



Author: Jon Barnett
Date: 11th November 2003

Jetty 4.2.14 for JBoss

This technote is a quick guide to building Jetty 4.2.14 for JBoss. The JBoss version this build was tested on was 3.2.2. The reason for the guide is because of some issues with the build.

Preparing for the build

You will require Ant 1.5.1 or higher, XDoclet 1.2b2 or higher and a Java Development Kit (JDK) to build Jetty. It is preferable to use a JDK version at 1.4.0 or higher. You will also need an installed copy of JBoss 3.2.2.

In order to build from the source you will need to obtain the Jetty 4.2.14 source and the 4.2.14 extra packages from the Jetty SourceForge site, <http://sourceforge.net/projects/jetty/>. Unpack these into the same directory, unpacking the Jetty-4.2.14-src.tar.gz package first. We will call this root directory JETTY_SRC from now on.

The package I downloaded had two missing files in JETTY_SRC. I created two empty files in my Linux environment to get around the build problem.

```
cd JETTY_SRC
touch LICENSE.html
touch VERSION.TXT
```

Should you want an optimized Jetty build, you will need to modify the ant.properties. You will also have to add some properties so that the build will create the components for a JBoss deployment. The following shows the changes and additions to JETTY_SRC/ant.properties.

```
javac.debug=off
javac.optimize=on
javac.deprecation=off

jmx.available=true
jboss.available=true
```

The first three lines are modifications to existing definitions, and are configured for code optimization. The last two lines are additions to ensure that the JBoss components are built.

You also need to create a directory ext in JETTY_SRC.

```
mkdir ext
```

Obtain copies of jasper-compiler.jar, jasper-runtime.jar and jboss-jmx.jar and place them in JETTY_SRC/ext. You should be able to obtain copies of the Jasper libraries from your JBoss 3.2.2 install, most likely in the directory JBOSS_HOME/server/default/deploy/jbossweb-tomcat41.sar. You can obtain the jboss-jmx.jar file from JBOSS_HOME/lib.

IBM JSSE (HTTPS for IBM SDK 1.4.x)

This module provides a HTTPS listener for Jetty when running Jetty within an IBM Java 2 1.4.x environment. You will need an IBM 1.4.x SDK to build this. Only make these changes if you intend to build this IBM SDK module.

The package exists in JETTY_SRC/extra/ibmjsse. There are a few changes required to the build.xml for the module as the project name and package name are defined as the load balancer by mistake. The changes are shown below for JETTY_SRC/extra/ibmjsse/build.xml.

```
<project name="IBMJsse" default="jars" basedir=". ">
  <property name="package"          value="ibmjsse" />
  <property name="ibmjsse.available" value="true" />
```

You will also need to add the security.jar from the IBM SDK in order to build the module. An example is given below.

```
<path id="classpath">
  <pathelement location="/usr/local/IBMJava2-141/jre/lib/security.jar" />
  <pathelement location="{jetty.jar}" />
  <pathelement location="{servlet.jar}" />
  <path refid="extpath" />
</path>
```

Note the change to org.mortbay.jetty.http.ibmjsse.IbmJsseListener where previously it was org.mortbay.jetty.http.IbmJsseListener.

J2EE (JBoss SAR components)

This module provides the JBoss components to allow Jetty to operate as a web application container plug-in for JBoss. The package exists in JETTY_SRC/extra/j2ee. There are a few changes required to jboss.properties to allow the module to be built. The location of JBoss must be given as well as the location of XDoclet. XDoclet is used to generate the MBean support code and MBean configuration files for JBoss. The changes for my environment are shown below for JETTY_SRC/extra/j2ee/jboss.properties.

```
jboss.version=3.2.2
jboss.home=/usr/local/JBoss-3.2.2
#jboss.version=3.2.1
#jboss.home=/usr/local/java/jboss/jboss-3.2.1
xdoclet.home=/usr/local/xdoclet-1.2b4
```

Modify xdoclet.home and jboss.home according to your own requirements.

Building Jetty 4.2.14

Return to JETTY_SRC and start the build.

```
cd JETTY_SRC
ant
ant extra
```

When this completes successfully, you will have the base Jetty libraries, the Jboss-Jetty SAR and the extra modules.

Relocating the libraries

The base libraries **javax.servlet.jar**, **org.mortbay.jetty.jar** and **org.mortbay.jmx.jar** are created in JETTY_SRC/lib. We will ignore this as the libraries are copied into the SAR distribution.

The extra libraries are contained in JETTY_SRC/extra/lib. These include the IBM JSSE listener implementation. Should you have followed the build.xml changes suggested earlier, the filename for this listener will be **org.mortbay.ibmjsse.jar**.

The JBoss plug-in is created in JETTY_SRC/extra/j2ee/lib. For our example, it will have created the SAR as **jboss-3.2.2-jetty-4.2.14.sar**, taking the version number of JBoss set in jboss.properties and the version of the Jetty source being built. An easier way to deploy your new Jetty plug-in for JBoss is to change to the JETTY_SRC/extra/j2ee directory and run the deploy task.

```
cd JETTY_SRC/extra/j2ee
ant deploy
```

This will build and create an unpacked SAR in JBOSS_HOME/server/default/deploy. The SAR will be called **jboss-jetty.sar**. You may wish to rename it **jbossweb-jetty.sar** or some other variant.

You can copy any extra libraries from JETTY_SRC/extra/lib into the JBOSS_HOME/server/default/deploy/jboss-jetty.sar directory. You probably don't want to copy the **org.mortbay.jetty-jdk1.2.jar** as that is intended for JDK 1.2 deployments.

You should remove the **javax.servlet.jar** from **JBOSS_HOME/server/default/deploy/jboss-jetty.sar** as this file exists in JBOSS_HOME/server/default/lib. You should however replace the file in JBOSS_HOME/server/default/lib with the Jetty copy if you have built Jetty with optimizations. The **javax.servlet.jar** bundled with JBoss is not optimized.

You should also remove the other web application server. Most likely this will be Tomcat so for JBoss 3.2.2, you would delete the JBOSS_HOME/server/default/deploy/jbossweb-tomcat41.sar directory and everything below it. For other release bundles, you may need to identify the relevant SAR deployment in order to delete it. In JBoss 3.2.0 the Tomcat SAR was jbossweb-tomcat.sar and the Jetty SAR was jbossweb-jetty.sar.

Since this deployment only makes changes to the default run-time for JBoss, you will need to appropriately manually modify other run-time deployments for JBoss, based on these instructions.

That completes the work required to upgrade your JBoss bundle to Jetty 4.2.14. You will need to edit the jboss-service.xml for Jetty for your requirements as for any Jetty or Tomcat deployment.