Integrating IBM Rational Build Forge with IBM Rational ClearCase and IBM Rational ClearQuest

Setup requirements and adaptor templates

John H. Gough

July 13, 2011

Note

Before using this information and the product it supports, read the information in " Notices," at the end of this document.

Introduction

You can integrate IBM Rational Build Forge with other IBM Rational ClearCase and IBM Rational ClearQuest by using a command-line interface or Rational Build Forge adaptors. This document supplements Build Forge documentation and includes specific information on the sample adaptor templates provided with the product.

The paper provides the following information:

- Overview of Rational Build Forge command-line integrations and adaptor integrations
- Rational ClearCase adaptor integration
- Rational ClearQuest adaptor integration
- Setup topology – the relationship of the Build Forge console host and the application host
- Environment requirements – environment variables required by the adaptor template

Overview of Rational Build Forge Integrations

This section discusses the difference between command-line interface integrations and adaptor integrations with Rational Build Forge.

Command-line integration

Rational Build Forge integrates with any application that has a command-line interface. To set up this kind of integration, do the following:

1. Install the Rational Build Forge console
2. Install the Rational Build Forge agent on the application host or a host that can access the application.
3. In the console, create a server resource and a server authentication. Configure the server resource to access the Rational Build Forge agent that you installed.
4. Configure the agent and host environments as necessary for commands to run on the application.
    - Add the user that the agent runs as to the list of users for the application and set the permissions for that user.
    - Set the PATH for the user to include the directories that contain the executable programs for the application.
    - As required: Install and configure a client that is used to run commands in the application. For example, Rational ClearCase and Rational ClearQuest require the use of client applications to run commands.

When the setup is complete, the projects you create in Rational Build Forge can contain steps that run application commands. The step or project is configured to run the commands on the server resource you select and in a directory that you specify. You can control project execution based on pass or fail status of a step or set up log filters to scan the Build Forge log for patterns of output that the commands return. Applications might require additional setup in the project logic. For example, integrating with Rational ClearCase requires that you include steps to create, start, and populate views.

You can use condition steps and loop steps to control what happens in response to a command passing or failing.

A typical use with source control applications is *build avoidance*. A command is run to query whether source code updates have been checked in since the last time a build was run:

- If no updates exist, no build is run.
- If updates exist, a build is run.

Module dependencies can be expressed in the execution logic of project steps. You can control whether a build of a particular module triggers a build of other modules or the entire software project.

You use notification templates to control how groups of project members are notified in response to a build success or failure. In a continuous integration environment, you must notify on failure so that problem code can be fixed. By using a command-line integration, you can notify an entire group of project members. The project members would need to inspect the step log for the project to determine what code caused the failure.

Command-line integration of Rational ClearQuest and Rational Build Forge

Rational Build Forge contains features that support a command-line integration with ClearQuest 7.0 or later. The integration causes build records to be updated in Rational ClearQuest when Rational Build Forge jobs are completed.

For more information, read Integrating Build Forge with ClearQuest on the IBM Support website.

Adaptor integration

With the Rational Build Forge Adaptor Toolkit, you can use an adaptor when you integrate Rational Build Forge with an application. Adaptor behavior is defined by XML elements that are specified by an included DTD. Build Forge runs the adaptor in association with a project step.

Adaptors give you additional tools for integration-based builds:

- Internal condition logic: Based on settings of internal variables
- Command definition: Command definitions can be built using commands and variables.
- Response scanning: You can define patterns to scan for in response to each command. Each line of output is scanned for the specified pattern.
- Dynamic notification groups: Notification can be based on data that is gathered from the application. For example, a set of team members to notify can be defined to include only the members who checked in code changes.
- Script execution: Adaptors can run scripts on the Rational Build Forge host. This capability is independent of scripts and commands run on the application host.

Rational Build Forge provides a set of adaptor template samples to use as a starting point.

Adaptor setup

Adaptors require the following setup:

1. Install the Rational Build Forge console on a host.
2. Install the application to integrate on another host. The application must be a version that Build Forge supports.
3. Install a Build Forge agent is installed on the application server host or a host that can communicate with the application server. Requirements vary with the Rational Build Forge support offered and the client requirements of the application.
4. In Rational Build Forge, do the following:
   A. Define a server resource to access the agent.

