

# Understanding and Implementing Stakeholder Needs: the Integration of Rational ClearQuest and Rational RequisitePro

Rational Software White Paper

---



**Rational**<sup>®</sup>  
the e-development company™

# Table of Contents

<b>Introduction</b> .....	<b>1</b>
<b>Background</b> .....	<b>1</b>
<b>From Requests to Requirements</b> .....	<b>3</b>
Collecting Requests.....	4
Associating Enhancement Requests with Requirements.....	4
Viewing the Origin of Requirements.....	5
Querying on Enhancement Requests and Requirements Association.....	5
<b>About Rational tools</b> .....	<b>5</b>
About Rational Suite™.....	5
About Rational Administrator.....	6
About Rational ClearQuest.....	6
About Rational RequisitePro.....	8
<b>Rational ClearQuest and Rational RequisitePro for the Analyst</b> .....	<b>9</b>
Logging Enhancement Requests.....	9
Organizing Enhancement Requests.....	12
Enabling the Rational ClearQuest-Rational RequisitePro Integration.....	13
Creating Requirements to Associate with Enhancement Requests.....	13
Associating requirements to enhancement requests.....	16
Viewing enhancement requests properties from Rational RequisitePro.....	16
Creating enhancement requests or defects from requirements.....	18
Querying on Enhancement Requests and Requirements Association.....	19
<b>Customizing the Integration</b> .....	<b>20</b>
<b>Conclusion</b> .....	<b>20</b>

## Introduction

---

Managing change is one of the biggest challenges of effective software development. Any step in controlling change can help you deliver faster and better. Rational ClearQuest allows you to uniformly collect the change and evaluate it before it impacts your project. ClearQuest acts as a clearinghouse so that change does not prematurely affects your project.

As change is accepted and subsequently drives new system functionality, it is important to link system functionality to the original change to maintain the source of requirements put on the system.

Having a clean integration between a change request management tool (Rational ClearQuest) and a requirement management tool (Rational RequisitePro) allows you to stay focus on what you committed to deliver, but still consider your stakeholders feedback to ensure that by the time you deliver your system, it still provides value to them.

The intent of this paper is to describe how the Analyst can benefit from the integration between Rational ClearQuest and Rational RequisitePro to effectively collect stakeholder requests, then associate these requests to new or existing requirements. This integration minimizes the amount of requirement change by cleanly separating change requests from requirements.

## Background

---

The following are some common terms used in this paper.

<b>Analyst</b>	An individual who must interpret the user needs, and communicate those needs to the entire team. Job titles of those with Analyst responsibilities include, but are not limited to systems analyst, project manager, product manager, systems engineer, marketing manager, etc.
<b>Configuration (or Change) Control Board (CCB)</b>	The board that oversees the change process consisting of representatives from all interested parties, including customers, developers, and users. In a small project, a single person, such as the project manager or software architect, may play this role.
<b>Change Request (CR)</b>	General term for any request from a stakeholder to change an artifact or process. Documented in the CR is information on the origin and impact of the current problem, the proposed solution, and its cost. See also: <i>enhancement request, defect</i> .
<b>Defect</b>	A product anomaly. Examples include such things as omissions and imperfections found during early lifecycle phases and symptoms of faults contained in software sufficiently mature for test or operation. A defect can be any kind of issue you want tracked and resolved.
<b>Enhancement Request (ER)</b>	A type of stakeholder request that specifies a new feature or functionality of the system. See also: <i>Change Request</i> .
<b>Requirement</b>	A requirement describes a condition or capability to which a system must conform; it can be either derived directly from user needs, or stated in a contract, standard, specification, or other formally imposed document.  A desired feature, property, or behavior of a system.
<b>Stakeholder</b>	An individual who is materially affected by the outcome of the system. End users are often thought of as the primary stakeholders, but other stakeholders, such as shareholders and executive management, among others, also have a stake in the project.

To fully understand the value of the Rational ClearQuest-Rational RequisitePro integration, one needs to understand the difference between enhancement requests and requirements, and how they relate to one another. They differ in various ways:

- Enhancement requests are suggested changes to the system. They are not agreed-upon changes until they are evaluated by the CCB.
- Enhancement requests can be submitted by anyone (end users, customers, team members, etc.), and consequently, are formulated in the submitter’s language. Requirements, on the other hand, must comply with the team-developed common vocabulary, ensuring that all members of the team have the same understanding of the need. Requirements

must additionally conform to a set of stringent criteria: compliant with the project business case, correct, complete, verifiable, modifiable, understandable, traceable, and unambiguous. Enhancement requests do not have to conform to these criteria.

- Requirements should be unique, while there may be several enhancement requests representing the same functionality, simply because several stakeholders have the same input. Two enhancement requests, identical in nature, can be submitted in two different ways, but really represent the same request. The input from these two (or more) essentially identical enhancement requests would typically be consolidated into a single requirement.
- Requirements may represent a contractual obligation for the team to deliver functionality. Enhancement requests are not contractual in nature. Customers do not sign off on enhancement requests, they sign off on requirements documents.

The relationship between enhancement requests and requirements is such that enhancement requests drive the creation of requirements. Hence enhancement requests represent the source of requirements.

To further contrast enhancement requests and requirements, let's look at the typical enhancement request and requirement lifecycles. The following is a fictitious but applicable enhancement request lifecycle:

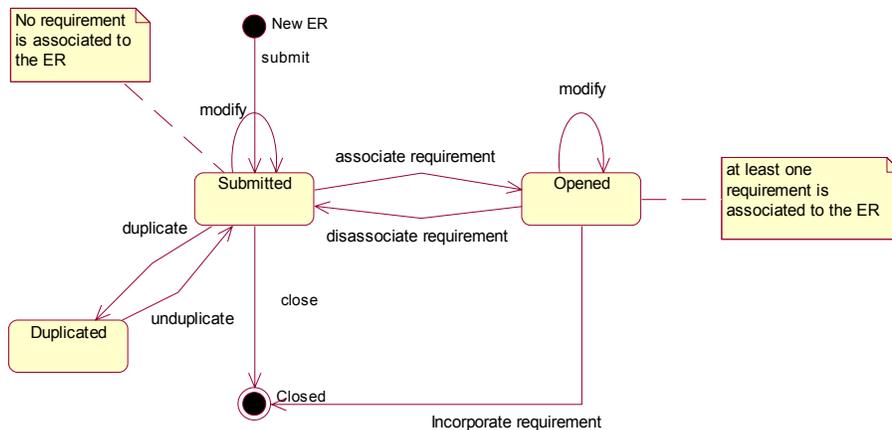


Figure 1: Sample enhancement request (ER) lifecycle

1. A stakeholder logs an enhancement request. The state of the enhancement request at that point is set to **Submitted**.
2. The CCB or other authority role performs triage on the **Submitted** enhancement requests, determines their impact to the project, and assigns one of the following states to the enhancement request:
  - **Rejected** if the enhancement request will not be considered;
  - **Duplicate** if the enhancement request is a duplicate of an existing enhancement request;
  - **Open** if the enhancement request will be considered.
3. The Analyst prioritizes the **Open** enhancement requests and filters on enhancement request fields such as Priority, Product Category, Product Area, etc.
4. Before accepting prioritized enhancement requests, the Analyst measures the impact of change by determining which existing use cases and implementation code are impacted. For example, this could be done using Rational tools by identifying in Rational Rose which packages and classes are involved in these affected use cases (*Show participants in UC* from the Rational Rose *Report* menu), and by determining in Rose the links between packages/classes and source code filenames.

5. To reflect that an enhancement request is being considered in the next release of the system, the Analyst links each **Open** enhancement request to a requirement by either:
  - associating the enhancement request with an existing requirement (the enhancement request might already have been considered, and a requirement previously created that fulfills it), or
  - creating a new requirement.
6. The Analyst transitions each enhancement request which is both **Open** and *associated with a requirement* to the **Closed** state.

