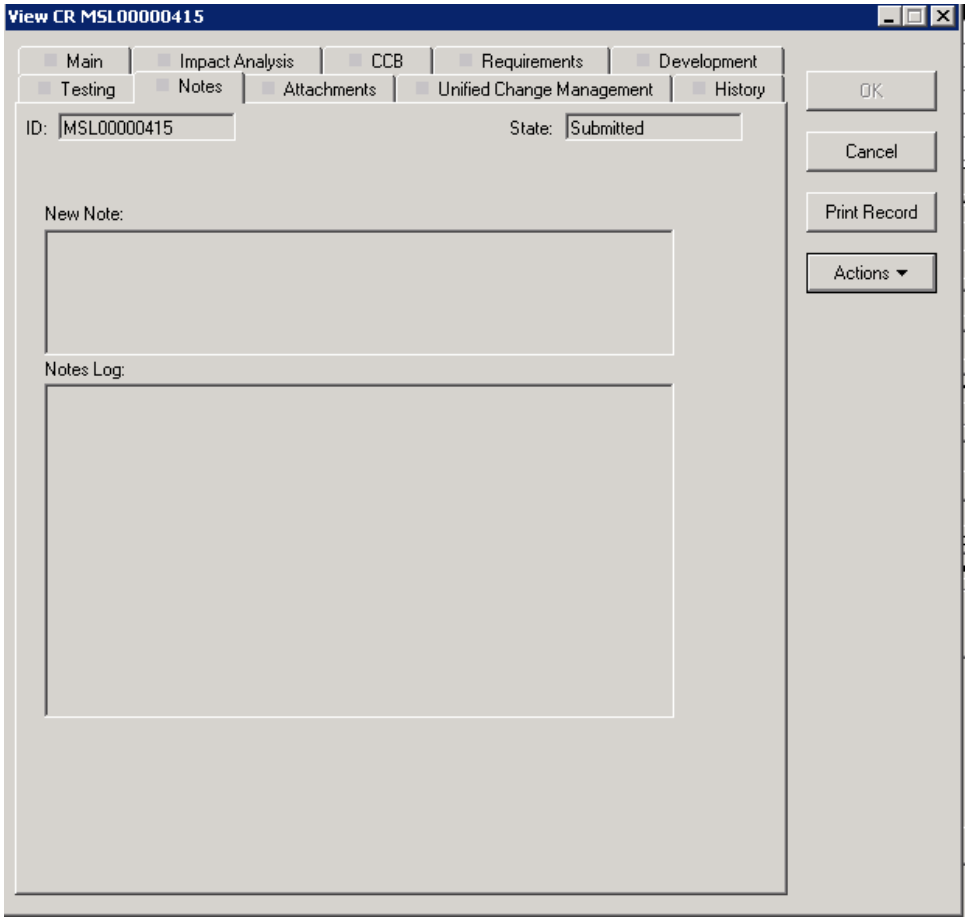
Display Name	Description	Field Value Choices
		assigned requirements analyst
Login (Assigned Developer)	The login id of the developer chosen	Auto-populated based on the name chosen for the assigned requirements analyst
Components	List of system components that the CR is related to	All active components associated with the system
ChildCR	ChildCRs that have been created in order to subdivide the work associated with implementing the CR	Nothing is selected. New ChildCRs are created.

CR (Testing Tab)



Display Name	Description	Field Value Choices
ID	CR's identifier	Auto-generated by ClearQuest
State	CR's state/status	Auto-populated and updated by ClearQuest
Name (Assigned Tester)	The name of the tester assigned to the CR	List of all project team members in the Requirements-Team group
Email (Assigned Tester)	The email address of the tester chosen	Auto-populated based on the name chosen for the assigned requirements analyst
Phone (Assigned Tester)	The phone number of the tester chosen	Auto-populated based on the name chosen for the assigned requirements analyst
Login (Assigned Tester)	The login id of the tester chosen	Auto-populated based on the name chosen for the assigned requirements analyst
Related Defects	List of defects identified when testing the CR.	Defects will NOT be associated with the CR from this screen. Defects will be associated with a CR from the Defect form. This field will list the defects that have previously been associated with the CR

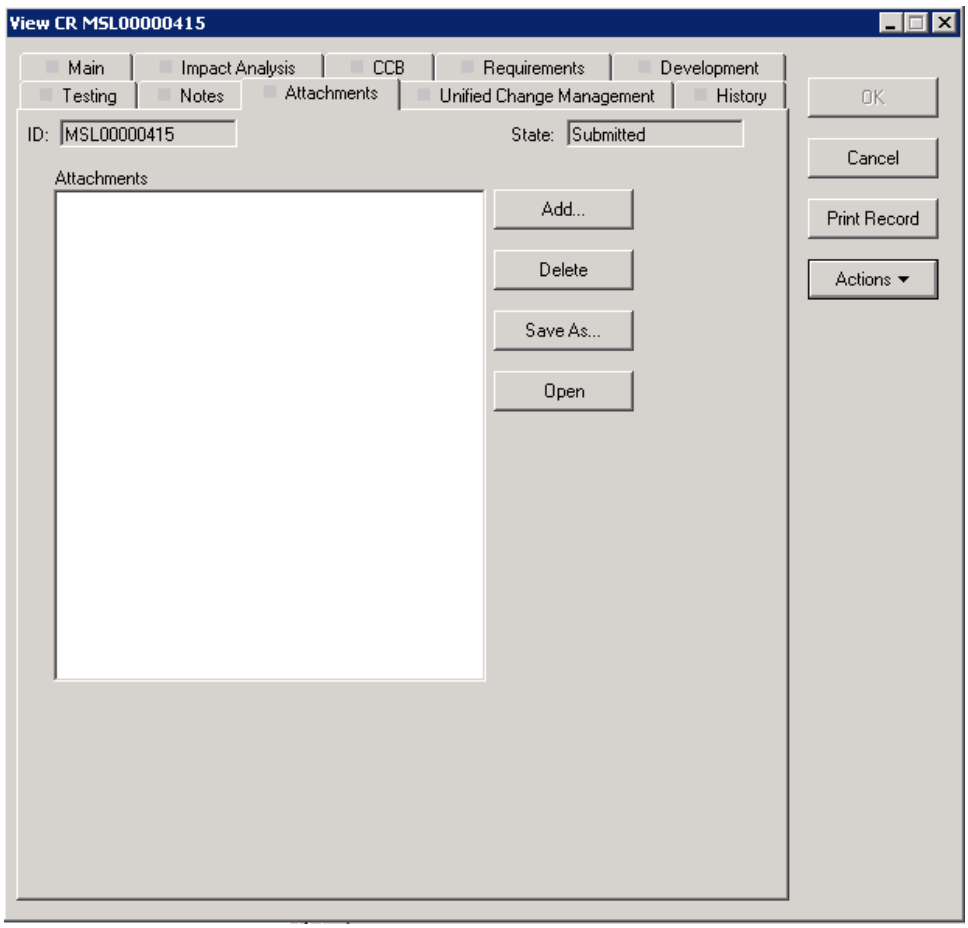
CR (Notes Tab)



Display Name	Description	Field Value Choices
ID	CR's identifier	Auto-generated by ClearQuest
State	CR's state/status	Auto-populated and updated by ClearQuest
New Note	A new note entry	Free-form text
Notes Log	A date/time stamped log of all note entries associated with the CR. For each note entry, the following information is captured: State that the CR is in The login id of the user that entered the note Current Date	Auto-populated by ClearQuest based on the New Note