B. Define an environment. Populate it with the variables required by the adaptor.
C. Define an adaptor, specifying an adaptor template to use. The adaptor template XML is copied into the adaptor, where it can be further customized as needed.
D. Define a Build Forge project or step to use the adaptor.
    1. Adaptor link: an adaptor link is defined and assigned to a project. When the project runs, the adaptor is run. If the adaptor failes, the adaptor link causes the project to fail and the build tag number to be rolled back.
    2. Dot command: the adaptor is called from within a step using one of these dot commands: .source, .defect, .test, .pack. The adaptor run result (pass or fail) drives the step result.

The specifics of setup can vary according to the needs of the application. Rational ClearCase and Rational ClearQuest each have unique requirements.

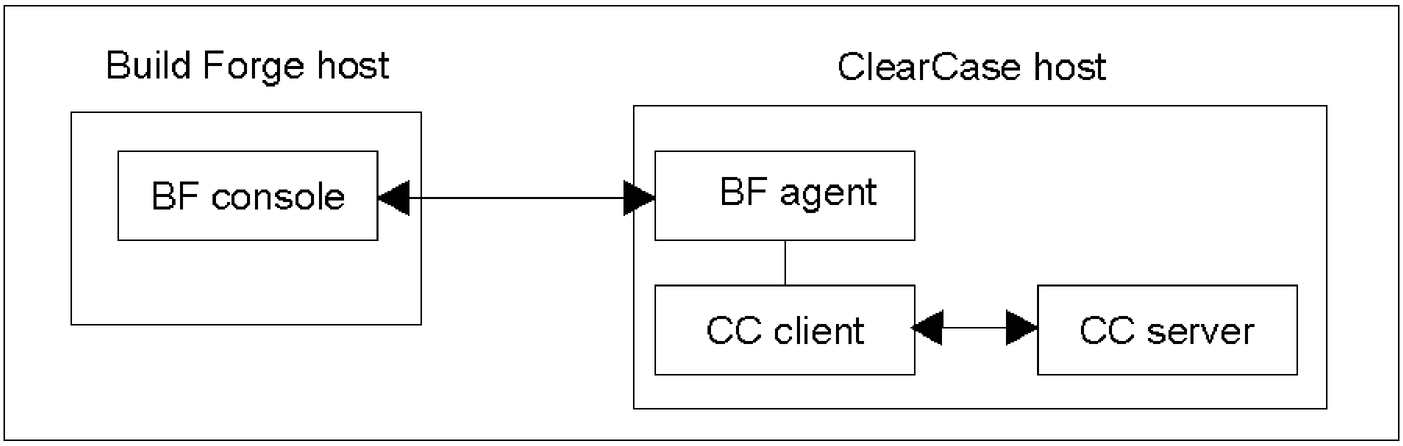
Rational ClearCase adaptor integration

The ClearCase adaptor template samples provide methods of analyzing changes to a baseline. Typically change analysis is used for build avoidance: if a baseline component has not changed, it is not rebuilt.

Setup

Before setting up individual adaptors, set up the products as described in the following steps and as shown in the diagram that follows.

1. Install a Rational Build Forge agent on a host that can connect to the Rational ClearCase server.
2. Install the Rational ClearCase full client on the agent host.
3. Set up the environment for the agent so that commands can be run through the Rational ClearCase client.
4. Determine how to implement and how and when to start the Rational ClearCase views that are required. The provided templates assume the use of dynamic views and include starting that view when they run a cleartool command.



Rational ClearCase Views and the Rational Build Forge features that support them

To access data you use *views* in Rational ClearCase. Two types of view can be used:

- Dynamic views: Dynamic views display the latest version of elements in the versioned object bases (VOBs) that you specify. The data is not copied to your local system. Dynamic views must be started, which updates the local view. You specify the VOBs to mount in the view.
- Snapshot views: Snapshot views are a copy of the repository. You can specify the version of objects to use. Snapshot views must be manually updated to get the latest items that have been checked in.