Let's now look at a typical requirement lifecycle:

1. As requirements are created, their origin is tracked. When a requirement originates from one or many enhancement requests, the requirement is associated with the enhancement request(s).
2. The state of the requirement is then set to **Proposed**.
3. All **Proposed** requirements are scope managed and the ones selected for the next release (or iteration) are marked as **Approved**.
4. Requirements that are **Approved** are detailed by the Analyst for developers to start designing the capability of the system expressed in the requirement.
5. A developer is assigned to each **Approved** requirement.
6. The developer checks out the pertinent source code file(s), implements the enhancement request, and checks the source code file(s) back in.
7. The requirement status is then transitioned to **Incorporated**.

Note that this is just an example to illustrate the difference between enhancement requests and requirements. Specific states will differ between organizations.

Another important clarification is that most enhancement requests do not *become* requirements but rather help define requirements. As mentioned earlier, enhancement requests are stated in the submitter's terminology, and might be untestable, incomplete or ambiguous. The Analyst's job is to ensure that fully defined requirements are created and communicated to the team. These requirements must be understood by all members of the team; this is achieved by restating the enhancement request description into a requirement that uses the team common vocabulary, and that is complete, testable, unambiguous, etc.

## ***From Requests to Requirements***

---

The Analyst's responsibility is to ensure that the team builds a system that really solves the stakeholders' needs. The primary activities to accomplish this goal are to:

- collect stakeholder requests (to understand the stakeholder needs),
- capture and manage requirements (to define the system that will fulfill the stakeholders needs),
- and communicate these requirements to the rest of the team (to ensure that the team builds the correct system).

In this paper, we will concentrate on the first key activity (collecting requests) since the focus of the paper is how Rational ClearQuest and Rational RequisitePro work together to facilitate that part of the Analyst's job. However, since the end goal of collecting requests is to elicit requirements, we will also examine how requirements are created from the collected requests.

The activity of capturing and managing requirements may also involve the tracking of requirements source and rationale, to provide sufficient information to the CCB to make smart decisions when scope-managing the work to do in each release. When requirements are derived from enhancement requests, the enhancement request itself represents the origin of the requirement(s). Managing requirements also involves querying requirements based on their origin (for instance requirements driven by enhancement requests submitted with a high customer priority).

### ***Collecting Requests***

Gathering requirements starts with understanding stakeholder needs. This entails:

- Identifying the various requirement sources (help desk, customer reports, requirement workshops, end users, etc.). We will refer to all the sources as “stakeholders” – anyone who has a stake in the system the team is about to build.
- Eliciting needs (often expressed in the form of requests) from each of the stakeholders. These may include modification to existing functionality or new functionality.

In the life cycle of a project, the Analyst will use various mechanisms to ensure that complete input is gathered. At the start of a project, the Analyst will likely collect stakeholder needs via face-to-face elicitation techniques such as requirement workshops, brainstorming sessions, storyboarding, interviews, JAD (Joint-Application Development) sessions, to name a few techniques. These face-to-face techniques are critical early in the inception phase of the system, because they provide the team with a good starting point of what the stakeholders initially want. They also result in the original set of requirements. If stakeholder needs were not changing, the collection activity would stop there, with the results of the face-to-face interaction with stakeholders.

But software life is not that simple. Requirements will likely change as stakeholders become more educated about the system and come up with new ideas on what the system should do. Or it may be that the business surrounding change as we build the system, and the Analyst needs to take these changes into account. The Analyst needs to be in touch with these evolving stakeholder needs to make sure that the requirement set reflects that evolution, so that the system is still of value to stakeholders by the time it is delivered.

To gather on-going changes, the Analyst typically uses various methods: customer visits, e-mail forum, trade shows, etc. For each of those methods, the Analyst might receive feedback in a different format (written notes, voice mail, personal conversation, teleconference, e-mail, spreadsheet, etc.). This discrepancy, resulting from a lack of automation, leads to a lot of time spent funneling all that crucial information into a form that makes it easy for the Analyst to weigh and prioritize the requests.

Eventually, a proficient Analyst collects many requests, representing the gamut of requests for the product to build or improve. A small number of requests might be indicative of either a system not being extensively used, or of the lack of communication between the Analyst and the stakeholders. There is a real danger to accept all submitted changes without evaluating the possible impact of the change to the requirements, and therefore to the project schedule, budget, and feature set. To avoid these pitfalls, change requests should be collected in an area cleanly separated from the requirement set. There, from a potential large number of requests, the Analyst evaluates each request in order to define which of these enhancement requests will be taken into account for future releases. Some change requests are rejected at that time, without having affected the requirement set. Others might be accepted for the time being, but might not affect any requirement, because of their low priority. The value of that triage process is to protect requirements from unnecessary (or not agreed upon) changes.

### ***Associating Enhancement Requests with Requirements***

Once enhancement requests are accepted and scoped for a release (or iteration), the Analyst needs a way to incorporate the stakeholders’ input into the system definition. This is done by creating requirements from the selected enhancement requests, and by maintaining an association between these two entities to track the source of requirements. Tracking the source of all requirements is important for gathering additional details later or monitoring the system design constraints. For example, a particular enhancement request might specify a constraint on the system. If the submitter of that enhancement request ceases to be a stakeholder for the system, the constraint can be removed. Failure to track the submitter information might result in unnecessary design constraints.

Typically, during the inception phase of a release the Analyst examines the current set of enhancement requests and determines which will drive new requirements or modification of existing requirements. To be effective, the evolution of enhancement requests to requirements requires careful analysis of the requests to ensure full understanding. Automatically creating requirements from enhancement requests, can only 'pollute' the set of requirements with requirements that may not be well defined (testable, complete, unambiguous, etc.), that may not conform to the common glossary, or that are not agreed upon by the team.

When an enhancement request is identified to drive system functionality, the Analyst creates a requirement by reformulating the text of the enhancement request into a valid requirement definition, checking that the requirement text complies to standard requirement criteria (testable, unambiguous, complete, etc.). If such a requirement already exists, the Analyst simply links the enhancement request with the existing requirement. Many enhancement requests could be associated with one requirement, and one enhancement request could be associated with several requirements. In short, there may be a many-to-many relationship between enhancement requests and requirements.

Once enhancement requests are associated with requirements, the Analyst can run queries to measure:

- how many enhancement requests drove requirements for the system; this provides information on the team's responsiveness to customers;
- what types of requests (new feature, enhancement request, etc.) are driving system definition. Do we need to fix the current functionality or add new functionality?

### ***Viewing the Origin of Requirements***

As mentioned earlier, enhancement requests are a source of requirements. Consequently, when viewing requirements, visibility to the enhancement request(s) that may have led to creating this requirement provides valuable information. When scope managing the project, the Analyst may want to consult the relationships between a requirement and enhancement requests that may have led to it. The details of the associated enhancement requests may reveal important information about why that requirement was put upon the system to be built.

### ***Querying on Enhancement Requests and Requirements Association***

To analyze where requirements are coming from, the Analyst should be able to quickly query on requirements that have enhancement requests associated with them.

## ***About Rational tools***

---

### ***About Rational Suite™***

Rational Suite products were created to unify software development teams and optimize tool sets for each member on the team. All Rational Suite products contain a team-unifying platform, which includes:

- Rational ClearQuest for change request management;
- Rational RequisitePro for requirements management;
- Rational SoDA for software documentation automation;
- Rational TestManager for test planning and test assets management;
- Rational ClearCase-LT for software configuration management;
- and the best-practice process guidelines of the Rational Unified Process.

These five tools are essential to unifying the team with a common set of tools that are pertinent for all roles on the project team.

The integration between ClearQuest and RequisitePro described in this paper relies on the presence of a Rational project, created in the Rational Administrator.

### ***About Rational Administrator***

The Rational Administrator centralizes the management of the development lifecycle artifacts into one Rational project (note: “Rational repository” in older versions). A Rational project consists of the following components:

- One Rational Test datastore
- One Rational ClearQuest database
- One Rational RequisitePro project
- One or more Rational Rose models

### ***About Rational ClearQuest***

Rational ClearQuest is a scalable and configurable change request management system used for tracking and reporting on defects and other types of change requests throughout the development lifecycle. ClearQuest shortens development cycles by unifying all team members -- project managers, QA managers, testers, developers, for example. -- in managing software development change.