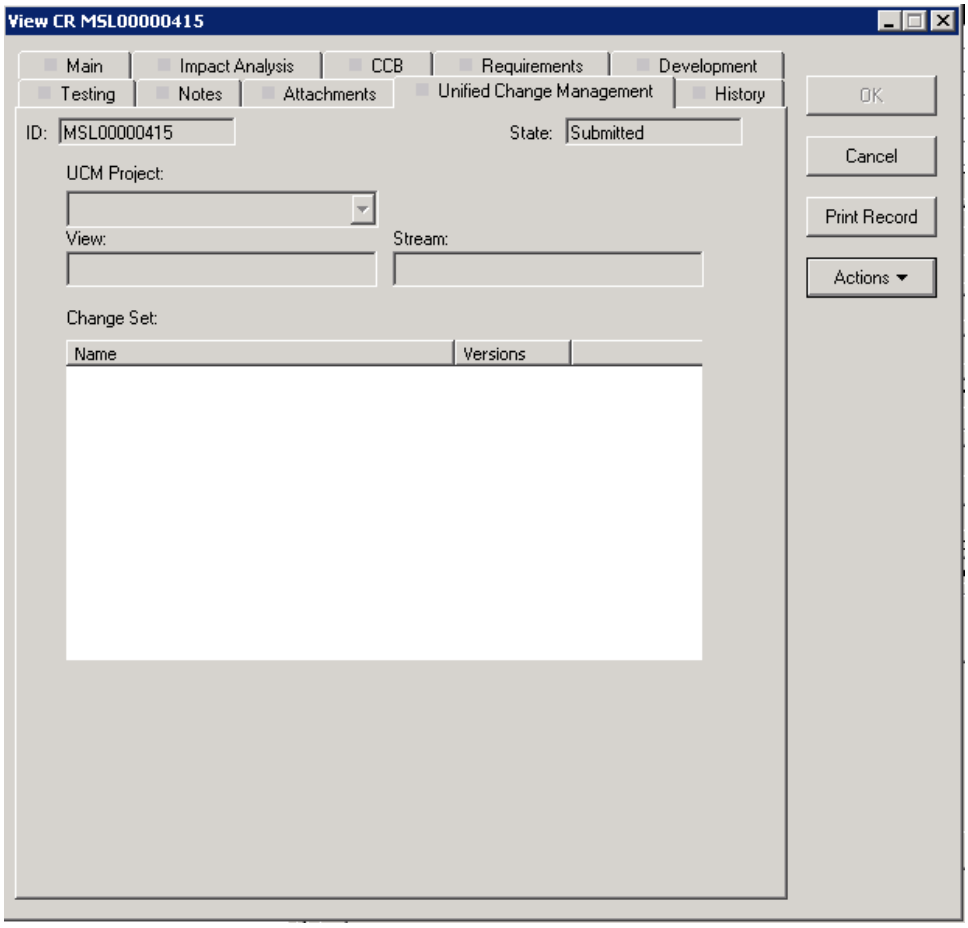
Display Name	Description	Field Value Choices
	Current Time	

CR (Attachment Tab)



Display Name	Description	Field Value Choices
ID	CR's identifier	Auto-generated by ClearQuest
State	CR's state/status	Auto-populated and updated by ClearQuest
Attachments	External files that can be attached to the CR. NOTE: The files will be stored directly in the DB.	n/a

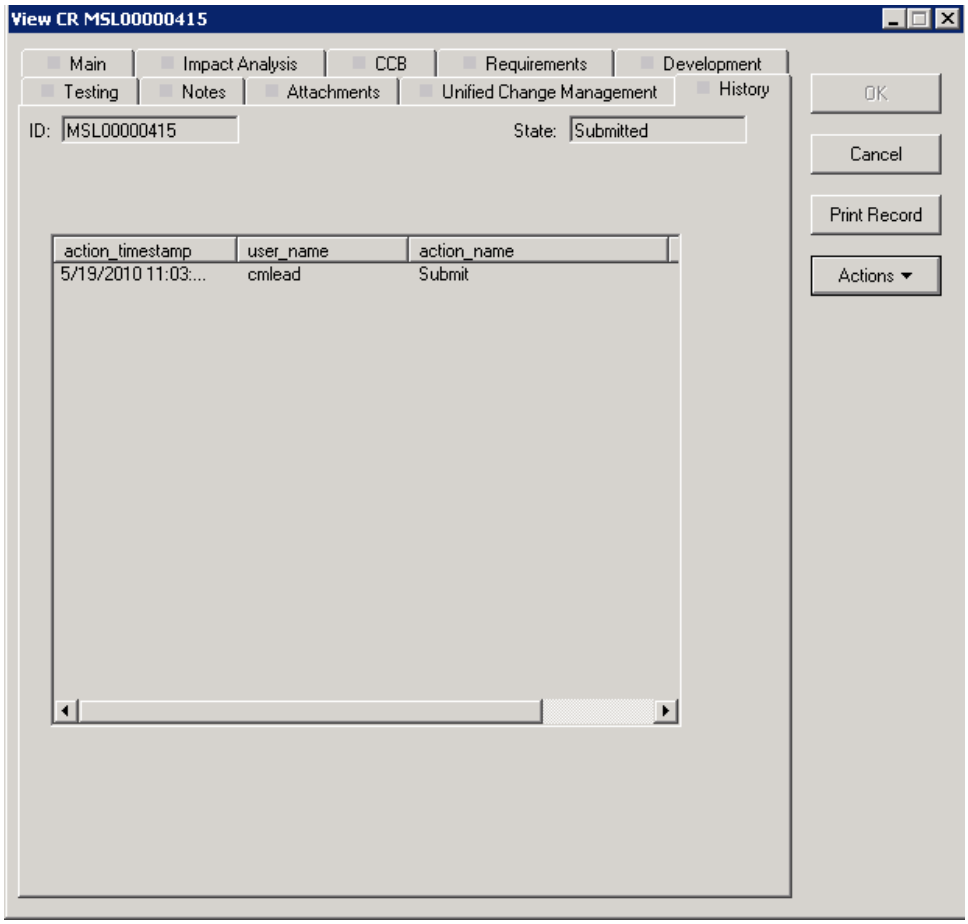
CR (Unified Change Management Tab)



Display Name	Description	Field Value Choices
ID	CR's identifier	Auto-generated by ClearQuest
State	CR's state/status	Auto-populated and updated by ClearQuest
UCM Project	The UCM project that will be associated with the CR. At the time that a CR is assigned to a developer to work on, the UCM project will be chosen. Typically, only a single UCM project will be available and it will auto-populate this field.	List of active ClearCase UCM projects associated with the ClearQuest database.
View	The ClearCase view associated with the CR	When a developer checks out a file in ClearQuest and associates the file with the CR, this field is auto-populated.

Display Name	Description	Field Value Choices
Stream	The ClearCase stream (branch) that this CR will be worked on	When a developer checks out a file in ClearQuest and associates the file with the CR, this field is auto-populated.
Change Set	The list of files and directories that were modified while working on the CR. For each file and directory, each version created is listed	Auto-populated based when files are checked in using the CR.

CR (History Tab)



Display Name	Description	Field Value Choices
ID	CR's identifier	Auto-generated by ClearQuest
State	CR's state/status	Auto-populated and updated by ClearQuest
History	A list of actions and state transitions for the CR	Auto-populated based on actions

Appendix D: ClearQuest Report Samples

Appendix D: ClearQuest Report Samples

If viewing the electronic version of this document, the reports below can be expanded into full PDF documents by double clicking anywhere on the report.

