



BEA AquaLogic Enterprise Repository®

Adapter for Apache Ant Guide

Version 3.0
Document Revised: August 2007



Overview

AquaLogic Enterprise Repository's integration with Apache Ant allows automated asset submission as part of the build process.

Suggested artifacts for sharing include:

- Compiled programs
- API documentation such as JavaDoc
- UML models
- Web Service definitions
- Automated test results
- Code coverage reports

Suggested situations in which to submit the results of a build to the repository include:

- Major releases
- Minor releases
- Patch releases
- Prototypes

- Sharing in-progress assets with other groups

Use Cases

Archive and share artifacts that were generated by an Ant build.

Installation/Configuration

1. Add the following jars to the Ant classpath

- activation.jar
- integration-ant.jar
- log4fl.jar
- mail.jar
- openapi-client.jar
- soap.jar

2. Add the RegistryTask definition to the beginning of the build.xml.

```
<taskdef name="RegistryTask" classname="com.flashline.registry.  
ant.RegistryTask"/>
```

3. Add the call to the RegistryTask to your build.xml

- ```
<RegistryTask username="YOUR_USERNAME" password="YOUR_PASSWORD" serverURL="http://SERVER.NAME/
APPLICATION_PATH/messagerouter"
assetTypeID="YOUR_ASSETTYPE_ID" assetName="YOUR ASSET
Name" assetVersion="YOUR ASSET VERSION"
assetDescription="YOUR DESCRIPTION" assetComment="YOUR
COMMENTS" assetFileLocation="URL TO YOUR ASSETS S FILE" />
```

## Example

This sample build.xml for Ant uses the RegistryTask to submit an asset to ALER.

```
<project name="ExampleRegistryTask" default="registry" basedir=".">
 <!-- Example of how to use Ant task to submit an asset to BEA
 ~AquaLogic Enterprise Repository-->
```

```

<taskdef name="RegistryTask" classname="com.flashline.registry.
ant.RegistryTask"/>

<target name="registry">
<!-- Build the project -->

<!-- Package the built project and copy it to a shared location
such as a network drive (unc share) or a web server -->

 <!--Prompt user for ALER Admin password-->
 <input message="Please enter ALER password:"
addproperty="flashline.password"/>

 <!-- Submit the packaged project to the AquaLogic
Enterprise Repository -->
 <RegistryTask
 username="admin"
 password="\${flashline.password}"
 serverURL="http://www.example.com/flashline/messagerouter"
 assetTypeID="118"
 assetName="asset name"
 assetVersion="1.2"
 assetDescription="my description"
 assetComment="my comment"
 assetFileLocation="\file\location" />
</target>
</project>

```

## Parameters

Ant allows the substitution of command line, prompted, and dynamically generated strings for any of the parameters passed into the `RegistryTask`. This eliminates the need to hard code usernames and passwords, and dynamically determines the `assetVersion` based on the type of build.

The `assetTypeID` can be determined by editing an asset of the desired type in the **Asset Editor**. The `assetTypeID` is listed in the status bar at the bottom of the screen.

The `assetFileLocation` can reference a network drive (UNC share), any URL, or a file

inside a repository. Files stored in a repository are stored in the format

```
rep://REPOSITORY_NAME/RELATIVE_LOCATION
```

The `REPOSITORY_NAME` is the same for every file in a repository, while the `RELATIVE_LOCATION` should change based on where artifacts are stored in the repository. The file information plug-in element inside the **Asset Editor** allows experimentation with this format for files inside a repository.