In addition to supporting easy out-of-the-box deployment, Rational ClearQuest is also designed to enable comprehensive and easy-to-implement customization. With ClearQuest, each type of change request (for example defect, documentation change, enhancement request, new feature.) can be associated with a unique process, or change request lifecycle. You can use ClearQuest's predefined process and associated rules right out-of-the-box or simply define your own. Powerful administrative tools allow you to tailor nearly every aspect of the system, including the request process lifecycle, database fields, user interface layout, reports, charts and queries. Most customizations can be implemented easily and without programming using the ClearQuest Designer (installed as part of a Rational Suite Administrator install), but ClearQuest also supports Visual Basic Script for more complex customizations.

By being part of the team-unifying platform in all Rational Suite products, Rational ClearQuest:

- provides every team member a consistent and intuitive way to log defects – this increases the ability to identify defects in the software prior to shipping it to end users;
- provides every team member a consistent and intuitive way to log any suggested new feature, or enhancement to existing functionality – this ensures that every team member’s feedback is reviewed and considered;
- captures all the information to help prioritize any type of change request. This may include the submitter priority, contact information, release-specific information, etc.

Change requests can be accessed from the Rational ClearQuest Windows client as well as a fully functional Web client (ClearQuest Web). ClearQuest does provide a UNIX interface as well, but the integration with Rational RequisitePro is currently provided only with the Windows client.

Rational ClearQuest can manage many types of requests: requests to fix defects, to improve an existing feature, to add a new feature, and others. For the purpose of this paper, we will focus on enhancement requests, and requests for new features, since the Analysts are mostly concerned with those types of requests as sources of requirement information.

### ***Rational ClearQuest Schemas***

Before using a Rational ClearQuest database to store change requests, the project administrator defines what ClearQuest schema to use (figure 2). A schema specifies the Change Request Management record types, fields, GUI forms, actions, behaviors, states and hooks. “Out-of-the-box”, ClearQuest provides several sample database schemas from which customers can choose to build their change request user database. ClearQuest also provides schema packages that can be applied to a base schema to supplement it. Each package may provide some specific record UI elements (like a tab on a GUI form), hooks, and other information. Discussing ClearQuest packages is beyond the scope of this paper. For more information, please refer to the ClearQuest Designer Understanding Packages help topic. The table below lists the schemas provided in ClearQuest.

Schema Name	Schema Description
Blank	Contains system fields only. Use Blank to create a schema from scratch.
Common	Contains metadata that is common to the remaining schemas.
Defect Tracking	Contains the fields necessary to start using Rational ClearQuest to track defects in a software development environment.
Rational Suite AnalystStudio	For use with Rational Suite AnalystStudio. Provides code for the ClearQuest and Rational RequisitePro integration.
Rational Suite DevelopmentStudio	For use with Rational Suite DevelopmentStudio. Contains fields and rules that work with Rational's Purify, Visual Quantify, and Pure Coverage. Provides code for the ClearQuest and RequisitePro integration
Rational Suite TestStudio	For use with Rational Suite TestStudio. Contains fields and rules that work with Rational's TeamTest, RequisitePro, Purify, Visual Quantify, and Pure Coverage. Provides code for the ClearQuest and RequisitePro integration
Rational Suite Enterprise	For use with Rational Suite Enterprise. Contains fields and hooks that work with all Rational products. Provides code for the ClearQuest and RequisitePro integration
UnifiedChangeManagement	For use with Rational ClearCase. Supports the UCM process by integrating with ClearCase.

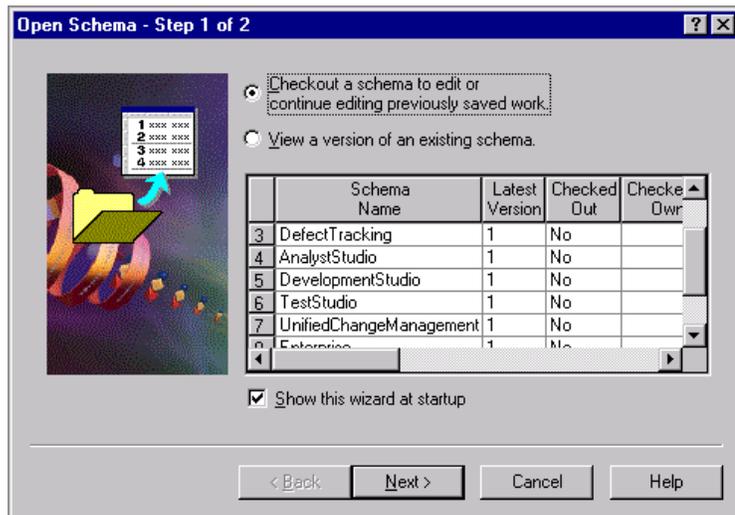


Figure 2: Schema selection screen in the ClearQuest Designer

It is important to realize that a Rational ClearQuest schema is chosen for the *entire* change request database (not for each role on that project), as the team typically uses one change request database. This database is used by all team members regardless of their role on the team. When a team uses a mix of Rational Suite products, such as testers using Rational Suite TestStudio, developers using Rational Suite DevelopmentStudio, analysts using Rational Suite AnalystStudio, we recommend using the Rational Suite Enterprise schema.

Using the Rational ClearQuest security scheme in the ClearQuest Designer, the project administrator can additionally optimize the ClearQuest interface for each user group. Schema visibility can be modified for each user group (analysts, testers, developers, etc). For instance, the administrator could hide the Test Data tab (available in the Rational Suite

Enterprise schema) for the analysts user groups, since the information on this tab is not pertinent to analysts. Note that the visibility is set at the tab level, not the individual field level.

### ***Rational ClearQuest Out-Of-The-Box Record Types***

All “out-of-the-box” ClearQuest schemas described above include the following record types:

- Defect record type, and
- EnhancementRequest record type

The Enhancement Request record type provides a simpler interface (than the Defect record type) for submission of enhancement requests. Typically submitting enhancement requests requires less information than submitting defects. Defects and enhancement requests are recorded using specific Rational ClearQuest fields, and can be assigned different states. Only enhancement request-specific information is requested from the submitter, while typically more extensive, defect-specific information (such as reproducibility, build, environment, etc.) is requested when logging a defect.

Note that for customers who are familiar with previous versions of Rational ClearQuest, the Defect record type still allows users to log enhancement requests by setting the Severity field to the “5-Enhancement” value. Available options for the Defect Severity field are: 1-Critical, 2-Major, 3-Average, 4-Minor, 5-Enhancement. When starting with a new project, we recommend using the Defect record type to log defects and the Enhancement Request record type to log enhancements. As mentioned earlier, simplifying the enhancement request submission form results in less effort to log enhancement requests. To minimize end-user confusion, we recommend removing the “5-Enhancement” severity option on the Defect form.

### ***Rational ClearQuest Forms***

Each Rational ClearQuest record (Defect, EnhancementRequest, for instance) has two GUI forms: a Submission form and a Record form. The Submission Form is the GUI form that is provided to create new records. The Record Form is the GUI form that is available once the record has been created. The Record Form typically displays more fields than the Submission Form. These additional fields are used to internally organize submitted requests, once they have been submitted. Typically information added on the Record Form represents information that the team does not want to expose to the submitters (like internal priority), but adds tremendous organizational help to analysts.

### ***About Rational RequisitePro***

Rational RequisitePro is a flexible and easy-to-adopt requirements management tool, used for documenting and managing requirements throughout the development lifecycle. RequisitePro increases the likelihood to deliver quality systems on time and on budget by unifying all team members -- project managers, QA managers, testers, developers, etc. -- in communicating and managing systems requirements.

With the tightest Microsoft Word integration in the requirements management tool market, Rational RequisitePro fits in customers’ environment better than any requirements management tool, making it easy to use and easy to adopt. Requirements documents, under RequisitePro control, can be created, modified and managed, and are complemented with database information, such as requirement attributes, traceability relationships, and revision history.

Additionally, e-mail-enabled discussion groups capture the team feedback on project-wide or requirement-specific issues. Customers can use Rational RequisitePro’s predefined project structures out-of-the-box or simply define their own. RequisitePro can also be extended using the RequisitePro Extensibility Interface, a COM-based API, which allows programmatic access to requirements.

