Test Management and Tracking with the IBM Rational ClearQuest Test Management

A case study

Kevin Yeung-Kuen See
Pierre Regazzoni

Learn about the experience and findings of a study conducted by the IBM® DB2® Function Verification Test team on the feasibility of using IBM Rational® Enterprise ClearQuest® test management for functional verification testing, tracking, and monitoring. This article also describes ClearQuest test management component and explains setup steps.

Context of this case study

Development of IBM DB2 database software for Linux®, UNIX®, and Microsoft® Windows® involves multiple teams across different locations. Each release is complex in nature and requires constant monitoring to verify its quality. In such large projects, each team might use different ways to manage and track testing. The IBM DB2 Function Verification Test team is no exception. Therefore, we conducted a study using the new test management functionality in IBM Rational ClearQuest in the hope of finding a more effective way to track, report, and monitor the progress of functional tests. The Function Verification Test team evaluated the Beta 2 version of ClearQuest test management from January 2006 through May 2006.

Overview of ClearQuest test management

Rational ClearQuest software is a well-known defect- and change-tracking system. ClearQuest test management is the latest Rational test management solution, built as an alternative to IBM Rational Test Manager™. ClearQuest test management runs as a schema on top of ClearQuest Version 7. It integrates with other IBM products, such as Rational ClearCase®, Rational RequisitePro®, Rational Manual Tester, Rational® Functional Tester, and Rational® Performance Tester. ClearQuest test management also offers a plug-in that can be installed on top of ClearQuest V7 clients.

When and how ClearQuest test management is useful

ClearQuest test management can be useful in four areas of testing:
• **Test planning:** To define and group test cases

• **Test authoring:** To create the documents or automated scripts for executing the test cases

• **Test execution:** To run test cases written with Rational Functional Tester or Rational Performance Tester and to record the test results and store them in the backend database

• **Test reporting:** To analyze testing during planning through execution with built-in or custom-created queries or graphs

**ClearQuest test management clients**

ClearQuest test management provides three graphical user interfaces (GUIs):

• The full Eclipse plug-in ClearQuest client
• A native ClearQuest client
• A Web client

Each interface provides a different level of functionality. The Eclipse client has the most capabilities. The chart in Figure 1 compares the capability of each of the clients.

**Figure 1. Comparison of the 3 ClearQuest test management clients**

<table>
<thead>
<tr>
<th></th>
<th>ClearQuest Client (Eclipse RCP client)</th>
<th>ClearQuest Client for Eclipse (plug-in)</th>
<th>ClearQuest Web</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test planning</td>
<td>⚡</td>
<td>⚡</td>
<td>⚡</td>
</tr>
<tr>
<td>Test authoring</td>
<td>⚡</td>
<td>⚡</td>
<td>⚡</td>
</tr>
<tr>
<td>(associate script)</td>
<td>Rational Manual Tester</td>
<td>Rational Functional Tester</td>
<td>⚡</td>
</tr>
<tr>
<td>Test execution</td>
<td>Rational Manual Tester</td>
<td>Rational Functional Tester</td>
<td>⚡</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rational Performance Tester</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rational Manual Tester</td>
<td></td>
</tr>
<tr>
<td>Test reporting</td>
<td>⚡</td>
<td>⚡</td>
<td>⚡</td>
</tr>
</tbody>
</table>

**ClearQuest test management object model terminology**

The ClearQuest test management uses the following terminology to define the object model.

• **Test plan:** Groups test cases hierarchically. A test case may belong to only one test plan.

• **Test case:** Does not contain the scripting or the execution instructions.
- **Configured test case**: A test case with a configuration applied.
- **Configuration**: Defines the context of a test case, such as operating system, machine specification, software version, and so on.
- **Iteration**: IRUP iteration, phase, cycle, and so forth.
- **Test suite**: Enables sequential execution of test scripts.
- **Test script**: An asset in IBM® Rational® Manual Tester, IBM® Rational® Functional Tester, or IBM® Rational® Performance Tester. Contains the actual script, or execution instructions. A configured test case is associated with a test script.
- **Asset registry**: Stores the file locations of all of the assets related to ClearQuest test management.