Dynamic views

Dynamic views show a workspace that is constantly updated whenever anything is changed in the view context that you choose to see. Only the elements that you check out are copied to your workspace. You must check them in to make your changes available to other users.

When you use dynamic views, you use three specifications to access the data:

- **View**: Defines the list of versions to present. The list is specified in a configuration specification. The view must be started. It is represented on your local disk as a root directory for content. That directory is also called the *view root*. The commonly used directories are as follows:
  - Windows® systems: drive M:, a shared drive
  - UNIX® or Linux® systems: /view, a mounted file system
- **View context**: Defines a directory of content. A *view tag,* defined in Rational ClearCase, specifies the directory. The directory becomes available to you when the view is started.
- **VOB** (versioned object base): Defines a subdirectory of the view context. VOBs are defined in Rational ClearCase. You mount a VOB in the view context. The VOB subdirectory contains the versioned artifacts (source files and other artifacts).

Paths to a VOB on your local machine are built up out of those three items:

Example path for Windows systems, using the default:
`M:\`*`view_tag`*`\`*`vob_name`*

Example path for UNIX and Linux systems, using the default:
`/view/`*`view_tag`*`/`*`vob_name`*

Snapshot views

When you use snapshot views, you explicitly specify the view context and load the data. You then have a local copy of all of the data you specify. To get changes that were checked in after you last loaded it, you must reload the snapshot view.

Support for dynamic views in special environment variables

You can use special environment variables to define and start dynamic views from a Rational Build Forge job.

CLEARCASE_VIEW

This variable starts the specified dynamic view. The view specified in this variable must exist, and the step using this variable must be set to "absolute". On Windows systems, this variable must be used along with the `cc_suppress_server_root` parameter for the agent in bfagent.conf.

_CLEARCASE_VIEWS

This variable specifies a list of dynamic views to start before command execution. Set the value to a comma-separated list of views; for example, "View1,View2,View3".

_CLEARCASE_VOBS

This variable specifies a list of ClearCase VOBs to mount before running a command. Set the value to a comma-separated list of VOBs. Example: "\Vob1,\Vob2,\Vob3".

Support for dynamic views in agent parameters

Agent parameters *must* be set if the agent you are using with Rational ClearCase is running on a Windows system. The parameters are also useful if the view root that you want to use through this agent is different than the view root set by the job. This would happen if you used the job for more than one agent and the specifications must be different on each agent host system.

ccviewroot *root-path*
This parameter specifies the default view root for this host. See Rational ClearCase documentation on init.
The internal defaults are as follows:

- Windows systems: ccviewroot M:
- UNIX or Linux systems: ccviewroot /view

cc_suppress_server_root
If this parameter is set, then the view path is the path set by ccviewroot. If it is not set, then the path that is set in the server definition is appended to the path that is set by ccviewroot. This setting does not need a value. If the parameter is present in bfagent.conf, then it is set.

Rational ClearCase adaptor template samples

The following adaptor template samples are provided for use with ClearCase:
- ClearCaseBaseline
- ClearCaseByBaselineActivities
- ClearCaseByBaselineVersions
- ClearCaseByDate
- ClearCaseByLabel

ClearCaseBaseline

This adaptor template does the following:

1. Gets a list of development streams in an integration stream.
2. Checks for deliveries in progress. The adaptor stops and sets the project to FAILED if there are deliveries in progress.
3. Checks for pending delivery activities. If there are not pending delivery activities, the adaptor fails the project.
   Note: the section about listing activities is written for use on Windows systems. Modify the command definition for cc_actdescribe if the adaptor is running on a UNIX system or Linux system.
4. Writes the list of activities to the BOM report. Activities are represented by ID, Owner, Stream, and Title. Change sets within activities are listed within their activity.

The following table shows the variables that are defined by the adaptor template. All variables must be set in an environment for the project.

Environment variable name	**Description**
CCSERVER	The host that has the Rational ClearCase client and Rational Build Forge agent installed. The value must be a fully-qualified domain name.
INT_STREAM	The integration stream to use. Use the simple name.
PROJECT_VOB	The project VOB to use. This variable is used only with Unified Change Management (UCM) ClearCase. Example: \ProjectVob
UNIXCLIENT	The platform where the client is running. Set to 0 if the client is running on Windows. Set to 1 if the client is running on UNIX or Linux.
VIEW	The view to use. It is assumed to be a dynamic view. It is started by commands in the adaptor.
_CHAR_NATIVE	A variable that is used internally. It must always be set to 1.

