

Build Forge Samples – Clear Case Project

This example demonstrates a project library that creates and deploys a Java applet using Clear Case and Clear Quest. Other common tools and applications are used throughout the steps of this project. Please note that these examples are not intended to execute successfully without modification to match a specific environment.

Importing the Project

Before you begin, import the file named ClearCase.xml by following the instructions in the README.pdf file in this directory. Be sure to check the boxes marked 'Import Selectors' and 'Import Filters'

NOTE: After the file is imported, a project named 'Clear Case Sample' will exist in the Projects section of your Build Forge installation. It will be necessary to be logged in as a user with Build Engineer Group Access in order to view the project and its subcomponents.

Servers

One selector, 'ClearCase Server' is created with this project. To view it navigate to Servers → Selectors. At this time Server Authentications are not imported, and therefore these are merely examples.

Notice that on the Projects view, the imported project has a Selector of 'ClearCase Server'. The rest of the steps inherit the default (ClearCase Server) selector from the project definition.

Environments

This project uses five environments to build and deploy the application to a web server.

Environment Name	Description
TicTacToe Environment	Contains common environment variables required for ANT and other Java applications. This environment also includes two environments that are specific to ClearCase and ClearQuest.
Deployment Unit Environment	Contains environment variables that point to Perl and the DU tool in Clearcase.
Test Environment	Sets variables that are used by the adaptor called in the final step of the project.
TTAppClearCaseEnv	Contains environment variables pertinent to Clearcase. NOTE: This environment is not called directly in the project, but included (via .include) in the TicTacToe Environment.
TTAppClearQuestEnv	Contains environment variables pertinent to ClearQuest. NOTE: This environment is not called directly in the project, but included (via .include) in the TicTacToe Environment.

All of the variables specified with in the environments are required in order for the third party tools to behave correctly. More information about each individual variable can be found in the documentation for the given product.

Filters

Browsing to Projects → Log Filters shows three filters that were imported with the project. Steps employ filters to determine via the output from the command, if the result is successful or not.

The first filter, 'Ant Filter' searches through the output of the ant commands for the phrase 'BUILD FAILED'. If the phrase is found, the step will be marked as failed.

In the second filter, 'ClearCase Checkin', several conditions are set in order to determine the result of trying to checking in the new project file. If one of the included phrases is present, the step is set to Fail.

The third filter, 'ClearQuest Errors', checks to make sure the Build Forge database has the correct credentials for ClearQuest. If not, the step is set to Fail.

Steps

The commands for each step can be viewed by clicking on the step name in the Projects view. The step names are deliberately descriptive to make it easier to see the overall flow of the project. Below is additional detail about how each step is configured. Note that familiarity with the tools being automated by Build Forge (CVS, Java, and Tomcat) is assumed and these examples are not intended as a tutorial on the tools themselves.

Step	Description
Create Build View	Creates a new build view in the specified directory
Add Load Rules For Viewer	Adds the load rules to this view
Generate BF Baseline	This step stores in memory information about the files in the MODULE directory for use in the Bill of Materials. For more information on the .scan command, see the Build Forge documentation
Lock Integration Stream	Causes cleartool to lock the specified stream
Generate UCM Baseline	Creates a baseline in ClearQuest
Obtain list of UCM Activities	Checks for differences in the code by running a Perl script and storing its output in a variable using .bset. NOTE: For more information on .bset or other dot commands, please see the Build Forge documentation
Shutdown Tomcat Web Server	Uses an ANT target to shutdown Tomcat.
Clean Build Area	Uses an ANT target to clean the build directories
Compile Source	Uses an ANT target to compile the code

Create Deployable War File	Uses an ANT target to compress the compiled java into a war archive (.war file)
Copy War File	The parent step for two threaded steps (following) this step just echo's the intent of the next two steps
Unlock Stream	Unlocks the stream Runs concurrently with 'Restart Tomcat Web Server'
Restart Tomcat Web Server	Calls the command startup.bat from the TOMCAT_HOME/bin directory on a server that meets the criteria for the selector 'Web Server' Runs concurrently with 'Unlock Stream'
Resolve ClearQuest Activity Records	Uses .defect to add information about defects resolved to the build. NOTE: For more information on .defect or other dot commands, please see the Build Forge documentation
Generate BF Checkpoint	Stores information about the current state of the MODULE directory for the Bill of Materials
Checkout Project XML File	Checks the projects \$APPLICATION.xml file out
Export Project XML File	Runs the .export command to create a new version of \$APPLICATION.xml and then copies it to a networked drive NOTE: For more information on .export or other dot commands, please see the Build Forge documentation
Checkin Project XML File	Checks the newly created \$APPLICATION.xml file into the branch in CQ created earlier (step 5 – Create UCM Baseline), also un-check out the file from the main branch.
Generate Deployment Unit Information	Only echo a statement about creating the DU
Create Deployment Record	Calls a Perl script with many arguments to create a record of this deployment.
Deploy TTT to QA Staging Area	Calls a .bat script to remove the outdated war file and application from the Tomcat home directory and then copy the newly made war into its place
Execute Tests	Uses .test to execute adaptor driven tests NOTE: For more information on .test or other dot commands, please see the Build Forge documentation