Change Request Summary Report

MyStartingLine Change Request Summary **Federal Student Aid**

As of: 06/17/10

ID	Related CRs	Title	Found In Release	Target Release	Opened On	Assigned Developer	State
00000404		Add support for up to 4 shipping addresses	01.00.004		05/14/2010		Impact_Analysis
00000450		test			06/15/2010		Submitted
00000402		Test 1			05/12/2010		Submitted
00000400		Test	01.00.004	00.00.000	05/09/2010		Ready_Development
00000061		test	01.05.000	01.01.000	03/08/2010	Shah, Maulikkumar	Development
00000301		Troubleshoot empvadmin's access to additional func		01.01.001	12/22/2009	Kaminsky, Mark	Development
00000306		Separate WCM design items into the Design library	00.00.000	01.01.001	01/11/2008	Kaminsky, Mark	Development
00000307		Redesign Executive Dashboard Page	01.05.000	01.01.001	02/26/2010	RSG Test Account	Development
00000299		Review access privilege of all MSL authors	01.01.001	01.01.001	01/22/2010	Shah, Maulikkumar	Development
00000330	00000423 00000424	Author reported inability to move content to arch.		01.01.001	01/20/2010	Shah, Maulikkumar	Development
00000298		Separate sample content library	01.01.001	01.01.001	01/31/2008	DevTeam user1	Implementation
00000292		Fix Inconsistency in Opening New Windows	01.00.004	01.01.001	11/22/2008	Shah, Maulikkumar	Implementation
00000403		Add support for up to 2 shipping addresses	01.01.000	01.03.000	05/14/2010	DevTeam user1	First_Live_Batch
00000415	00000416	Sample CRsgsdfgsdfgsfdgsfgsgggsdfggfgfgsdfgsdgsdfgsdfgsdfg sdfgsdfgsdfgsdfgsdfgsdfgssabcdefghijklmnopqrstuvwxy zmark mark mark	01.04.000	01.04.000	05/19/2010	DevLead user	Closed
00000405	00000406 00000407	Add support for up to 5 shipping addresses	01.06.000	01.04.000	05/14/2010	DevTeam user2	Development
00000016		test	01.04.001	01.04.001	02/10/2010	CMLead user	Development
00000049		4.4 Test Case	01.04.010	01.04.001	03/05/2010	Kaminsky, Mark	Development
00000012	00000015	Testing 4	01.04.001	01.04.001	02/08/2010	Shah, Maulikkumar	First_Live_Batch
00000082		Test	01.04.001	01.04.001	03/12/2010	Shah, Maulikkumar	Implementation
00000039		mkaminsky test	01.04.001	01.04.001	03/03/2010		Ready_Development
00000001		Just a test	01.01.000	01.04.010	01/27/2010	Betteann Sturrup	Development
00000349	00000383 00000384 00000423	Just a test	01.04.001	01.04.010	03/18/2010	Kaminsky, Mark	Development
00000313		Remove unneeded presentation/authoring templates	01.01.001	01.05.000	08/11/2008	Kaminsky, Mark	Development
00000041	00000053	Testing UAT Round#2	01.04.001	01.05.000	03/05/2010	Shah, Maulikkumar	Development
00000059		working again on the 4.8 error	01.01.000	01.05.000	03/08/2010	Shah, Maulikkumar	Development
00000008		Testing 3	01.04.001	01.05.000	02/08/2010	Shah, Maulikkumar	Development

1 of 2

Change Request Report

MyStartingLine Change Request **Federal Student Aid**

As of: 06/16/10

Change Request Number: 00000426

Title: massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel
State: Development
Requested Priority: Routine
Priority: Urgent
Category: Correction
Found In Release: 01.01.000
Target Release: 01.06.000

Opened By: Kaminsky, Mark **On:** 06/03/2010
Requested By: Kaminsky, Mark
Email: mkaminsky@trinity-software.com
Phone: 703.626.5014
Organization: CIO

Description:
Sed congue tortor sit amet lorem consequat semper pretium felis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Quisque nec massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel ullamcorper convallis, nisi sapien convallis ante, id viverra magna augue in risus. Praesent in diam massa. Sed congue tortor sit amet lorem consequat semper pretium felis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Quisque nec massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel ullamcorper convallis, nisi sapien convallis ante, id viverra magna augue in risus. Praesent in diam massa. Sed congue tortor sit amet lorem consequat semper pretium felis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Quisque nec massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel ullamcorper convallis, nisi sapien convallis ante, id viverra magna augue in risus. Praesent in diam massa. Sed congue tortor sit amet lorem consequat semper pretium felis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Quisque nec massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel ullamcorper convallis, nisi sapien convallis ante, id viverra magna augue in risus. Praesent in diam massa. Sed congue tortor sit amet lorem consequat semper pretium felis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Quisque nec massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel ullamcorper convallis, nisi sapien convallis ante, id viverra magna augue in risus. Praesent in diam massa. Sed congue tortor sit amet lorem consequat semper pretium felis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Quisque nec massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel ullamcorper convallis, nisi sapien convallis ante, id viverra magna augue in risus. Praesent in diam massa. Sed congue tortor sit amet lorem consequat semper pretium felis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Quisque nec massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel ullamcorper convallis, nisi sapien convallis ante, id viverra magna augue in risus. Praesent in diam massa. Sed congue tortor sit amet lorem consequat semper pretium felis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Quisque nec massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel ullamcorper convallis, nisi sapien convallis ante, id viverra magna augue in risus. Praesent in diam massa. Sed congue tortor sit amet lorem consequat semper pretium felis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Quisque nec massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel ullamcorper convallis, nisi sapien convallis ante, id viverra magna augue in risus. Praesent in diam massa. Sed congue tortor sit amet lorem consequat semper pretium felis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Quisque nec massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel ullamcorper convallis, nisi sapien convallis ante, id viverra magna augue in risus. Praesent in diam massa. Sed congue tortor sit amet lorem consequat semper pretium felis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Quisque nec massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel ullamcorper convallis, nisi sapien convallis ante, id viverra magna augue in risus. Praesent in diam massa. Sed congue tortor sit amet lorem consequat semper pretium felis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Quisque nec massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel ullamcorper convallis, nisi sapien convallis ante, id viverra magna augue in risus. Praesent in diam massa. Sed congue tortor sit amet lorem consequat semper pretium felis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Quisque nec massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel ullamcorper convallis, nisi sapien convallis ante, id viverra magna augue in risus. Praesent in diam massa. Sed congue tortor sit amet lorem consequat semper pretium felis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Quisque nec massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel ullamcorper convallis, nisi sapien convallis ante, id viverra magna augue in risus. Praesent in diam massa. Quisque nec massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel ullamcorper convallis, nisi sapien convallis ante, id viverra magna augue in risus. Praesent in diam massa. Sed congue tortor sit amet lorem consequat semper pretium felis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Sed congue tortor sit amet lorem consequat semper pretium felis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Quisque nec massa neque. In hac habitasse platea dictumst. Fusce

1 of 7

Configuration Control Board Review Report

MyStartingLine Configuration Control Board Review **Federal Student Aid**

As of: 06/10/10

ID	Title
00000338	sample defect from rqm

Description:
sample from rqm

Opened On: 03/16/2010
State: Impact_Analysis
Opened By: Kaminsky, Mark
Requested By: Kaminsky, Mark
Requested Priority: Emergency
Assigned Priority: Routine
Category: Correction
Found In Release: 01.06.000
Target Release: 01.05.000