ClearCaseByBaselineActivities

This adaptor template does the following:

1. Creates a baseline from the contents of a Rational ClearCase view.
2. Compares the new baseline and the baseline from the previous adaptor execution to identify change activity.
3. For each change activity, writes the following information to the BOM report: activity, files changed, user, date, comments, and version.
4. For each changed file, writes change details (from diff command output) to the BOM report.

The following table shows the variables that are defined by the adaptor template. All variables must be set in an environment for the project.

Environment variable name	**Description**
BASELINE	The baseline to use for comparison.
CCSERVER	The host that has the Rational ClearCase client and Rational Build Forge agent installed. The value must be a fully-qualified domain name.
CurDate	The current date and time. At run time, the adaptor uses a .date command to generate the date in the format that Rational ClearCase requires. Do not change this value.
LAST_RUN	The time and date of the last run of the adaptor. This variable used to compare baselines by date. If the adaptor runs successfully, the value is automatically updated to the current date. If the adaptor does not run successfully, this value is unchanged. The default value is 1-Jan-05.00:00:00. You can test the adaptor by setting this date to a date that precedes known changes.
PROJECT_VOB	The project VOB to use. This variable is used only with Unified Change Management (UCM) ClearCase. Example: \ProjectVob
UNIXCLIENT	The platform where the client is running. Set to 0 if the client is running on Windows. Set to 1 if the client is running on UNIX or Linux.
VIEW	The view to use. The view must be available. Dynamic views must already be loaded and started.
_CHAR_NATIVE	A variable that is used internally. It must always be set to 1.

ClearCaseByBaselineVersions

1. Creates a baseline from the contents of a Rational ClearCase view.
2. Compares the new baseline and the baseline from the previous adaptor execution to identify changed files.
3. For each changed file, writes the following information to the BOM report: file name, version, date, user, and comments.
4. For each changed file, writes change details from diff command output to the BOM report.

The following table shows the variables that are defined by the adaptor template. All variables must be set in an environment for the project.

Environment variable name	Description
BASELINE	The baseline to use for comparison.
CCSERVER	The host that has the Rational ClearCase client and Rational Build Forge agent installed. The value must be a fully-qualified domain name.
CurDate	The current date and time. At run time, the adaptor uses a .date command to generate the date in the format that Rational ClearCase requires. Do not change this value.
LABEL	The label to use for comparison.
LAST_RUN	The time and date of the last run of the adaptor. This variable used to compare baselines by date. If the adaptor runs successfully, the value is automatically updated to the current date. If the adaptor does not run successfully, this value is unchanged. The default value is 1-Jan-05.00:00:00. You can test the adaptor by setting this date to a date that precedes known changes.
PROJECT_VOB	The project VOB to use. This variable is used only with Unified Change Management (UCM) ClearCase. Example: \ProjectVob
UNIXCLIENT	The platform where the client is running. Set to 0 if the client is running on Windows. Set to 1 if the client is running on UNIX or Linux.
VIEW	The view to use. The view must be available. Dynamic views must already be loaded and started.
VOB_PATH	The component VOB to use. It can specify a subdirectory. Use a comma-separated list for multiple names.
_CHAR_NATIVE	A variable that is used internally. It must always be set to 1.

ClearCaseByDate

This adaptor template does the following:

1. Queries a Rational ClearCase view for changes between two dates. The default dates are the current time stamp and the time stamp of the previous adaptor execution.
2. For each changed file, writes the following information to the BOM report: file name, version, date, user, and comments.
3. For each changed file, writes change details (from diff command output) to the BOM report.

The following table shows the variables that are defined by the adaptor template. All variables must be set in an environment for the project.