Figure 2 shows the ClearQuest test management object model.

**Figure 2. The ClearQuest test management object model**

As test leader or manager creates a **test plan**, and the plan evolves through the states (or stages) illustrated in Figure 3. The plan is in **Draft** state during the writing stage, in the **Foreview** state during the review process, and then progresses to the **Approved** state.

**Figure 3. States of the test plan as it evolves**
Each test case can be defined by states, too. The test case is in Draft state while the team is still defining it, then it changes to Planned when it is finalized. (See Figure 4.)

**Figure 4. States of the test case as it develops**

![Diagram showing states of the test case]

The corresponding configured test case is in the Draft state while it is in development, then it changes to the Implemented state, which indicates a successful execution. The configured test case can be marked Blocked if a defect or other issues prevent the completion of the test case. (See Figure 5.)

**Figure 5. States of a configured test case**

![Diagram showing states of a configured test case]

As Figure 6 shows, the states for a test suite are the same as for a configured test case, because a suite represents a group of configured test cases.

**Figure 6. States of a test suite (same as for a configured test case)**

![Diagram showing states of a test suite]

**Executing test cases from within ClearQuest test management**

ClearQuest test management provides the ability to invoke the appropriate testing tool for the configured test case. It includes the ability to read test logs created from the testing tool and to create test log results for the configured test case. Enterprise-level test management with ClearQuest supports Rational Manual Tester, Rational Functional Tester, Rational Performance Tester, and the Eclipse TPTP testing framework tools (JUnit, manual, and URL).
Note:
Test execution is supported in the ClearQuest Client for Eclipse (an Eclipse plug-in), which is the only client that can execute all supported testing tools. The ClearQuest client (the Eclipse RCP, or rich client platform) supports only Rational Manual Tester scripts.

With ClearQuest test management, you can also create test cases and configured test cases that use unsupported testing tools. This is accomplished by using the test type called **ExternalFile**. You can use a ClearQuest form to enter test results manually for a configured test case with a test type of ExternalFile. For these unsupported tools, you execute the configured test case, and then create the test results record by entering the date and time that the test was run, the verdict, and other pertinent information about the test.

Notes:

- SPI Dynamics (see Related topics) has integrated their testing tool, QAInspect, with ClearQuest for enterprise test management. After their adapter is installed, QAInspect appears as an additional test type when you associate an implementation with a test case or configured test case.
- Ring Zero has developed adapters for Mercury Quick Test Pro and Mercury WinRunner. When installed, these adapters enable enterprise-level test management with ClearQuest to associate and execute a test case or a configured test case with a Mercury Quick Test Pro or WinRunner script. (See Related topics.)

**Accessing the test case script:** The ability to automate your test script is based on the individual testing tool. Thus, ClearQuest test management needs to be able to access the file system or the location of IBM® Rational® ClearCase® where the test scripts are located. After you have written the test case or configured test case, you can modify it and associate it with the script. ClearQuest test management maintains a pointer to the location of the script that is used during test execution.

**Monitoring and analyzing test results**

ClearQuest provides a set of built-in queries, charts, and graphs for analysis of test management, from planning through execution. You can see the impact that a requirement change or defect fix will have on test cases. You can also see the status of planned and already implemented test cases, as well as passed and failed results in a given iteration interval. More importantly, you can see the test progress trend over time for the number of test cases and number that have of passed or failed.

The screen capture in Figure 7 shows some of the built-in charts and queries.
In addition, you can build your own customized queries within ClearQuest test management. ClearQuest also supports other reporting tools, such as Crystal Reports, IBM Rational ProjectConsole, and IBM Rational SoDA®, if you prefer to use one of those products to build customized queries, charts, or graphs.

