

Build Forge Samples – WebSphere Deployment

This example demonstrates a very basic project library that creates and deploys a Java application to WebSphere. 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 WebSphereDeploy.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 will exist in the Projects section of your Build Forge installation named 'WebSphere Deployment Sample'. 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

Two new selectors, 'Java Build Server' and 'WebSphere Server', are created with this project. To view them 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 'Java Build Server'. Also notice that when the steps for the WebSphere Deployment Sample project are shown, one of the steps has the 'WebSphere Servers' selector. The rest of the steps inherit the default (Java Build Server) selector from the project definition.

Environments

This project uses two environments to build and deploy the application to a web server. The first, 'WebSphere Environment', contains many environment variables that are required for WebSphere and Clearcase.

The second environment, 'WebSphere Ant Environment' contains all of the environment variables that need to be set in order to use ANT executables.

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 two 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 Failure' 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, 'Test Analysis Filter', several conditions are set in order to determine the outcome of the unit test steps. If the phrases 'Exception', '[F,f]ailure', or '[E,e]rror' are found the step is set to failed. However, if one of the phrases 'Hello World!', 'PASS', or 'OK' are also found the failure is overridden or cleared.

NOTE: The syntax shown in the phrases '[F,f]ailure' and '[E,e]rror' tell the filter that the case of the first letter is unimportant.

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 (cleartool, Java, ANT and WebSphere) is assumed and these examples are not intended as a tutorial on the tools themselves.

NOTE: In this example, many steps are performing similar operations and therefore will be grouped by their step number and relevance.

Step #	Step Name(s)	Description
1	Check Out Files	These are the commands to checkout code from Clearcase using the command line tool, cleartool The directory that the code is checked out into is also named the value of the MODULE variable
2	update version	Looks in the specified directory for a file named banner.html and changes all instances of the string '@Version@' to the major and minor revision numbers (stored in the Tags section of the Projects configuration)
3-12	Call Target ...	Each of these steps calls an Ant target from the build.xml file in order to clean, create, compile and bundle the software being built. Each of these steps uses the special WebSphere Ant Environment in order to access the Ant executables. Note that steps 3-8 are run threaded (simultaneously), but that 9-12 must run only after 3-8 are completed and in the order specified.
13	Create Checkpoint	Run in threaded mode with step 14. Stores information about the current state of the working directory for the Bill of Materials. For more information on the .scan command, please see Build Forge Help documentation.

Step #	Step Name(s)	Description
14	Update ear file	Run in threaded mode with step 13. Deploys the newly created .ear file to the Websphere Servers.
15	Remove View	Remove the view specified by this Build tag