Environment variable name	**Description**
BASELINE	The baseline to use for comparison.
CCSERVER	The host that has the Rational ClearCase client and Rational Build Forge agent installed. The value must be a fully-qualified domain name.
CurDate	The current date and time. At run time, the adaptor uses a .date command to generate the date in the format that Rational ClearCase requires. Do not change this value.
LABEL	The label to use for comparison.
LAST_RUN	The time and date of the last run of the adaptor. This variable used to compare baselines by date. If the adaptor runs successfully, the value is automatically updated to the current date. If the adaptor does not run successfully, this value is unchanged. The default value is 1-Jan-05.00:00:00. You can test the adaptor by setting this date to a date that precedes known changes.
PROJECT_VOB	The project VOB to use. This variable is used only with Unified Change Management (UCM) ClearCase. Example: \ProjectVob
UNIXCLIENT	The platform where the client is running. Set to 0 if the client is running on Windows. Set to 1 if the client is running on UNIX or Linux.
VIEW	The view to use. The view must be available. Dynamic views must already be loaded and started.
VOB_PATH	The component VOB to use. It can specify a subdirectory. Use a comma-separated list for multiple names.
_CHAR_NATIVE	A variable that is used internally. It must always be set to 1.

ClearCaseByLabel

This adaptor template does the following:

1. Creates and applies a label to the contents of a Rational ClearCase view.
2. Compares the new label and the label from the previous adaptor execution to identify changed files.
3. For each changed file, writes the following information to the BOM report: file name, version, date, user, and comments.
4. For each changed file, writes change details from diff command output to the BOM report.

The following table shows the variables that are defined by the adaptor template. All variables must be set in an environment for the project.

Environment variable name	**Description**
BASELINE	The baseline to use for comparison.
CCSERVER	The host that has the Rational ClearCase client and Rational Build Forge agent installed. The value must be a fully-qualified domain name.
CurDate	The current date and time. At run time, the adaptor uses a .date command to generate the date in the format that Rational ClearCase requires. Do not change this value.
LABEL	The label to use for comparison.
LAST_RUN	The time and date of the last run of the adaptor. This variable used to compare baselines by date. If the adaptor runs successfully, the value is automatically updated to the current date. If the adaptor does not run successfully, this value is unchanged. The default value is 1-Jan-05.00:00:00. You can test the adaptor by setting this date to a date that precedes known changes.
PROJECT_VOB	The project VOB to use. This variable is used only with Unified Change Management (UCM) ClearCase. Example: \ProjectVob
UNIXCLIENT	The platform where the client is running. Set to 0 if the client is running on Windows. Set to 1 if the client is running on UNIX or Linux.
VIEW	The view to use. The view must be available. Dynamic views must already be loaded and started.
VOB_PATH	The component VOB to use. It can specify a subdirectory. Use a comma-separated list for multiple names.
_CHAR_NATIVE	A variable that is used internally. It must always be set to 1.

Rational ClearQuest adaptor integration

The ClearQuest adaptor template samples provide methods of scanning Rational ClearCase and updating build records in Rational ClearQuest. The adaptor run is usually tied to the success or failure of builds run in Build Forge. A passed job would result in the adaptor running and updating information in Rational ClearQuest. A failed job would not run the adaptor.