**Steps for setting up ClearQuest test management**

In any given release of DB2 for Linux, UNIX, and Windows (hereafter, referred to simply as DB2), there are a significant number of solutions, each of which is to solve a customer problem or fulfill a requirement. Some solutions are more complex and include multiple line items (work items), while others involve only one item. This case study evaluated the feasibility of using ClearQuest test management to track the function verification test progress at the release, solution, or line item level. DB2 is supported on multiple platforms, including UNIX, Linux, and Windows. Thus, the idea that each test case needs to be associated with a configuration is quite relevant, because the DB2 Function Verification Test team runs each test case against a different configuration (supported operating system platforms or supported system configuration). We chose to stay with the ClearQuest for Eclipse client for this project, because it offers the richest functionality.
This rest of this section describes how we set up the basic ClearQuest test management environment.

**Step 1. Customize the ClearQuest test management object model for DB2 development**

Based on the DB2 process, we mapped the ClearQuest test management object model to our needs. We configured it for all of the platforms that we use and the different modes within each platform. For example, DB2 supports massively parallel processing (MPP) architecture and can run on a single partition (EE) or multiple partitions (EEE). DB2 also supports symmetric multiprocessing (SMP) single or multiple processors within a single system. We used an asset registry to represent a solution in a given release. (See Figure 8.)

**Figure 8. The DB2 customized ClearQuest test management object model**

For each solution in a release, we can have either one or more test plans for each line item or a test plan for solution-level integration testing, or both. Each test plan consists of multiple planned test cases. Each test case is then associated with one of the supported system configurations, for example: IBM AIX® UNIX EE environment, IBM AIX UNIX EEE environment, Microsoft Windows EE environment, or Microsoft Windows EEE environment. A test script can then be associated with the configured test cases. It is possible to reuse the same test script for different configured test cases.

Each configured test case needs to be associated with at least one iteration. We define an **iteration** as a phase, a milestone, or a period of time. For example, the alpha build test cycle could be a test iteration. It has a name and a beginning and end date. It belongs to an asset registry. This means that we must define a set of iterations or milestones for each solution.

The Beta 2 version of ClearQuest test management did not have a published API on how to implement a customized adapter to enable ClearQuest test management to associate and execute a configured test case with other testing tools. Therefore, we took the approach of using Rational Manual Tester, and we associated a test file that contained only the test unit within each configured test case. The test was executed outside of the ClearQuest test management setup, which meant that the tester could use Rational Manual Tester to mark the success or failure of each test case.
During this evaluation, we also wanted to use the test suite to represent testing of different components of the product. Different configured test cases in each solution may test the same area or component of the product. However, the limitation of not being able to associate configured test cases from different asset registries to form a suite prevented us from implementing this.

With the schema in place, we started setting up the backend server.

**Step 2. Set up DB2 as the backend server for ClearQuest test management**

To create a backend server for ClearQuest test management, we selected an IBM AIX system and a user ID on the machine to create two databases. One of the databases was for the schema repository and the other one was a user database. For this purpose, we used `desktop` as the name for the schema repository and `notebook` for the name of the user database.

**Note:**

From here on, we describe the required tasks in a way that you can follow along to get hands-on experience in setting up ClearQuest test management.

After issuing a `create database` command to create the two databases, you might need to configure the user database to add a new buffer pool, and then set the page size to 16k by following these two steps:

To create a buffer pool:

1. Connect to the database, and then issue the commands shown in **Listing 1**.

**Listing 1. Create a new buffer pool**

```sql
db2 connect to [user database name] user [user name] using [password]
db2 create bufferpool [buffer pool name] immediate size 250 pagesize 16k
```

2. Create a new table space that uses the new buffer pool. For the AIX platform, see **Listing 2**.

For the Windows platform, see **Listing 3**.