By being part of the team-unifying platform in all Rational Suite products, Rational RequisitePro provides:

- access to all requirements for every team member, by using a central database;
- an easy way to query on requirements information for all team members;
- an easy way to check for requirement coverage to all team members; developers can quickly assess that they have documented in detail all features defined by marketing;

- quick and easy impact analysis tailored to each team member; developers can quickly review the impact of marketing requirements change on their specifications; documentation writers can quickly review the impact of any requirement change on the user manual.

Either the Windows client (Rational RequisitePro) or the Web client (Rational RequisiteWeb) allows users to create, view and modify requirements stored in a commercially available database (Microsoft Access, Microsoft SQL Server or Oracle). If you purchased a floating license of RequisitePro or any Rational Suite product, RequisiteWeb is installed using the Custom installation.

In Rational RequisitePro, requirements are organized by type. Each requirement type provides a set of requirement attributes, which can easily be modified.

## ***Rational ClearQuest and Rational RequisitePro for the Analyst***

---

In this section, we'll take a look at the functionality available to the Analyst in Rational ClearQuest and Rational RequisitePro.

### ***Logging Enhancement Requests***

Rational ClearQuest helps the Analyst gather enhancement requests by providing a complete and consistent enhancement request form, whose entries are logged in the same central change request database, regardless of who logs the request. Internal stakeholders (team members) can use the ClearQuest Windows client or the ClearQuest Web client, while external stakeholders use the ClearQuest Web client with limited access.

Submitted enhancement requests are *complete* because Rational ClearQuest's advanced customization capabilities enable the Analysts to define their own fields and values to gather the information they need to weigh the enhancement requests. Fields can be marked required, ensuring that critical information is gathered upon submission and maintained in the database. For example, fields such as customer priority, description of the request, and origin, helps the Analyst document the source of requirements that will later be created from the enhancement requests. Most customers will be satisfied with the "out-of-the-box" support for enhancement requests fields, but the potential to modify or extend these fields is provided in the customization features of ClearQuest.

Submitted enhancement requests are *consistent* between submitters: submitters can use the Rational ClearQuest Windows client or the ClearQuest Web client. Additionally, customers and end users can be provided the same enhancement request submission form without consuming a ClearQuest license. (Security features allow the project administrator to control external stakeholders access to the system.) This unique feature is key to easily engaging customers and end users in providing ongoing feedback.

Submitted enhancement requests are all gathered in a *central* database. Regardless of which Rational ClearQuest client the submitter uses, all enhancement requests are stored in the same database. This makes it very convenient for the Analyst to view all requests at once, and query on them for prioritization purposes.

To submit an enhancement request, a stakeholder selects *Submit a Record* from either the Rational ClearQuest Web client or the ClearQuest Windows client. The Enhancement Request submission form is displayed, with fields organized in three tabs (see Figure 3). The stakeholder fills out the mandatory fields (Headline, Customer Priority, and Customer Name). Additionally, the stakeholder can provide contact information such as a phone number or e-mail address. This is valuable in case the Analyst needs to later get in touch with that stakeholder for clarification. Once the information is entered, the submitter clicks the OK button to end the submission. Upon submission, the information is automatically entered in the ClearQuest database, readily available to the Analyst.

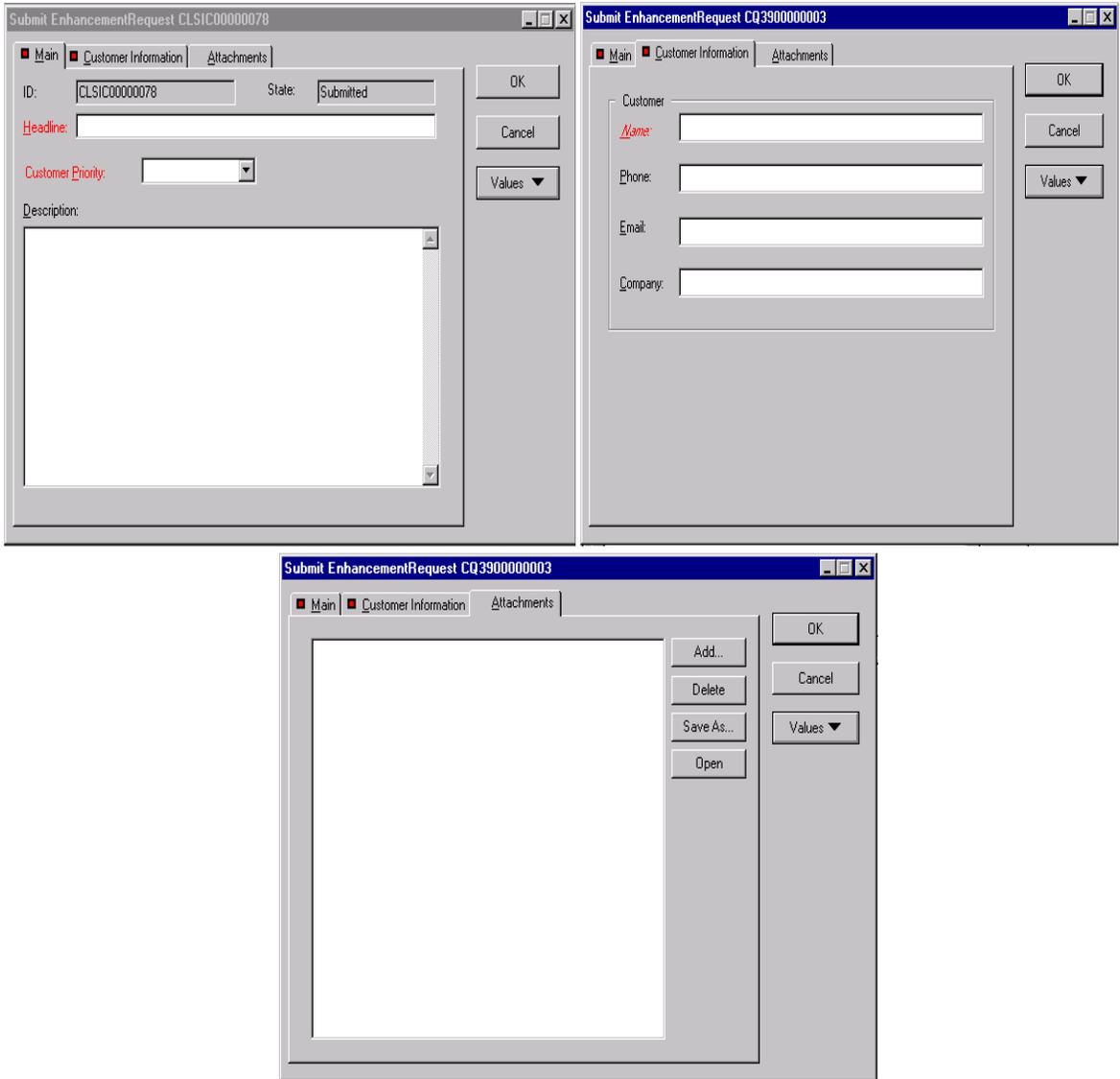


Figure 3: “out-of-the-box” Rational ClearQuest *Enhancement Request* submission form

As mentioned earlier, a unique feature of the Rational ClearQuest Web client is that it can be configured to provide restricted functionality to non-ClearQuest licensees. With restricted access, a ClearQuest Web user can submit change requests and access a single administrator-defined ClearQuest query. This unique feature allows key stakeholders, who do not belong to the software team building the system, to enter their feedback via the Web.

The Rational ClearQuest administrator defines the user group with limited access, allowing an unlimited number of external stakeholders to log requests at no additional expenses to the project (without consuming a ClearQuest license). See Figure 4.