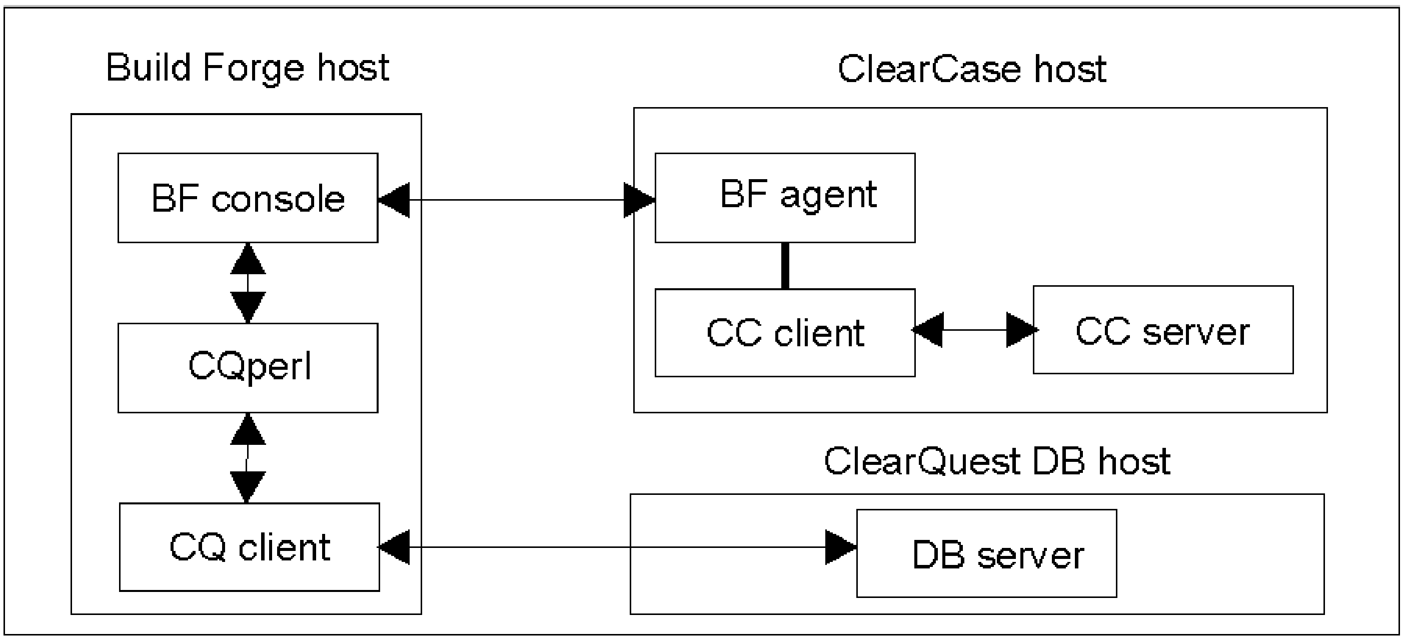
The adaptor requires access to Rational ClearCase and Rational ClearQuest.

- The adaptor must access Rational ClearCase to scan the source. During a job run, the adaptor runs cleartool commands through an agent and the Rational ClearCase client. The adaptor runs commands by using the Rational ClearQuest Perl API (cqperl).
- The adaptor must access Rational ClearQuest to update build records. During the job run, the adaptor runs cqperl scripts directly on the console host. The commands are interpreted by the Cqperl utility and are run through the Rational ClearQuest client. The Cqperl utility and the Rational ClearQuest client must be installed on the Rational Build Forge console host.

Setup

Before setting up individual adaptors, set up the products as described in the following steps and as shown in the diagram that follows.

1. Install a Rational Build Forge agent on a host that can connect to the Rational ClearCase server.
2. Install the Rational ClearCase full client on the agent host.
3. Set up the environment for the agent so that commands can be run through the Rational ClearCase client.
4. Install the Rational ClearQuest full client on the Rational Build Forge console host.
5. Add the cqperl (Rational ClearQuest Perl API) directory to the system path.
6. Define a connection that the Rational ClearQuest client on the Rational Build Forge host can use to access the Rational ClearQuest database. Perform these actions on the Rational ClearQuest client host.
   A. Use the cqreg command to add the database set (cqreg add_dbset).
   B. Use the CQ Maintenance Tool to set up a connection to the ClearQuest database.
7. Determine how to implement the Rational ClearCase views and how and when to start the views that are required.



ClearQuest adaptor templates

The following adaptor templates are provided:

- ClearQuestBaseClearCaseByDate
- ClearQuestClearCaseByActivity
- ClearQuestUCMByDate

ClearQuestBaseClearCaseByDate

This adaptor template does the following:

1. Queries a Rational ClearCase view for changes between two dates. The default dates are the current timestamp and the timestamp of the previous adaptor execution.
2. For each changed file, looks for a CrmRequest hyperlink attribute that identifies a Rational ClearQuest change ID. The adaptor attempts to resolve the defect and add job information to the defect record.
3. For each changed file, writes the following information to the BOM report: the file name, defect ID, defect status, and any Rational ClearQuest errors.

The following table shows the variables that are defined by the adaptor template. All variables must be set in an environment for the project.

