

CORE : Auto Deploy Project Configuration

This page last changed on Oct 15, 2008 by Hebell.

Contents

- [Introduction](#)
- [Project Directory](#)
- [JBoss](#)
- [Ant](#)

Links

- [Auto Deploy Project Installation](#)
- [Build and Deployment Automation Handbook](#)

Introduction

This page details the Project Configuration requirements for conformance to the "Auto Deploy" designation at NCICB. This is a developer's companion to the [Auto Deploy Project Installation](#). Where that is written for someone installing a project designated as "Auto Deploy", this is written to aid and expand upon the [Build and Deployment Automation Handbook](#) for someone wishing to conform to the "Auto Deploy" designation.

In this document, SCM is an acronym for "Source Control Management" and is a generic reference. For any specific project, CVS, SVN, etc, may be used as the project specific SCM software.

Project Directory

The project directory structure must conform as much as practical and possible to that outlined in the [Build and Deployment Automation Handbook](#).

In this document "project/" refers to a specific project root directory (i.e. folder). It is used as a generic reference and should be substituted with the SCM root module name, e.g. when comparing this document to the caDSR Sentinel Tool project directory tree, "project/" refers to "sentinel/".

conf

The source for the configuration files, e.g. XML, properties, etc. All files in this folder begin with "template." if any filter substitution is performed by an Ant <copy> task or if there is the potential for substitution. Files that do not begin with "template." must **not** contain any environment specific especially directory or path names.

deployment-artifacts

The project/deployment-artifacts directory is created by any of the "build*" targets detailed below. It must never be checked into SCM. All artifacts required for a successful and complete product deployment can be found in this directory. The files are organized by the deployment container in subdirectories, e.g. JBoss specific files are in the project/deployment-artifacts/jboss directory. The deployment-artifacts is a staging area. Should any manual steps be required during a deployment because of network issues or other unforeseen circumstances, someone can pull files from this directory and not search throughout the project for required files.

JBoss

This section applies to all web applications deployed to a JBoss container.

conf/log4j.xml

Per the JBoss installation modifications detailed in the [Auto Deploy Project Installation](#), the conf/log4j.xml is built by the project deployment. Place a template.project-log4j-appender.xml and template.project-log4j-category.xml in the project/conf directory. Using the template build.xml for "Auto Deploy" projects, these files are copied with filter substitution to the \${JBOSS.CONF.DIR}/log4j/ directory and merged with all other deployed project appenders and categories.

An example template.project-log4j-appender.xml. Remember to change the "PROJECT_FILE" and "project_log.txt" values. The "txt" extension is used because a web browser will know to show the text without a special reader. Not all web browsers know the ".log" extension traditionally used for this kind of file.

```
<!-- Project appender -->
<appender name="PROJECT_FILE" class="org.jboss.logging.appender.RollingFileAppender">
  <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler" />
  <param name="File" value="@JBOSS.LOG.DIR@/project_log.txt" />
  <param name="Append" value="true" />
  <param name="MaxFileSize" value="1MB" />
  <param name="MaxBackupIndex" value="3" />
  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%d %-5p [%c : %L] %m%n" />
  </layout>
</appender>
```

An example template.project-log4j-category.xml. Remember to change the "gov.nih.nci..." and "PROJECT_FILE" values. Please also set the <priority> appropriate to the project needs.

```
<!-- Project category -->
<category name="gov.nih.nci..." additivity="false">
  <priority value="DEBUG" />
  <appender-ref ref="PROJECT_FILE" />
</category>
```

conf/login-config.xml

To avoid making edits to the conf/login-config.xml, beginning with JBoss 4.0.5, an application can define a dynamic <policy> during deployment. This requires the deployment of a project-login-config-service.xml to the \${JBOSS.DEPLOY.DIR} and a project-login-config.xml to the \${JBOSS.CONF.DIR}. Replace "project-" with the SCM module name, e.g. "formbuilder-", in **all** examples in this section.

In conformance with the naming standard and locations used for an "Auto Deploy" project, the source files must be placed in the project/conf directory and begin with "template.".

An example template.project-login-config-service.xml. Remember to change "project-login-config.xml" and "projectDynamicLoginConfig".

```
<?xml version="1.0" encoding="UTF-8"?>
<server>
  <classpath codebase="${jboss.server.lib.url:lib}" archives="*" />
  <mbean code="org.jboss.security.auth.login.DynamicLoginConfig"
name="jboss:service=projectDynamicLoginConfig">
    <attribute name="AuthConfig">project-login-config.xml</attribute>
    <!-- The service which supports dynamic processing of login-config.xml configurations. --
  >
  <depends optional-attribute-
name="LoginConfigService">jboss.security:service=XMLLoginConfig</depends>
```

```

        <!-- Optionally specify the security mgr service to use when this service is stopped to
flush the auth caches of the domainsregistered by this service. -->
        <depends optional-attribute-
name="SecurityManagerService">jboss.security:service=JaasSecurityManager</depends>
    </mbean>
</server>

```

An example template.project-login-config.xml. Remember to change "**projectDomain**".

```

<?xml version='1.0'?>
<!DOCTYPE policy PUBLIC "-//JBoss//DTD JBOSS Security Config 3.0//EN" "http://www.jboss.org/j2ee/
dtd/security_config.dtd">
<policy>
    <!-- Project policy -->
    <application-policy name="projectDomain">
        <authentication>
            <login-module code="gov.nih.nci.ncicb.cadsr.common.security.jboss.DBLoginModule"
flag="required">
                <module-option
name="ServiceLocatorClassName">gov.nih.nci.ncicb.cadsr.common.servicelocator.ejb.ServiceLocatorImpl</
module-option>
            </login-module>
        </authentication>
    </application-policy>
</policy>

```

Copy the template files to the deployment-artifacts/jboss directory using the normal filtering done with all other template files.

As noted above the **project-login-config-service.xml** must be deployed in the **`\${JBOSS.DEPLOY.DIR}** and the **project-login-config.xml** must be deployed in the **`\${JBOSS.CONF.DIR}**. These destination references are the standard "Auto Deploy" properties as detailed in the [Auto Deploy Project Installation](#).

Ant

The Ant project file is always named build.xml. A complete build and deploy is done using the command line:

```
ant -f build.xml -D"PROP.FILE=..." build-all deploy
```

The PROP.FILE is the name of the file that contains the build and deploy properties documented in [Auto Deploy Project Installation](#).

Tasks & Targets

The following public targets are required. Private targets are named with a dash (-) as the first character to prevent execution from a command prompt.

- build-all
Performs the clean, build-product, javadoc and package-download targets.
- build-product
Creates all EAR, WAR, JAR, etc binary files as required for a complete software deployment and places the files in the project/deployment-artifacts directory.
- clean
Purges all temporary, working and artifacts directories.
- deploy
Distribute and deploy all binaries hosted in the project/deployment-artifacts directory.
- init

Display information message about possible execution targets and stops.

- javadoc
Builds a Javadoc ZIP file for the project.
- package-download
Builds an Open Source download package in ZIP and GZ formats for public posting on GForge.

All targets must be self documenting and self validating. If a dependant property is not present, the build must fail with an informative message.

Build & Deployment Properties

Per the [Auto Deploy Project Installation](#), there are properties which control the description of the build and deployment environment. These must be used as needed. Those not needed for a specific project can be ignored. If there are additional environment (non-project specific) properties needed, send a request through the NCICB Help Desk with appropriate details.