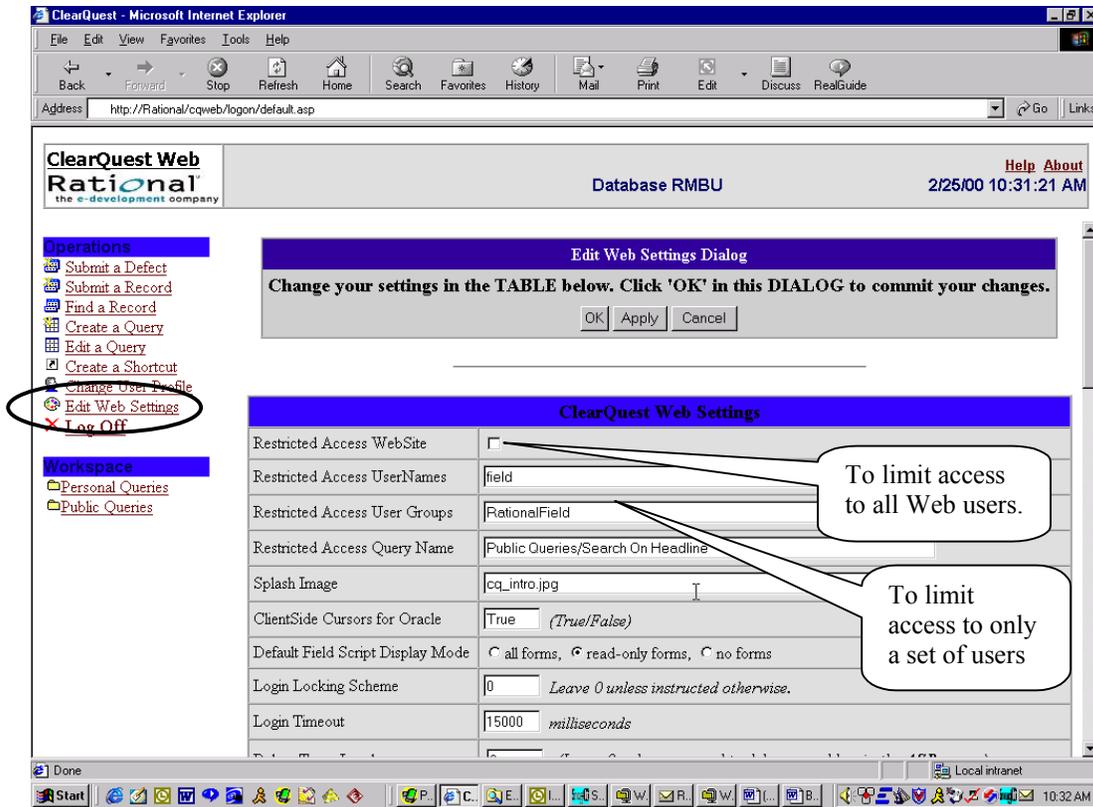


Figure 4: Restricting access to Rational ClearQuest Web users (login as the Administrator and click the Edit Web Settings link)

Users with restricted-access can log any records (enhancement requests or defects for instance) the same way internal users would log these requests (see Figure 5 and 6).

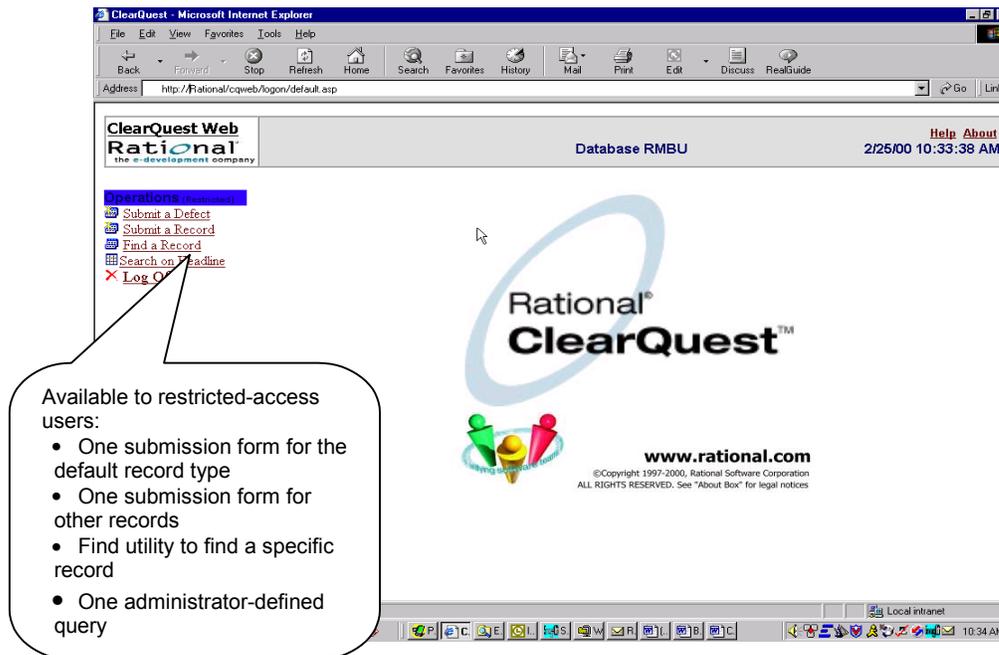


Figure 5: Web interface for users with limited access to the Rational ClearQuest database

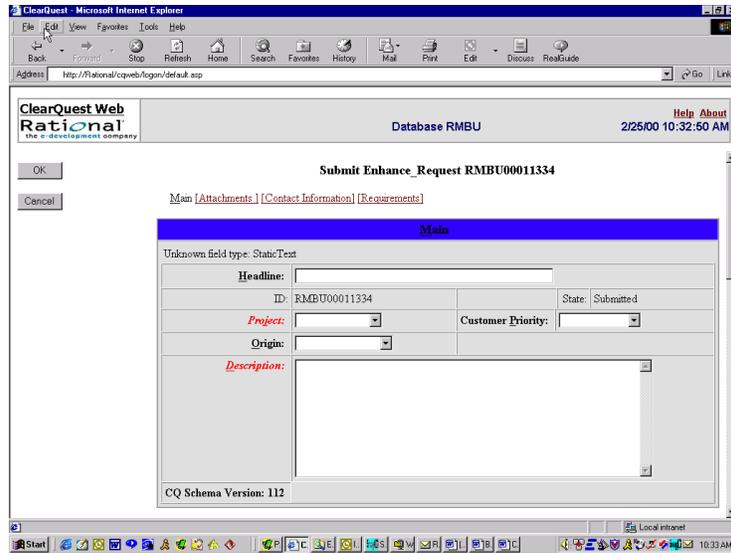


Figure 6: ClearQuest restricted access Submission Form for enhancement requests

Note that users provided with restricted access still have the choice to submit any record type defined in the Rational ClearQuest database (Defect, Enhancement Request or any other).

### Organizing Enhancement Requests

Once enhancement requests have been submitted in Rational ClearQuest, the Analyst further qualifies these requests by adding additional information to each request. This again can be done using either the ClearQuest Windows client or the ClearQuest Web client, although the Analyst will likely prefer the more convenient Windows client.

To complement the information submitted, Rational ClearQuest provides several tabs on the record form: the Analysis tab, the Resolution tab, and the Notes tab.

The intent of the Analysis tab is to allow the Analyst to complement the submitter information with information to organize this enhancement request with all the others for the purpose of querying enhancement requests later (see Figure 7). For instance, the Analyst might set the Marketing Priority of an enhancement request to Low even though the submitter Customer Priority was set to High, because, that particular request might not comply to the project business case as well as others. The customer might not know the business case, but it is the Analyst’s responsibility to ensure that requirements do not stray from it.

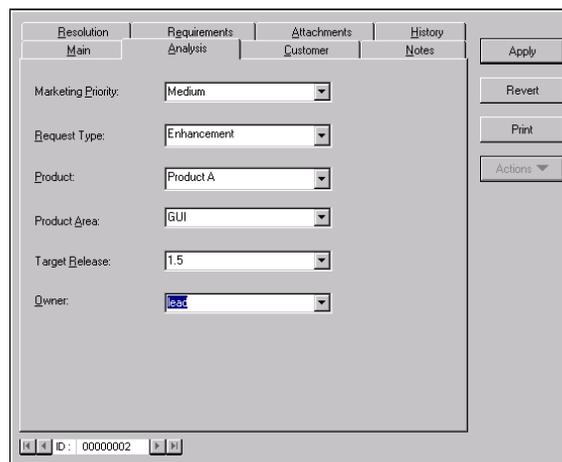


Figure 7: ClearQuest Enhancement Request Analysis tab

On the Resolution tab, the Analyst can record the fact that an enhancement request is a duplicate of an existing enhancement request. The Notes tab can be used to add additional information.