Impact Analysis

General Impact Analysis:

Requirements Impact:
Testing UCM integration

Development Impact:
Testing UCM Integration

Security Impact:
Testing UCM Integration

00000404	Add support for up to 4 shipping addresses

Description:
The current release of the application only supports one shipping address that is different from the billing address. Suggest that the same customer may wish to send purchases to multiple destinations

Opened On: 05/14/2010
State: Impact_Analysis
Opened By: DevTeam user2
Requested By: DevTeam user2
Requested Priority: Routine
Assigned Priority:
Category: Enhancement
Found In Release: 01.00.004
Target Release:

Impact Analysis

General Impact Analysis:

Requirements Impact:
zzzzz

Development Impact:
zzzzz

Security Impact:
zzzzz

1 of 2

Configuration Control Board Targeted Release Report

MyStartingLine Configuration Control Board Targeted Release **Federal Student Aid**

As of: 06/16/10

ID	Title
00000057	Add support for up to 5 shipping addresses

Description:
The current release of the application only supports one shipping address that is different from the billing address. Suggest that the same customer may wish to send purchases to up to 5 multiple destinations

Opened On: 03/08/2010
State: Ready_Sys_Testing
Opened By: TestTeam user2
Requested By: TestTeam user2

Assigned Priority: Urgent
Found In Release: 01.01.000
Target Release: 01.06.000

Complexity: Unknown
Assigned Developer: DevTeam user1

Associated Requirements:
Associated Components:
Related Change Requests:

00000312 Link to a JSP on thestartingline site

Description:
A user is trying to access a jsp page from thestartingling.gov website.

The Data Analysis link on PCNet no longer works. I've been relying on this when I need help with PEPS queries. When I clicked on the data analysis link today, it took me here:

http://thestartingline.ed.gov/INTRANETWebApp/FSAnetApps/down.jsp?URI=%22application_eligibility_delivery/school_eligibility/data_analysis/%22&QUERY=%22%22

Opened On: 12/29/2009
State: Implementation
Opened By:
Requested By: Rong Xu

Assigned Priority: Urgent
Found In Release: 01.04.010
Target Release: 01.06.000

Complexity: Medium
Assigned Developer: RSG Test Account

Associated Requirements:
Associated Components:
Related Change Requests:

1 of 3

Child Change Request Summary Report

MyStartingLine Child Change Request Summary **Federal Student Aid**

As of: 06/16/10

ID	Related CR	Title	Found In Release	Target Release	Opened On	Assigned Developer	State
00000423	00000330	Update the Help Text		01.01.001		Kaminsky, Mark	Sys_Testing
00000407	00000405	A new ChildCR	01.06.000	01.04.000		DevTeam user1	Closed
00000416	00000415	Sample Child CR	01.04.000	01.04.000		Kaminsky, Mark	Closed
00000406	00000405	Son of CR 406	01.06.000	01.04.000		DevTeam user2	Development
00000022	00000019	Testing 3.14	01.04.001	01.04.001		CM Team user	Ready_Sys_Testing
00000383	00000349	Update the graphical user interface	01.04.001	01.04.010		DevTeam user1	Development
00000384	00000349	Apply changes to the backend Oracle database	01.04.001	01.04.010		DevTeam user2	Development
00000423	00000349	Update the Help Text	01.04.001	01.04.010		Kaminsky, Mark	Sys_Testing
00000053	00000041	Do X	01.04.001	01.05.000		DevTeam user1	Development
00000428	00000426	06/03/10: Another CR #2	01.01.000	01.06.000		Betteann Sturrup	Closed
00000432	00000426	06/03/10: Sample for testing reports	01.01.000	01.06.000		Shah, Maulikkumar	Closed
00000427	00000426	6/3/10: Child CR Sample	01.01.000	01.06.000		Kaminsky, Mark	Development
00000425	00000319	testing another one	00.00.000	01.06.000		Kaminsky, Mark	Development
00000451	00000426	testing submit date	01.01.000	01.06.000	06/16/2010	Kaminsky, Mark	Development
00000437	00000426	Test	01.01.000	01.06.000		Betteann Sturrup	UAT

Child Change Request Total: 14

1 of 1

Child Change Request Report

MyStartingLine Child Change Request **Federal Student Aid**

As of: 06/16/10

Child Change Request Number: 00000427

Title: 6/3/10: Child CR Sample
State: Development
Description:
massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel Quisq
ue nec massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel

ullamcorper convallis, nisi sapien convallis ante, id viverra magna augue in risus.
Praesent in diam massa. Sed congue tortor sit amet lorem consequat semper pretium f
elis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Sed congue tortor sit amet lorem consequat semper pretium felis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Quisque nec massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel ullamcorper convallis, nisi sapien convallis ante, id viverra magna augue in risus. Praesent in diam massa. Sed congue tortor sit amet lorem Quisque nec massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel ullamcorper convallis, nisi sapien convallis ante, id viverra magna augue in risus. Praesent in diam massa. Sed congue tortor sit amet lorem consequat semper pretium felis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Sed congue tortor sit amet lorem consequat semper pretium felis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Quisque nec massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel ullamcorper convallis, nisi sapien convallis ante, id viverra magna augue in risus. Praesent in diam massa. Sed congue tortor
sit amet lorem massa neque. In hac habitasse platea dictumst. Fusce sempe
r,
massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel Q
uisque nec massa neque. In hac
habitasse platea dictumst. Fusce semper, nunc vel ullamcorper convallis,
nisi sapien convallis ante, id viverra magna augue in risus. Praesent in diam massa. Sed congue tortor sit amet lorem consequat semper pretium felis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Sed congue tortor sit amet lorem consequat semper pretium felis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Quisque nec massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel ullamcorper convallis, nisi sapien convallis ante, id viverra magna augue in risus. Praesent in diam massa. Sed congue tortor sit amet lorem Quisque nec massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel ullamcorper convallis, nisi sapien convallis ante, id viverra magna augue in risus. Praesent in diam massa. Sed congue tortor sit amet lorem consequat semper pretium felis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Sed congue tortor sit amet lorem consequat semper pretium felis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Quisque nec massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel ullamcorper convallis, nisi sapien convallis ante, id viverra magna augue in risus. Praesent in diam massa. Sed congue tortor sit amet lorem massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel Quisque nec massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel ullamcorper convallis, nisi sapien convallis ante, id viverra magna augue in risus. Praesent in diam massa. Sed congue tortor sit amet lorem consequat semper pretium felis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Sed congue tortor sit amet lorem consequat semper pretium felis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Quisque nec massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel ullamcorper convallis, nisi sapien convallis ante, id viverra magna augue in risus. Praesent in diam massa. Sed congue tortor sit amet lorem massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel Quisque nec massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel ullamcorper convallis, nisi sapien convallis ante, id viverra magna augue in risus. Praesent in diam massa. Sed congue tortor sit amet lorem consequat semper pretium felis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Sed congue tortor sit amet lorem consequat semper pretium felis tempus. Praesent tempus, dui quis viverra mattis, magna turpis ultricies sapien, sit amet sagittis ligula sem nec sapien. Quisque nec massa neque. In hac habitasse platea dictumst. Fusce semper, nunc vel ullamcorper convallis, nisi sapien convallis ante, id viverra magna augue in risus. Praesent in diam massa. Sed congue tortor sit