**Listing 2. Create a new table space that uses the new buffer pool on the AIX platform:**

```sql
db2 connect to [user database name] user [user name] using [password]
db2 "create regular tablespace ts4cq pagesize 16k managed by system
    using ('/home/regres1/regres1/NODE0000') extendsize 16
    overhead 12.67 prefetchsize 16 transferrate 0.18 bufferpool
    [buffer pool name] dropped table recovery off"
```

**Listing 3. Create a new table space that uses the new buffer pool on Windows:**

```sql
db2 connect to [user database name] user [user name] using [password]
db2 "create regular tablespace [table space name] pagesize 16k managed
    by system using ('c:\db2\node0001') extendsize 16 overhead 12.67 prefetchsize 16
    transferrate 0.18 bufferpool [buffer pool name] dropped table recovery off"
```
Step 3. Set up a license server

Set up a license server so that users can point to the license server to use the ClearQuest test management client. After following a few simple steps with the GUI on the license server installation launch pad, you will be asked to import your license key from a file. After you import the license key, you should see something like what Figure 9 shows.

Note:
Your license type, expiration date, license term, and quantity are based on the license that you purchased, so they will be somewhat different from what you see here.

Figure 9. Example of a license server setup

Step 4. Create a schema repository and configure the user database

To create a schema repository, your administrator must first launch the ClearQuest Maintenance tool and enter the host name of the backend DB2 database server, the schema repository database name, and the user ID and password to use to connect to the DB2 database. You can choose a name for the schema repository when you are creating the connection profile. The actual backend database can have a different name. (In this example, the database is called desktop, and the schema repository is called cqtm.)

You can then export this profile so that other users can simply import the profile. In a deployed environment, the user ID and password of the backend database are generally hidden from end users. For that reason, the ClearQuest test management administrator creates a profile and distributes it to end users so they can set up individual clients.

Use the Rational ClearQuest Designer to start configuring your empty backend DB2 user database, as Figure 10 shows.

Figure 10. Start creating a new user database
To register your user database with ClearQuest test management, follow these steps, which are illustrated in Figures 11 through 13:

1. Create a new logical database (Figure 11).
2. Link that database to the user database that you created earlier (Figure 12). In this example, the user database is called notebook.
3. Accept the default values for timeout and poll interval. (No illustration is shown here for Step 3.)
4. Associate the user database with a schema (Figure 13).

**Figure 11. Step 1 of 4, create a logical database**

![Create logical database](image1)

**Figure 12. Step 2 of 4, link the new database to the user database**

![Link database](image2)
Step 5. Create a new user account in the ClearQuest test management tool

To create a new user account, you need administrator privileges. By default, ClearQuest test management uses an administrator ID of `admin` with an empty password, but be sure to change this to require a password that you choose. You need to use this ID and password to create accounts for other users and their initial passwords.

The administrator needs to launch the ClearQuest User Administration tool that allows the authorized user to add users, edit properties of existing users, and change privileges. The steps that follow and the diagrams in Figures 14 through 18 show how to add a user named Wenjie Wu to the user database.

1. First, log into the system as `admin`, using the administrator password that you created (default was no password required). (See Figure 14.)

Figure 14. Log in through the ClearQuest User Administration Login window

2. Click **User Action** and select **Add User** from the dropdown menu (Figure 15).
3. An Add User window will display. Enter this information (also shown in Figure 16):

- **Login:** `wenjiewu`
- **Password:** Type a password for this user, and then retype it in the net field to confirm.
- **Name:** `Wenjie Wu`
- **E-mail:** `wenjiewu@ca.ibm.com`
- **Phone:** `(905)413-2199`
- **Mastership:** Select `Local` from the dropdown menu

4. Under Privileges, at the right, select **Active User** and **All Users/Groups Visible**.

5. Under Subscribe, select **MAR** as the database to use, and then click **OK**. The new user should be listed when the Rational ClearQuest User Administration window reappears.

---

**Figure 15. Start adding the new user**

![Add User window](image)

**Figure 16. Enter the user and database information**

![Add User form](image)
6. Commit this change to your user database by clicking **DB Action**, and then selecting the **Upgrade** option from the dropdown menu that displays (Figure 17).