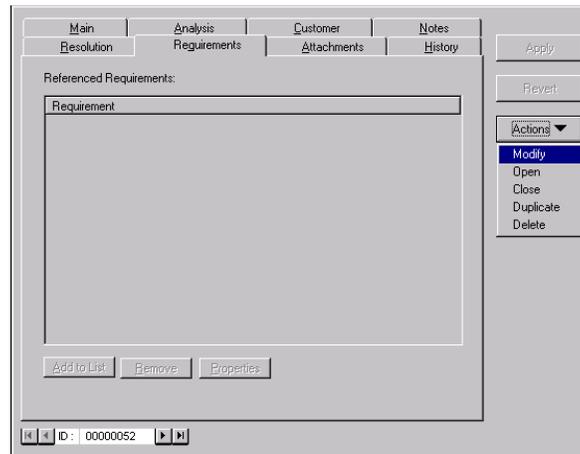
***Enabling the Rational ClearQuest-Rational RequisitePro Integration***

To enable the Analyst to associate enhancement requests with requirements, the Rational ClearQuest-Rational RequisitePro integration must be first enabled by the project administrator. This setup is performed once and involves editing then running a setup file (ASCQISetup.bat) described in the *Configuring Rational Suite* guide (available on the Rational Suite Documentation CD).

***Creating Requirements to Associate with Enhancement Requests***

The Analyst is then ready to associate the requirements with enhancement requests. Linking requests to requirements is done using the Rational ClearQuest and Rational RequisitePro Windows clients. The association cannot currently be set via the ClearQuest or RequisitePro Web clients. This association is also done once the enhancement request record has been created (not upon submission).

The EnhancementRequest record form contains a Requirements tab. Note that this tab is also available on the Defect record form, but since we are concentrating on the Analyst’s job in this paper, we will not be discussing in details relationships between defects and requirements. In brief, the requirements tab for defects can be used to indicate that a defect either affected a requirement (this defect may prevent the requirement from being completely fulfilled), or that a defect was the result of a badly specified requirement (for instance a defect analyzed as “As Designed” may come from the fact that the original requirement was not clearly stated, leading to a design “defect”).



**Figure 8: Rational ClearQuest Enhancement Request Requirements tab**

To associate one or many requirements with an enhancement request, the Analyst first invokes the Modify action on the record form (figure 8). This provides write-access to the record form and enables the action buttons (Add to List, Remove and Properties).

To be associated with one or many requirements, the enhancement request record form must point to a valid Rational project. This is done by setting the Suite Project field on the Main tab of the enhancement request record form to a valid Rational project (see Figure 9). Such a Rational project would have been created using the Rational Administrator. When creating a Rational project, the Rational administrator specifies, among other things, a Rational ClearQuest database and a Rational RequisitePro project.

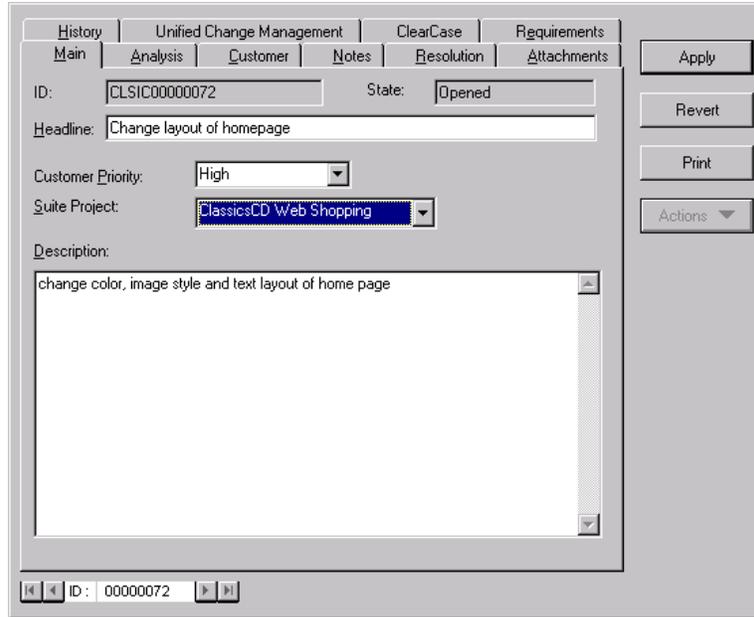


Figure 9: Suite Project field set to a valid Rational project

To associate requirement(s) with the enhancement request, the Analyst clicks the Add to List button on the Requirements tab. This brings up a Select Requirement dialog box to either select an existing requirement or create a new requirement (figure 10). Requirements displayed on this dialog box are dynamically queried from the Rational RequisitePro project. Requirement types are available for selection, to allow the Analyst to select the type of requirement to create. Most enhancement requests would drive high-level requirements, but some may drive detailed requirements, if the enhancement request is very specific and very detailed. Out-of-the-box, the integration is setup to associate Rational ClearQuest defect and enhancement request records to Feature type of requirements. This association can be customized by the administrator (see Customizing the Integration section of this paper).

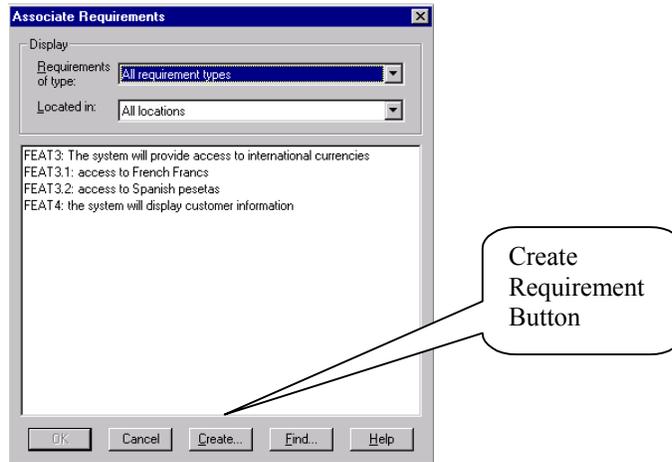


Figure 10: Select (and Create) Requirement dialog box

To create a requirement, the Analyst simply clicks the Create... button. The Rational RequisitePro Requirement Properties dialog box is displayed (see Figure 11). On the General tab, the requirement text is entered while requirement attributes are set on the Attributes tab. Traceability relationships are set on the Traceability tab, and parent-child relationships can be set on the Hierarchy tab.