1 of 2

Change Requests and Defect Summary

MyStartingLine Change Request and Defect Summary **Federal Student Aid**

As of: 08/09/10

ID	Title	Release	Component	Developer	State
00000300	Limit authors to specific site areas in authoring	00.00.000		Vasudeva Karthik	Hold
00000400	Test	00.00.000			Ready_Development
00000477	sk-test2	00.00.000		Kim, Scott	Development
00000450	test	01.00.004	MSL IFrame Portlet MSL Profile Portlet MSL Quickr Integration MSL Short URLs	Kaminsky, Mark	Admin-Central-State
00000476	sk-test	01.00.004		Kim, Scott	Development
00000061	test	01.01.000		Shah, Maulikkumar	Development
00000292	Fix Inconsistency in Opening New Windows	01.01.001		Shah, Maulikkumar	Implementation
00000298	Separate sample content library	01.01.001		DevTeam user1	Implementation
00000299	Review access privilege of all MSL authors	01.01.001		Shah, Maulikkumar	Development
00000301	Troubleshoot empvadmin's access to additional func	01.01.001		Kaminsky, Mark	Development
00000306	Separate WCM design items into the Design library	01.01.001		Kaminsky, Mark	Development
00000307	Redesign Executive Dashboard Page	01.01.001		RSG Test Account	Development
00000330	Author reported inability to move content to arch.	01.01.001		Shah, Maulikkumar	Development
00000494	This is a sample CR for testing UCM integration	01.01.001	MSL Bookmarks MSL Employee Finder	Kaminsky, Mark	Development
00000403	Add support for up to 2 shipping addresses	01.03.000		DevTeam user1	First_Live_Batch
00000405	Add support for up to 5 shipping addresses	01.04.000		DevTeam user2	Development
00000415	Sample CRsgsdfgsdfgsfdgsfgsgggsdfggfgfgsdfgsdgsdfgsdfgsdf gsdfgsdfgsdfgsdfgsdfgsdfgssabcdefghijklmnopqrstuvwx yzmark mark mark	01.04.000		DevLead user	Closed
00000012	Testing 4	01.04.001		Shah, Maulikkumar	First_Live_Batch
00000016	test	01.04.001		CMLead user	Development
00000019	Testing UCM	01.04.001		Shah, Maulikkumar	Development
00000039	mkaminsky test	01.04.001			Ready_Development
Related Defects: 00000035	Ready_Development	mkaminsky test			
00000049	4.4 Test Case	01.04.001		Kaminsky, Mark	Development

1 of 4

Appendix E: Security

Appendix E: Security

This appendix provides additional security details relating to Rational Quality Manager, ClearQuest and ClearCase.

RequisitePro Security

For every RequisitePro system repository, the following groups and associated privileges have been defined:

Group	Privileges
Configuration Management Team	Read access to all requirements assets located in the RequisitePro repository
Development Team	Read access to all requirements assets located in the RequisitePro repository
FSA Management	Read access to all requirements assets located in the RequisitePro repository
Operations Team	Read access to all requirements assets located in the RequisitePro repository
Requirements Team	**Read and write** access to all requirements assets located in the RequisitePro repository
Security Team	Read access to all requirements assets located in the RequisitePro repository
Test Team	Read access to all requirements assets located in the RequisitePro repository

ClearQuest Security

ClearQuest records are transitioned from one state to another by various actions. In each state, users perform actions such as "Modify" to change the contents of a record or by moving it to another state. The actions menu lists the actions that can be performed on the record while it is in a given state, including assigning, modifying, and opening and resolving change requests.

Change Request Actions and Permission

The table below lists the actions which transition a change request (CR) from one state to another and the users that are permitted to perform the action.

Action	Permission	Description
Submit	Any project team member	MSL members use this action to create a record or request for change.
Approve_Initial_Review	CMTeam	The CM Team starts the preliminary review of the CR.

Action	Permission	Description
Approve_Impact_Analysis	CMTeam	The CM Team member determines that the CR is valid and should be analyzed further.
Approve_CCB	CMLead, DevLead	After an impact analysis has been completed, the CR is sent to the CCB for review by either the CM Lead or the Dev Lead.
Approve_Ready_Developme nt	CMLead	The CCB approves the CR. This is a handoff to the Dev Lead.
Approve_Development	DevLead	Assigns the CR to a developer and transitions it to Development.
Approve_Ready_Sys_Testin g	DevLead, CMLead	Development has completed so the CR is passed to the test lead.
Approve_Sys_Testing	CM Lead, DevLead	System Testing begins
Approve_Ready_UAT	CM Lead	System Testing completed and the CR is being staged for UAT.
Approve_UAT	CMTeam	UAT begins.
Approve_FLB	CMTeam	First Live Batch begins.
Approve_Close	CMTeam	First live batch completes and the CR is closed.
Approve_Cancel	CMLead	The CM Lead determines that the CR is not valid.
Approve_Hold	CMLead	The CM Lead determines that the CR should be placed on hold until specified conditions are met.
Reject_to_Development	Test Team, CMTeam	The CR fails one of the states of testing and is turned over to the development team.
Reject_to_Impact_Analysis	CMTeam	During the CCB Review, it is determined that the CR needs additional analysis.
To-Admin-Central-State		ClearQuest administrative use only.

Child Change Request Action and Permission

The table below lists the actions which transition a Child CR from one state to another and the users that are permitted to perform the action.

Action	Permission	Description

Action	Permission	Description
Submit	DevLead	The DevLead will submit a new ChildCR in order to subdivide the work associated with a CR across multiple developers.
Approve_Ready_SysTesting	CMTeam	A CM Team member indicates that the ChildCR is ready for system testing
Approve_Sys_Testing	CMLead	The CM Lead approves the ChildCR to be system tested.
Approve_Ready_UAT	CMTeam	A CM Team member indicates that the ChildCR is ready for user acceptance testing.
Approve_UAT	CMTeam	The CM Lead approves the ChildCR to undergo user acceptance testing.
Close	CMTeam	The ChildCR passes user acceptance testing and is closed
Reject_to_Development	CMTeam	The ChildCR is rejected back to development because it either fails one of the test phases or it is determined that it is not ready to move into testing.
To-Admin-Central-State	RatlAdmin	***Rational Administrator use only***

Defect Action and Permissions

The table below lists the actions which transition a Defect from one state to another and the users that are permitted to perform the action.

Action	Permission	Description
Submit	TestTeam	A test team member creates a defect from Rational Quality Manger when a test case fails.
Approve-to-Review	DevLead, TestLead	The defect is determined to be valid and therefore move into a development review state. At this point, the defect will be corrected via modification to the application. NOTE: A CR or ChildCR will be linked to the defect.
Approve-to-re-Test	DevLead	The defect has been corrected and is ready to be retested.

Action	Permission	Description
Approve-to-Close	TestTeam	The test passed and the defect can be closed.
Cancel	TestLead	The defect is determined to be invalid.
Reject-to-Dev-Review	TestTeam	The defect failed the test case so it is sent back to development to be worked on.

ClearCase Security

The security protocols pertaining to the use of ClearCase that have been established are as follows:

- **VOB Access**: Read and write access to elements stored within the system's ClearCase VOBs is controlled by FSA active directory group membership. For each VOB, there will be two groups; one which allows read access and one which allows write access:
  - <system name >_<abbreviated VOB name>
  - <system name>_<abbreviated VOB name>_w

- **System name**: This is the abbreviated name for the system being developed. For example: UWF for the fictitious project Universal Widget Foundation.