**Figure 17. Commit the changes to your user database**

![Image of the ClearQuest User Administration window](image)

When you see a notification that your user database has been upgraded (Figure 18), the user is officially created in the ClearQuest test management tool.
All of the previous steps are performed by the ClearQuest test management administrator. After the administrator has set up the database backend server, license server, schema repository, and user database, the next task is to get each user to set up their individual ClearQuest test management clients by using the ClearQuest test management user ID that your administrator created in Step 5.

**Step 6. Set up ClearQuest test management for an Eclipse client**

Installing ClearQuest test management for an Eclipse client is relatively easy. The installation wizard guides you through the initial steps, and then you can complete the process with these steps:

1. Select **Point to a Rational License Server to get my licenses.**
2. Enter the information for the license server that was set up in Step 3.

The screen captures in Figure 19 and Figure 20 show you these two steps, where ClearQuest is setting up the license for the client. (In this example, the license server name is osgiliath.torolab.ibm.com.)
When the installation is complete, you will need to configure a connection to the backend database. ClearQuest test management supports connection pooling, whereby all connections to the backend database can use one shared user ID and password. This is the approach that we used in this case study.

To set up the connection, import the Profile file that you obtained from the ClearQuest test management administrator by following these steps:
1. From the Rational ClearQuest Maintenance tool, click **File > Import Profile**. (See Figure 21.)

**Figure 21. Import the Profile file**

2. Select the .ini file provided by your administrator (**cqtm** in this example). The appropriate connection information to import should already be filled in within the Schema Repository Properties fields.

3. Then click **Finish**. (See Figure 22.)

**Figure 22. Select the cqtm.ini file**
You should then see the screen that Figure 23 shows with the connection status. Notice that a log file is created for reference in case of errors.

**Figure 23. Connection status**

![Connection Status Screen]

After your client software has established a connection with the ClearQuest test management server, you can set up a user to connect to the actual database schema. The following steps show how to add a connection for the user ID *wenjiwu* to the existing ClearQuest test management schema repository, `cqtm`.

1. For the ClearQuest Connection Management window, select `cqtm` under Database Set, and then click **Add connection** (Figure 24).
2. In the Schema Repository display, select the appropriate schema from the dropdown menu (*cqtm* in this example), and then click **Next** (Figure 25).

3. Enter your **user ID** and click **Finish** (Figure 26).
Figure 26. Enter your user ID

When prompted, enter your password, and then click OK (Figure 27).

Figure 27. Enter your password

You should now be connected to the cqtm schema, as Figure 28 shows.
Figure 28. Verify connection to the schema

![Connection Management](image)

Step 7. Remind users to change their passwords

This requires that they have privileges to change their passwords, of course. See Figure 15 for how to grant that permission. Because the initial password is assigned by the administrator, advise your users to change their passwords when they log in for the first time by following these steps (also see Figure 29):

1. Launch ClearQuest User Administration.
2. Enter the user's user name and password, and click OK.

Figure 29. Enter the user's login information

![User Administration](image)

2. Right-click on the user's login name and select Edit User (Figure 30).
Figure 30. Choose which user's privileges to change

3. Change the user's password, and then click **OK** (Figure 31).
Step 8. Implement the object model and enter the test status

This section describes several tasks that you need to accomplish to complete the two parts of this step:

- Create and then expand an asset registry for each solution
- Create one or more configuration attributes
- Create one or more TM configurations
- Create one or more iterations
- Create one or more TM test plans and change the states
- Create one or more TM test case and change the states
- Create one or more configured test cases and change the states
- Associate a script with the configured test case
- Run your test case and check results
- Store the test results
- Create and analyze test result reports

Before you begin, click View on the ClearQuest toolbar, and mark the options that you want to display in the pane below. Now you're ready to work through the tasks on your to-do list.

1. Create an asset registry for each solution
   A. Click File > New > TMasset Registry,
   B. Enter the name of your asset registry, and then click OK.