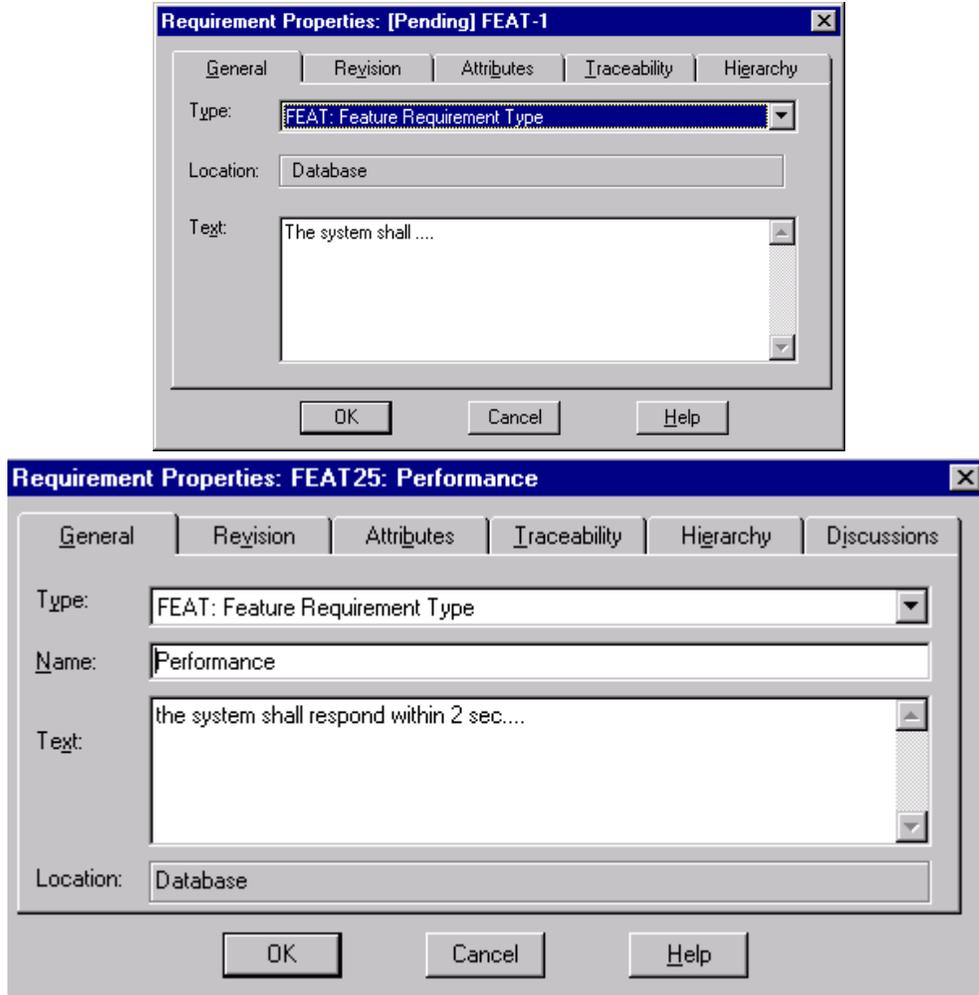


Figure 11: Requirement Text, Attributes, Traceability and Parent-Child settings dialog box

Requirements created from Rational ClearQuest are located in the Rational RequisitePro database (no requirements documents are created from ClearQuest). From RequisitePro, you can later relocate the requirement to a requirements document by cutting the requirement (using RequisitePro Requirement → Cut) out of the RequisitePro Views and pasting it (using RequisitePro Requirement → Paste) into a RequisitePro document.

To associate multiple enhancement requests with the same requirement, the Analyst would select the enhancement requests in the ClearQuest results set window, using the CTRL or SHIFT key, then click the Modify action. The Multiple Record Update dialog box provides options to the Analyst on how to update the set of enhancement requests. See figure 12.

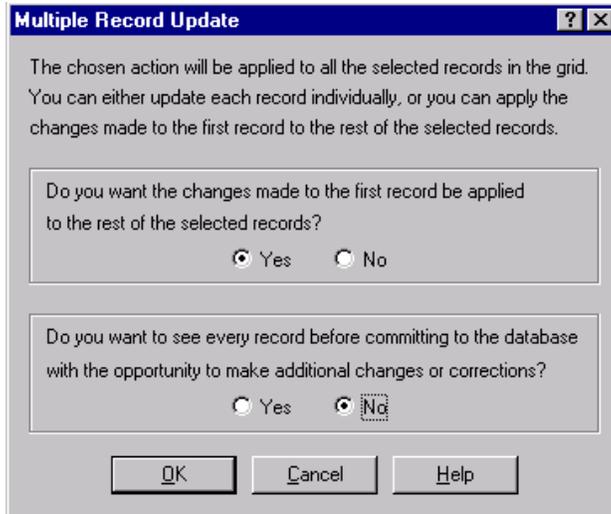


Figure 12: Rational ClearQuest dialog box when initiating association of multiple enhancement requests with requirements(s)

### Associating requirements to enhancement requests

From Rational RequisitePro, the Analyst can associate requirements to enhancement requests stored in Rational ClearQuest. This is achieved by setting the attribute value of the EnhancementRequest requirement attribute. This attribute can contain an unlimited number of enhancement requests.

### Viewing enhancement requests properties from Rational RequisitePro

While associating enhancement requests to requirements in Rational RequisitePro, the Analyst might need to review, or change the state of the enhancement request – for instance from Submitted to Open, or from Open to Assigned depending on the change request process implemented in Rational ClearQuest. The Analyst can make that change of state while in RequisitePro by accessing the EnhancementRequest attribute. When viewing requirements in RequisitePro’s attribute matrices or Requirement Properties dialog box (Attribute tab), the EnhancementRequest (and Defect) attribute is displayed with a ‘...’ button by its side. See figure 13.

Requirements:	EnhancementRequest	Defect
[-] FEAT3: The system will provide access to...	CQ3900000004	
▶ FEAT3.1: access to French Francs	CQ3900000002,CQ3900000004	...
FEAT3.2: access to Spanish pesetas	CQ3900000002,CQ3900000004	
FEAT4: the system will display customer information	CQ3900000002,CQ3900000004	
* <Click here to create a requirement>		

Attribute to select or view enhancement requests from RequisitePro

Figure 13: EnhancementRequest requirement attribute to list Rational ClearQuest enhancement requests

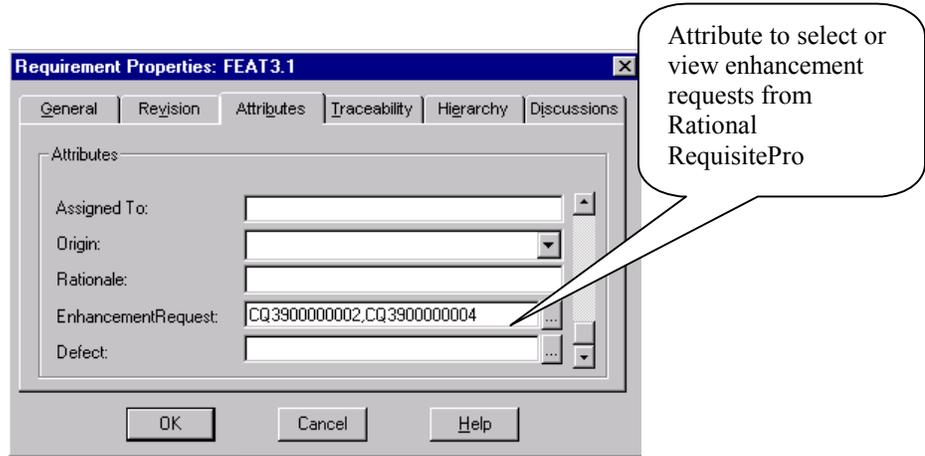


Figure 14: EnhancementRequest attribute in Requirement Properties dialog box

Clicking the ‘...’ button brings up a dialog box that displays an association dialog. Already associated enhancement requests are listed at the top.

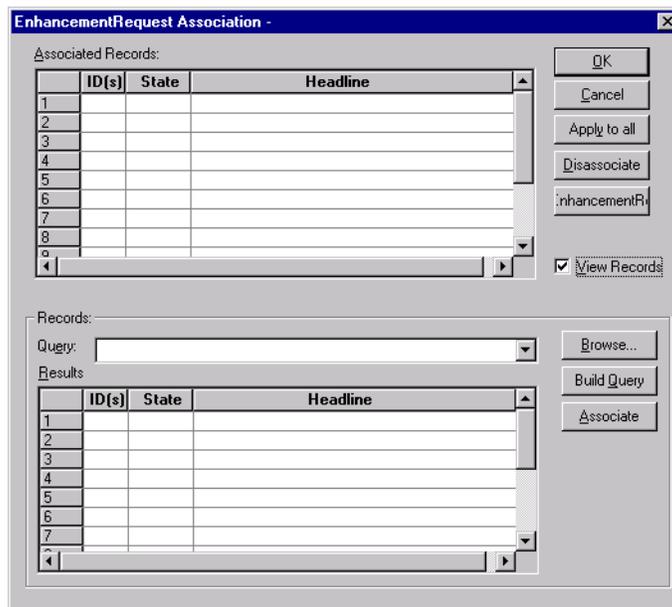


Figure 15: Associate Enhancement Request dialog box invoked from Rational RequisitePro

To select enhancement requests stored in Rational ClearQuest, the Analyst browses (using the Browse button) to select an existing ClearQuest query. The query is run and the results are displayed at the bottom of the screen. From that list, the Analyst can select enhancement requests and associate them with the requirement by simply clicking the Associate button.

Note that the Analyst can build a Rational ClearQuest query from that dialog box and save it in any existing ClearQuest query folders.

Important Note: Queries must display the ClearQuest ID field to work properly in the integration.