- **Abbreviated VOB name**: a short name for the VOB. For example: The abbreviated VOB name "Doc" would be used for the VOB \UWF_Documentation

A complete example is as follows:

System UWF has 2 contractors, C1 and C2, supporting the development effort. Each are working on a different component of the system in separate VOBs. C1 is working on the Database for the system and C2 is working on the graphical user interface. In this example we have the following 3 VOBs:

\UWF_Documentation: stores the project documentation
\UWF_Database: The C1 developers are working on the DB related code for the system.
\UWF_GUI: The C2 developers are working on the graphical user interface for the system.

Given this scenario, we have the following groups:

uwf: top level group containing all project team members
uwf_doc: membership indicates read access to the \UWF_documentation VOB
uwf_doc_w: membership indicates write access to the \ UWF _documentation VOB
uwf _DB: membership indicates read access to the VOB \ UWF_Database
uwf _DB_w: membership indicates write access to the VOB \ UWF_Database
uwf _GUI: membership indicates read access to the VOB \ UWF_GUI
uwf _GUI_w: membership indicates write access to the VOB \ UWF_GUI

The table below lists logical groups for a system as well as read vs. write access to multiple ClearCase VOBs used in the system development effort.

Logical Groups	Read Access to System Documentation VOB	Write Access to System Documentation VOB	Read Access to Contractor 1's System Software VOB(s)	Write Access to Contractor 1's System Software VOB(s)	Read Access to Contractor n's System Software VOB(s)	Write Access to Contractor n's System Software VOB(s)
CM Team	X	X	X	X	X	X
Development Team (Contractor 1)	X		X	X	X	
Development Team (Contractor N+1)	X		X		X	X
FSA Management	X		X		X	
Operations Team	X					
Requirements Team	X		X		X	
Security Team	X		X		X	
Test Team	X		X		X	

CM Policy Enforcement

The following rules are enforced within the FSA Enterprise Solution:

- Only CM team members will be permitted to deliver changes to the project integration stream.
- Only CM team members will be permitted to deliver changes to the release-specific integration stream.
- Only CM Team members will be permitted to create baselines on the project integration stream.
- Only CM Team members will be permitted to create baselines on the release-specific integration stream.
- ClearCase UCM baselines (labels) may be created on the release-specific development stream by the development team member.
- User specific child stream off any of the prescribed streams will NOT be permitted.

RQM Security

The table below shows various RQM objects and the permissions to operate on those objects that have been granted to the RQM users. NOTE: Leads will always be included in the related team group. For example, the Test Lead for a system development effort will be included in the Test-Team group.

RQM Artifact	Action	Test Lead	Test Team	CM Lead	CM Team	Dev Lead	Dev Team	Rqmts. Lead	Rqmts. Team	FSA PM	Project PM	FSA RSG Admin
Requirements	Import into Test Plan	X										X
	View		X		X		X		X	X	X	X
Requirements Work Item (should never be used)	View											
	Create											
	Edit											
	Delete											
	Archive											
Test Plan	View		X		X		X		X	X	X	X
	Create	X										X
	Edit	X										X
	Delete			X								X
	Archive				X							X
	Create Section											X
	Remove Section											X

RQM Artifact	Action	Test Lead	Test Team	CM Lead	CM Team	Dev Lead	Dev Team	Rqmts. Lead	Rqmts. Team	FSA PM	Project PM	FSA RSG Admin
	Snapshot				X							X
Test Schedule	Create	X										X
	Edit	X										X
Test Case	View		X		X		X		X	X	X	X
	Create		X									X
	Edit		X									X
	Delete		X	X								X
	Archive				X							X
	Execute		X									X
	Create Section											X
	Remove Section											X
	Snapshot				X							
Category Types for Test Plan and Test Case **Test Plan Examples include: Release, Test Phase. ** No one other than the FSA RSG group should be able to create new categories or remove existing categories.	View		X		X		X		X	X	X	X
	Create											X
	Edit											X
	Delete											X
	Archive											X

RQM Artifact	Action	Test Lead	Test Team	CM Lead	CM Team	Dev Lead	Dev Team	Rqmts. Lead	Rqmts. Team	FSA PM	Project PM	FSA RSG Admin
Categories (Test Plan and Test Case) Represents the choice lists for the defined categories. For example: The choice list for the category "Test Phase" will initially contain "UAT" and "System" however it may be necessary to add other test phases.	View		X		X		X		X	X	X	X
	Create			X								X
	Edit			X								X
	Delete			X								X
	Archive				X							X
Test Plan/Case Templates	Create											X
	Edit											X
	Archive											X
	Set Default											X
Test Script	Create		X									X
	Edit		X									X
	Archive				X							X
Test Data	Create		X									X
	Edit		X									X
	Archive				X							X
Build Definition	View		X		X		X		X	X	X	X
	Create				X							X

RQM Artifact	Action	Test Lead	Test Team	CM Lead	CM Team	Dev Lead	Dev Team	Rqmts. Lead	Rqmts. Team	FSA PM	Project PM	FSA RSG Admin
	Edit				X							X
	Delete			X								X
	Archive				X							X
Build Record	View		X		X		X		X	X	X	X
	Create				X							X
	Edit				X							X
	Delete			X								X
	Archive				X							X
Execution Results	View		X		X		X		X	X	X	X
	Save		X									X
	Edit		X									X
	Archive	X										X
Test Suite	Create		X									X
	Edit		X									X
	Archive				X							X