Environment variable name	Description
BFSERVER	The Rational Build Forge console host. It must be a fully-qualified domain name.
CurDate	The current date and time. At run time, the adaptor uses a .date command to generate the date in the format that Rational ClearCase requires. Do not change this value.
CQ_USER	The user name to use to log in to the Rational ClearQuest server.

CQ_PASSWORD	Password for the user name in CQ_USER.
LAST_RUN	The time and date of the last run of the adaptor. This variable used to compare baselines by date. If the adaptor runs successfully, the value is automatically updated to the current date. If the adaptor does not run successfully, this value is unchanged. The default value is 1-Jan-05.00:00:00. You can test the adaptor by setting this date to a date that precedes known changes.
UNIXCLIENT	The platform where the client is running. Set to 0 if the client is running on Windows. Set to 1 if the client is running on UNIX or Linux.
VIEW	The Rational ClearCase view to use. The view must be available. Dynamic views must already be loaded and started.
VOB_PATH	The Rational ClearCase component VOB to use. It can specify a subdirectory. Use a comma-separated list for multiple names.
_CHAR_NATIVE	A variable that is used internally. It must always be set to 1.

ClearQuestClearCaseByActivity

This adaptor template does the following:

1. Finds Rational ClearQuest defect records that are associated with a list of Rational ClearCase activities.
2. For each defect record found, the adaptor adds job information and attempts to resolve the defect.
3. Writes the following information to the BOM report: files that are associated with Rational ClearCase activity IDs and the Rational ClearQuest defect status.

The following table shows the variables that are defined by the adaptor template. All variables must be set in an environment for the project.

Environment variable name	**Description**
ACTIVITIES	A space-delimited set of activity IDs. Example: SAMPL0001@\ProjectVob
BFSERVER	The Rational Build Forge console host. It must be a fully-qualified domain name.
CurDate	The current date and time. At run time, the adaptor uses a .date command to generate the date in the format that Rational ClearCase requires. Do not change this value.
CQ_USER	The user name to use to log in to the Rational ClearQuest server.
CQ_PASSWORD	Password for the user name in CQ_USER.
UNIXCLIENT	The platform where the client is running. Set to 0 if the client is running on Windows. Set to 1 if the client is running on UNIX or Linux.
VIEW	The Rational ClearCase view to use. The view must be available. Dynamic views must already be loaded and started.
VOB_PATH	The Rational ClearCase component VOB to use. It can specify a subdirectory. Use a comma-separated list for multiple names.
_CHAR_NATIVE	A variable that is used internally. It must always be set to 1.

ClearQuestUCMClearCaseByDate

This adaptor template does the following:

1. Queries a Rational ClearCase view for changes between two dates. The default dates are the current timestamp and the timestamp of the previous adaptor execution. It uses Rational Unified Change Management (UCM) to produce its results.
2. For each changed file, writes the following information to the BOM report: the file name, defect ID, defect status, and any Rational ClearQuest errors.

The following table shows the variables that are defined by the adaptor template. All variables must be set in an environment for the project.

Environment variable name	**Description**
BFSERVER	The Rational Build Forge console host. It must be a fully-qualified domain name.
CurDate	The current date and time. At run time, the adaptor uses a .date command to generate the date in the format that Rational ClearCase requires. Do not change this value.
CQ_USER	The user name to use to log in to the ClearQuest server.
CQ_PASSWORD	The password for the user name in CQ_USER.
LAST_RUN	The time and date of the last run of the adaptor. This variable is used to compare baselines by date. If the adaptor runs successfully, the value is automatically updated to the current date. If the adaptor does not run successfully, this value is unchanged. The default value is 1-Jan-05.00:00:00. You can test the adaptor by setting this date to a date that precedes known changes.
UNIXCLIENT	The platform where the client is running. Set to 0 if the client is running on Windows. Set to 1 if the client is running on UNIX or Linux.
VIEW	The Rational ClearCase view to use. The view must be available. Dynamic views must already be loaded and started.
VOB_PATH	The component VOB to use. This variable can specify a subdirectory. Use a comma-separated list for multiple names.
_CHAR_NATIVE	A variable that is used internally. It must always be set to 1.

Notices

Trademarks