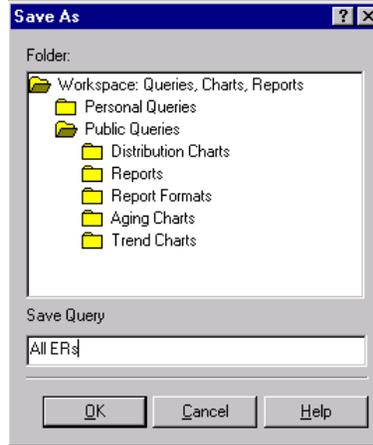


Figure 16: Saving a Rational ClearQuest query from Rational RequisitePro

Once the query results are displayed, the Analyst can select one or many of the resulting enhancement requests and click the Associate button to associate them with the requirement. Clicking the Associate button will add the selected enhancement requests to the list at the top of the dialog.

From either list the Analyst can select an enhancement request and get to its details.

Results			
	Headline	id	State
1	Ability to compare current proje	CQ3900000002	Submitted
2	improve performance	CQ3900000004	Submitted
3			
4			
5			

Figure 17: Accessing enhancement request details from Rational RequisitePro

From this dialog box in Rational RequisitePro, the Analyst can view the enhancement request properties.

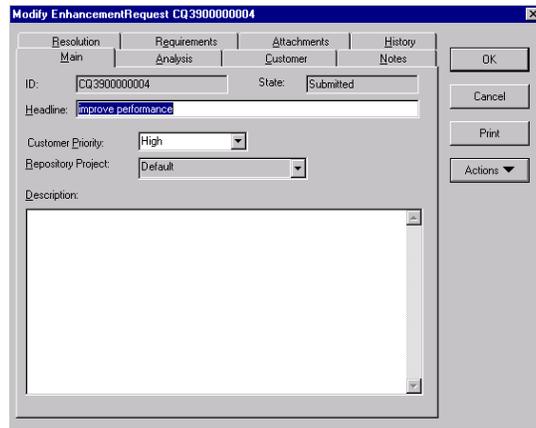


Figure 18: Enhancement request Properties dialog box from Rational RequisitePro

### ***Creating enhancement requests or defects from requirements***

From Rational RequisitePro, any team member can also submit either an enhancement request or a defect against the selected requirement. Creating a Rational ClearQuest enhancement request or defect from RequisitePro will automatically associate the enhancement request or the defect to the RequisitePro requirement from which the creation was initiated. The creation of ClearQuest records is performed from the dialog on Figure 15, clicking on the New Record button.

### Querying on Enhancement Requests and Requirements Association

From Rational RequisitePro, the Analyst can query requirements using enhancement request criteria.

In RequisitePro, create an attribute matrix for the requirement type to query and bring up the query interface. Select the Enhancement Request attribute to query on, and enter the Rational ClearQuest ID prefix as Value to include for querying. That query will return all requirements that have an association with one or many enhancement requests. If you only want to see requirements that have more than one enhancement requests associated with them, enter ‘,’ as the keyword, because enhancement requests are separated by commas.

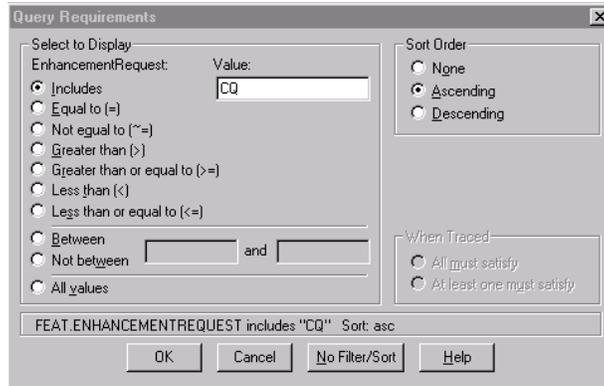


Figure 19: Querying the associations from the Rational RequisitePro query dialog box

From Rational ClearQuest, you can query enhancement requests that have (or don't have) requirements associated with them. Since the ClearQuest Requirements field of an enhancement request can contain multiple requirements, note that requirements cannot be displayed on the query Result Set. To display all enhancement requests that are associated with one or many requirements, create a ClearQuest query and filter on the *Requirements\_List.Requirements\_List.Reg\_GUID* field with the "Is Null" operator (with the "Not" check box marked or cleared).

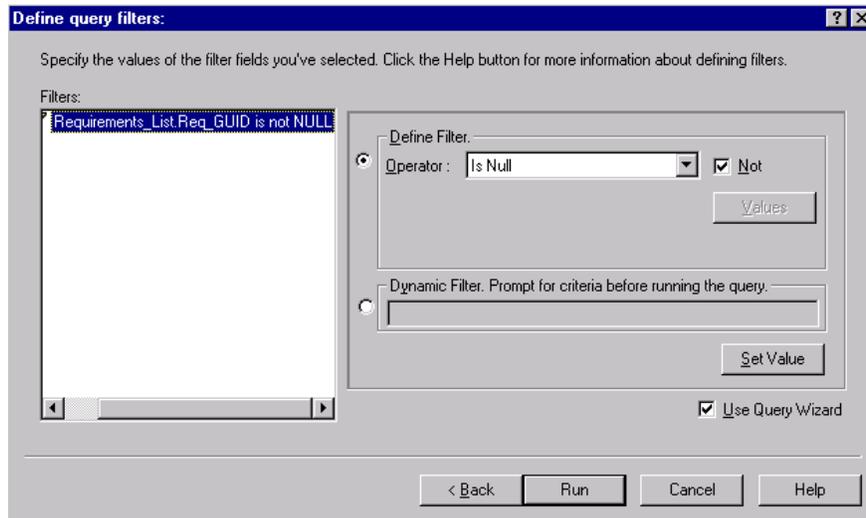


Figure 20: Querying the associations from the Rational ClearQuest query wizard pane

At this time the Rational RequisitePro metrics add-in does not provide queries on the association between enhancement requests and requirements.

## ***Customizing the Integration***

---

Out of the box, the integration provides users with the ability to associate either Defect or Enhancement Request to feature requirements. It specifically links Rational ClearQuest Enhancement Request and Defect records to Rational RequisitePro requirements of type Feature. Some customers prefer to use other record types than the provided Defect and Enhancement Request ClearQuest record types, or other requirement types than the provided Feature requirement type.

The integration has been built to allow such customization. If you decide to customize the integration, then before being deployed to the software team, the project administrator can customize which ClearQuest record and which RequisitePro requirement types can participate in the integration. Note that we strongly recommend ClearQuest administrative skills to perform the customization step. Customization steps are described in the *Configuring Rational ClearQuest and RequisitePro integration* chapter of the *Configuring Rational Suite* guide, available on your Rational Documentation CD.

## ***Conclusion***

---

In this paper, we discussed the value of the Rational ClearQuest and Rational RequisitePro integration from an Analyst usage standpoint.

This integration provides the Analyst an unbeatable value to effectively collect stakeholder feedback with the goal of creating requirements that truly represent user needs, and to track requirements' origin to better scope requirements.

By cleanly separating change requests from requirements, the amount of requirement change can be minimized to only accepted changes.

We value your feedback. Please subscribe to the Rational Requirements Management user group at <http://www.rational.com/products/reqpro/usergroups/index.jhtml>.



Corporate Headquarters  
18880 Homestead Road  
Cupertino, CA 95014  
Toll-free: 800-728-1212  
Tel: 408-863-9900  
Fax: 408-863-4120  
E-mail: [info@rational.com](mailto:info@rational.com)  
Web: [www.rational.com](http://www.rational.com)

For International Offices: [www.rational.com/worldwide](http://www.rational.com/worldwide)

Rational, the Rational logo, Rational the e-development company and Rational Rose are registered trademarks of Rational Software Corporation in the United States and in other countries. Microsoft, Microsoft Windows, Microsoft Visual Studio, Microsoft Word, Microsoft Project, Visual C++ and Visual Basic are trademarks or registered trademarks of Microsoft Corporation. All other names used for identification purposes only and are trademarks or registered trademarks of their respective companies. ALL RIGHTS RESERVED. Made in the U.S.A.

© Copyright 2000 Rational Software Corporation.

TP-##### /00. Subject to change without notice.