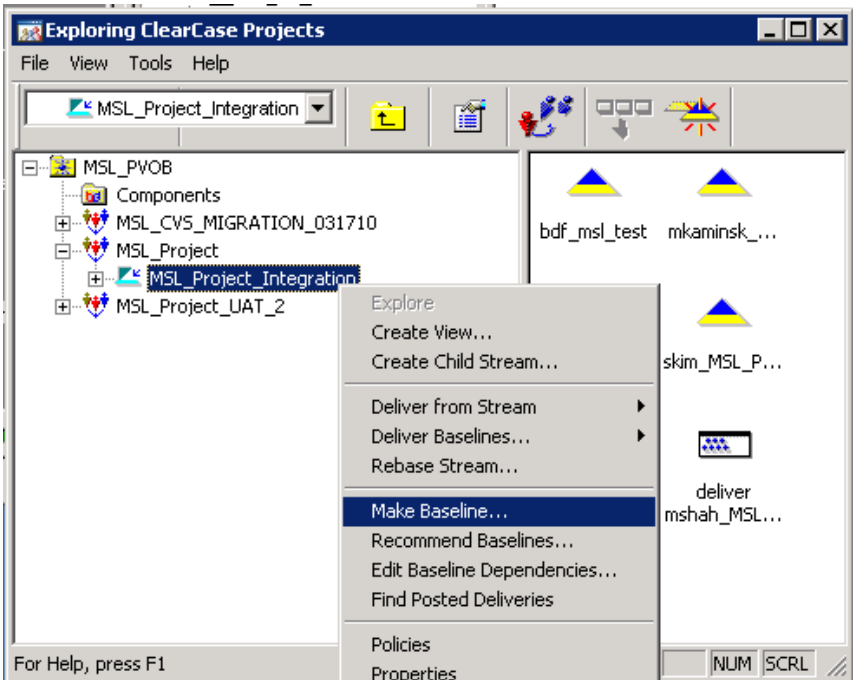
Appendix F: Job Aids

Appendix F: Job Aids

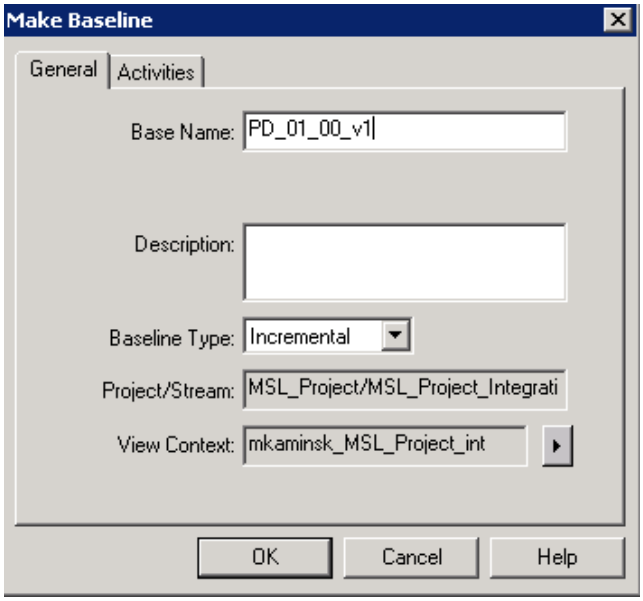
Creating a UCM Baseline using the ClearCase "Make Baseline" Function

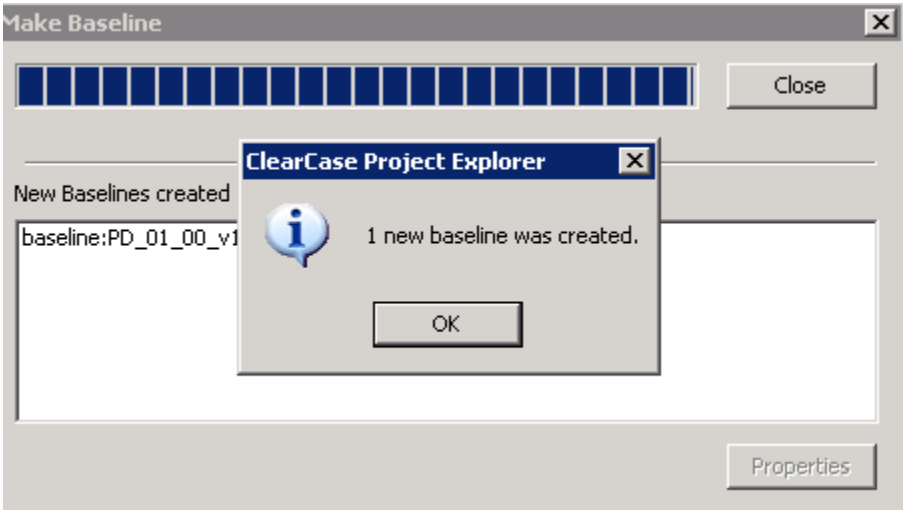
In the example below, the product baseline "PD_01.00_v1" is being created on the ClearCase UCM Project "MSL_Project"

Step 1: From the ClearCase project explorer, select the integration stream for the project being baselined. Righ-click and choose "Make Baseline"



Step 2: Enter the baseline name and an optional description and choose "OK" to create the ClearCase Baseline.

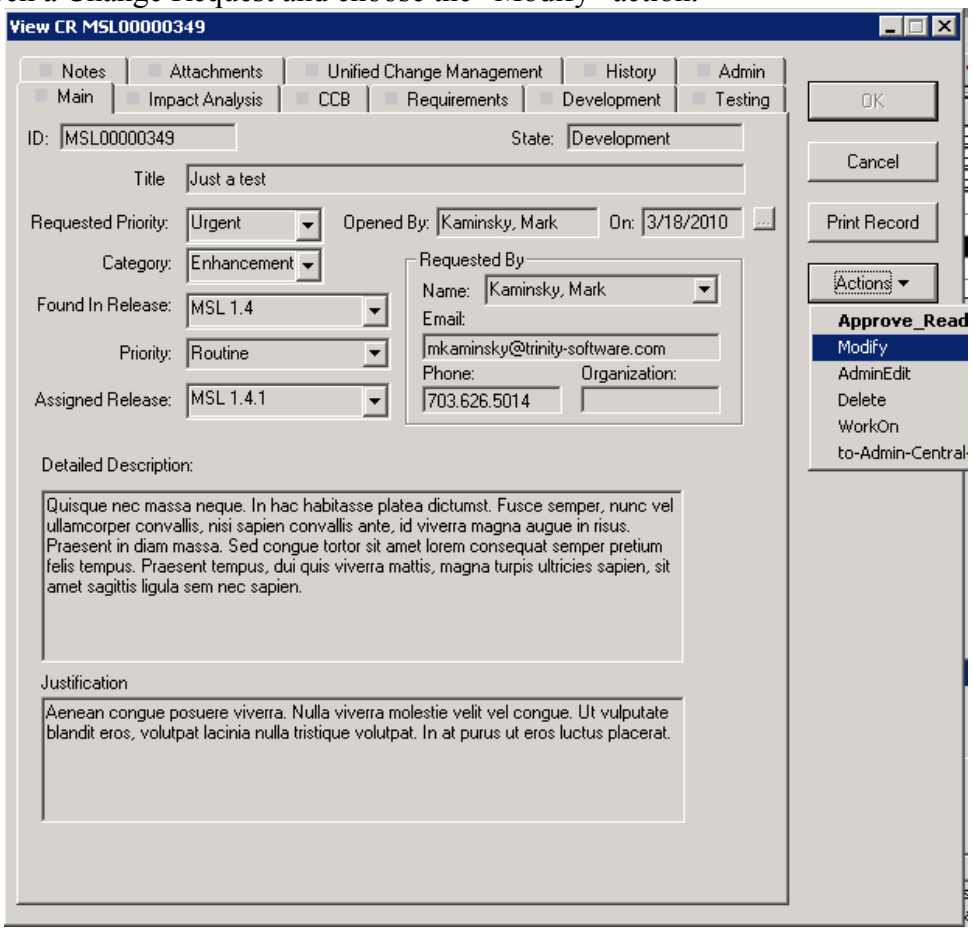

Make Baseline
General
Activities
Base Name:
PD_01_00_v1
Description:
Baseline Type:
Incremental
Project/Stream:
MSL_Project/MSL_Project_Integrati
View Context:
mkaminsk_MSL_Project_int
OK
Cancel
Help


Make Baseline
Close
ClearCase Project Explorer
New Baselines created
baseline:PD_01_00_v1
1 new baseline was created.
OK
Properties