2. Expand the asset registry
A. Right-click **File Location**. The Register File Location Wizard will display.
B. Select **Asset Registry name**, and then click **Next**.
C. Type **name** as the label.
D. Find the **File Location** and **Log Location** folders.
   - **File Location** is a shared folder on the client machine or on the server that contains the test script from Rational Functional Tester or Rational Manual Tester.
   - **Log Location** is an empty folder where the log files are created during the execution of a configured test case.

3. Create one or more configuration attributes
   A. **Click File > New > TM Configuration Attribute**.
   B. After you create the configuration attribute, enter the **name of your configuration attribute**, and then right-click on **name** and enter a configuration value.

4. Create one or more TM configurations
   A. Click **File > New > TM Configuration**,
   B. Enter the **name of your TM configuration**.
   C. To add a configuration value, click **Add**, and then click **Search**.
   D. From the displayed text, select one of the configuration attributes that you created, and then click **OK**.

5. Create one or more iterations (a period of time that you want to track)
   A. Click **File > New > TM Iteration**.
   B. Enter this required information: **iteration name**, **start date**, and **end date**.
   C. From the dropdown list, choose the asset registry that you created.

6. Create one or more TM test plans (TMTestPlan)
   A. Click **File > New > TMTestPlan**.
   B. For the headline, enter your TM test plan name, and then select items from the dropdown list for **Owner**, **Priority**, and **Asset Registry** (the name of the TM asset registry).
   C. Add an iteration by clicking **Add** and then **Search**, and then click the **name of the iteration**.

**Tip:**
You can create more than one test plan for an asset registry, and you can create a child TestPlan under a parent TestPlan in a hierarchical manner.

7. Change the state of TMTestPlan
   A. Right-click **Change state**, select **Ready for Review**.
   B. Then change the state to **Approve**. (The default state is Draft.)

8. Create one or more TM test cases
   A. Click **File > New > TM Test Case**.
   B. Enter the **name of your TM test case**.
   C. Under Test Case Name, select a priority.
D. Associate your test case with a test plan by clicking **Add**, and then clicking **Add**.
E. Select the iteration name, and then click **OK**.

**Tip:**
You can create one or more TestCases under the same TestPlan.

9. Change the TestCase state from **Blocked** to **Planned**. (The default state is Draft.)

10. Create one or more configured test cases for each TM test case
    A. Click **File > New > TMConfigured Test Case**.
    B. Enter the **name of your configured test case**.
    C. Under Configured Test Case name, select a priority for the configuration by clicking **Search Configuration Name for Test Case**
    D. Click **Select**, select the TestCase name, and then click **OK**.

11. Associate a script with the configured test case
    A. Right-click **Configured Test Case name**, and select **Associate Script** from the dropdown menu.
    B. For Test Type, select **Rational Manual Tester**.
    C. Select **Test script**, and then click **Finish**.
    D. Change the state of the test case to **Implemented** by right-clicking on the name of the configured test case, and then selecting **Change state**.

12. Run your test case and check results
    A. Now, run your test case.
    B. After executing the configured test case, you will see **Test Result** in the right pane. Check the box for the test case that you just ran, right-click on it, and select **Open log**.
    C. Rational Manual Tester will show the **Test Execution Details**, and you will be able to see the verdict.

13. Store the test results
    A. Right-click on **Test Result**, and then click **Commit**.
    B. The test result will move from Uncommitted Results to Recently Committed Results.

14. Create and analyze test result reports
    A. Go to ClearQuest Navigator.
    B. Create reports, queries, and charts to analyze your test results.

**Step 9. Review your test progress reports**

After you have completed the setup, it is time to see what kind of information you can glean as the fruit of your labor. Data is collected over a period of time, and you can generate the following kinds of reports, either by submitting a query or creating a chart.

**Note:**
In the context of this case study, the **Draft** state indicated that the test case was written but had not been executed yet. **Implemented** indicated that the test case was successful, and **Blocked** meant that the test case was failing because of some outstanding defect.