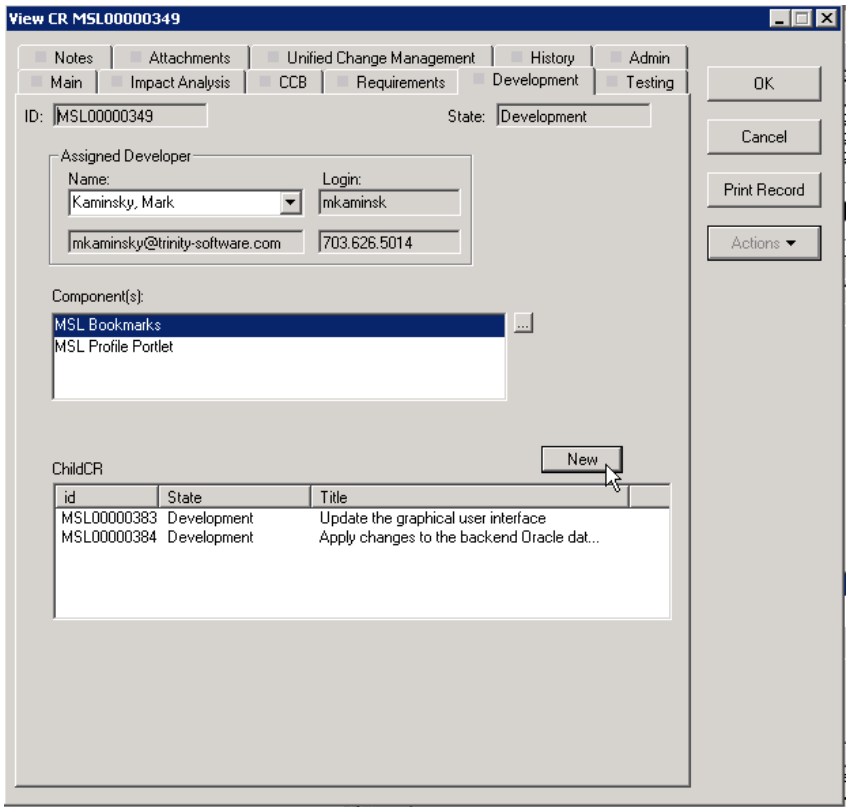
Using ChildCRs to Subdivide a CR

ChileCRs may only be created from a parent CR. It is not possible to create a ChildCR independently of a parent CR.

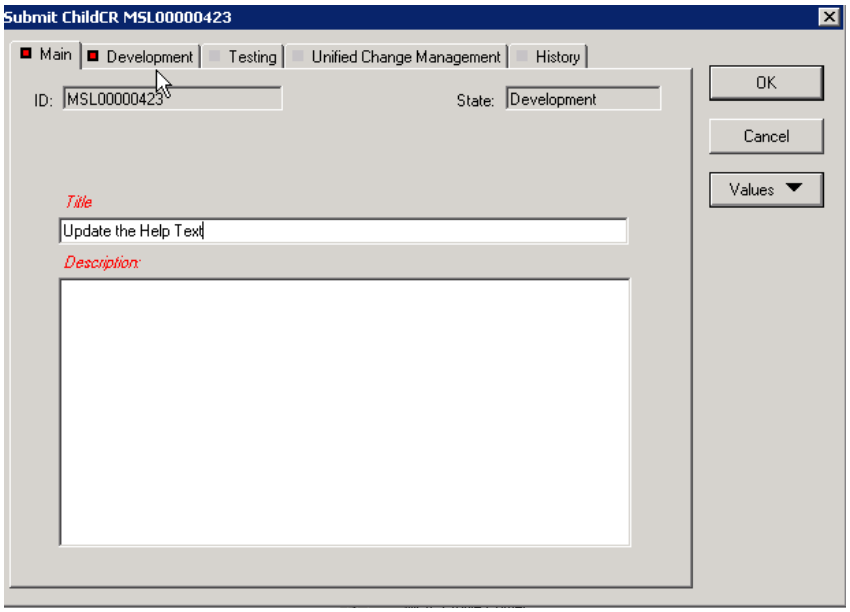
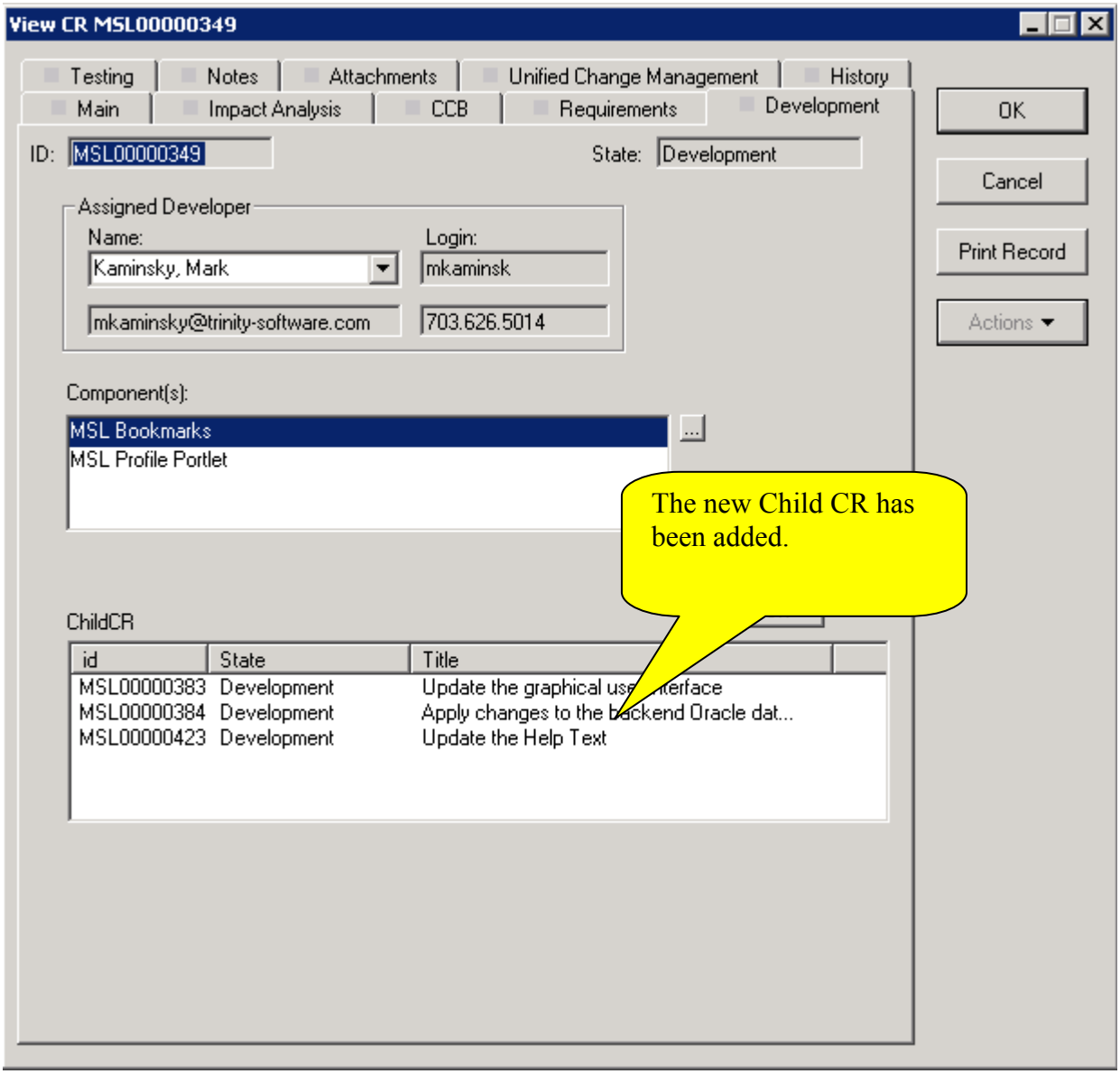
Step 1: Open a Change Request and choose the "Modify" action.



Step 2: On the "Development" tab, click the "New" button in order to spawn a new ChildCR.



Step 3: Fill in all of the required fields for the Child CR and Click "OK" to commit the changes to the database.



Step 4: Click the "OK" button on the parent CR in order to commit the changes to the database.