- Monthly report of effort remaining and trend of the testing
  1. Using the built-in chart, select **Expand TMCharts** (Figure 32).
  2. Select **CTC > State Trend**.

**Figure 32. Monthly chart of effort remaining and trend of the testing**

![Chart showing state change trend](image)

- Report of test cases planned, written, and executed as of today, using customized queries (Figure 33)

**Figure 33. Query about test cases planned, written, and executed to date**

![Query results](image)
• Weekly progress report of test cases planned, written, and executed, using customized queries (Figure 34)

**Figure 34. Query for weekly progress report of test cases planned, written, and executed**

![Query for weekly progress report](image)

• Report of successful and failed test cases as of today, using the built-in chart (Figure 35).

**Figure 35. Report of successful and failed test cases to date**

![Report of successful and failed test cases](image)

• Report of successful and failed test cases by weekly progress, using the built-in chart (Figure 36).
Problems encountered and workarounds

We experienced only two problems, and we found workarounds described here.

Problem 1. Changing an expired user database password

On each ClearQuest for Eclipse client, each user needs to set up the connection again, using a new version of the profile created by the administrator. Therefore, the administrator needs to create a new connection profile and distribute it to each user, and then reconfigure the user database after following these steps:

1. Launch ClearQuest designer.
2. Log in as the administrator.
3. Click Cancel when you are prompted to open a schema.
4. Click the Database tab, and then select Update user database properties.
5. Ensure that the logical database name is selected (in this example, MAR), and then click Properties.
6. Change the password field to a new password.
7. Click Update.
8. Click OK.
9. Close the ClearQuest designer.

Problem 2. Associating a test script with a configured test case

If you get the error message about an invalid value for the URL field when you are attempting to associate a Rational Manual Tester test script with a configured test case (see the screen capture in Figure 37), here’ a way to work around that problem:
1. Add the File field to the TMExternalFile record type, and move the Unique Key from the URL into the File field.
2. Then change the reference to the URL field in the both the TMTestCase and TMConfiguredTestCase forms in the File field.

Figure 37. Invalid URL value error message

Results and future development recommendations

ClearQuest test management proved to be very useful in tracking testing progress. It also integrates tightly with Rational ClearCase and ClearQuest (defect). If your product code and test case versions are already controlled by ClearCase, then you can easily kick off an execution if the test cases are written in Rational Functional Tester or Rational Performance Tester format. If your defect and test repository is managed on the same server as ClearQuest, you will be able to work with test cases and defect reports within the same client. The Rational TestManager piece naturally fits into any environment already using ClearCase and ClearQuest in software development.

In the long run, to deploy ClearQuest test management in a DB2 environment, we will need the ability to create our own, customized adapter so that our Perl-based test harness can be executed using this adapter, just like test cases written in Rational Functional Tester or Rational Performance Tester, and so that the result log will be written into a format that can be parsed and understood by the ClearQuest test management too.

We have submitted requests to the ClearQuest test management development team to consider the following additional capabilities or options for future releases:
• **Group asset registry by release.** Scenario: We have Release 1 of a product, and we create 10 registries. For Release 2, we create 10 different registries. We need a way to be able to group them.

• **Use two Draft states for configured test cases.** This will make the states more descriptive. For example, a test case can be either not written or written but not run yet.

• **Create two defects to associate with Configured Test Case 1.** When we found Configured Test Case 2 that fails with the same defect, we cannot associate the failure with the same defect without actually creating a test log and associating a test script with the test log.

• **Add a defect number field for default queries.** By default, the state queries do not include a field to display the defect number for configured test cases. Currently, we can get around this by logging in as the administrator and changing this display field, but that has to be done for each solution (assetRegistry), which can be time-consuming.
Related topics

- FAQ: Enterprise Test Management with IBM Rational ClearQuest, Version 7
- Understanding the Rational ClearQuest Client, available in the Rational Software Information Center

© Copyright IBM Corporation 2007

Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)