

Jetspeed2 Fusion

Fusion

Fusion runs JSR-168 Portlets inside of Jetspeed 1.6. To build Fusion, you need to first install and build Jetspeed-2. (building J2 will not be necessary with the final release and distribution of 1.6) Then set this property in your `${HOME}/build.properties`:

```
org.apache.jetspeed.fusion=true
```

and then rebuild Jetspeed 1.6

```
maven -Dmaven.test.skip=true clean war
maven -o deploy
goto: http://localhost:8080/jetspeed/portal
```

To install portlet applications, simply drop them into the `${TOMCAT_HOME}/webapps/jetspeed/portal/WEB-INF/deploy` directory. Portlets from these applications will automatically show up in the "Add Portlet" browser in the Jetspeed 1.6 customizer.

Installing Fusion on JBoss

Since moving the commons jars out of JBoss's `server/default/lib`, I had to take a different approach. IF the above procedure fails, try this one:

This has only been tested on JBoss 3.2.5 w/Tomcat so your mileage may vary.

1. Perform the standard *maven allBuild*
2. Expand the `/portal/target/jetspeed.war` into `jboss/server/default/deploy/jetspeed.war` (directory)
3. For the demo portlets to show up you will need to copy the following 6 wars into `jboss/server/default/deploy/jetspeed.war/WEB-INF/deploy`
 - `/layout-portlets/target/jetspeed-layouts.war`
 - `/applications/struts-demo/target/struts-demo.war`
 - `/applications/php/target/php.war`
 - `/applications/perl/target/perl.war`
 - `/applications/security/target/security.war`
 - `/applications/demo/target/demo.war`
4. Copy these 5 jars into `jboss/server/default/lib`
 - `pluto-1.0.1-SNAPSHOT.jar`
 - `portals-bridges-common-0.1.jar`
 - `jetspeed-commons-2.0-a1-dev.jar`
 - `jetspeed-api-2.0-a1-dev.jar`
 - `portlet-api-1.0.jar`
5. Move these 2 jars out of `jetspeed.war/WEB-INF/lib` into `jboss/server/default/lib`
 - `commons-logging-1.0.3.jar`
 - `log4j-1.2.8.jar`
6. copy the JBoss datasource definition of your choice from `/portal/src/resources` to `jboss/server/default/deploy/`
 - For the default Hsqldb, use `jetspeed-ds.xml`
 - For MySQL, use `jetspeed-mysql-ds.xml`
7. replace JBoss' `hsqldb.jar` with Jetspeed's, or drop in MySQL driver into `jboss/server/default/lib`
8. In `jetspeed-spring.xml`, locate the bean with the `id="org.apache.jetspeed.tools.pamanager.servletcontainer.ApplicationServer*_Manager"`. Comment out the current bean definition, which is targeted at Tomcat 4/5, and uncomment the bean definition below that one, which is targeted at JBoss. Remove all arguments.
9. don't forget the MySQL driver if using it. Copy it into `jboss/server/default/lib`
10. replace `dom4j.jar` under the root `jboss/lib` directory with this one <http://www.bluesunrise.com/jboss/dom4j.jar>

Installing Fusion on Weblogic

This has only been tested on Weblogic 8.1 In these instructions, I will refer to the BEA home directory as `${BEA}`
You will need to create a Weblogic domain. In these examples, we will call this domain "yourdomain"

1. Build the standard Jetspeed Fusion with Tomcat (see above) 2. create a shared/lib directory under bea home directory `${BEA}` 3. Copy these 5 jars into `jboss/server/default/lib` (from Tomcat's `shared/lib`)
 - `pluto-1.0.1-SNAPSHOT.jar`
 - `portals-bridges-common-0.1.jar`
 - `jetspeed-commons-2.0-a1-dev.jar`
 - `jetspeed-api-2.0-a1-dev.jar`
 - `portlet-api-1.0.jar`
4. From the Jetspeed build, copy the `/portal/target/jetspeed/` directory into Weblogic:
`${BEA}/user_projects/domains/yourdomain/jetspeed/` (directory)
5. Edit the `startWeblogic.cmd` found under `${BEA}/user_projects/domains/yourdomain/`

Add the following lines (NOTE: `${BEA}` needs to be replaced with your root BEA installation directory):

```
set SHLIB=${BEA}/shared/lib
set SHARED_PATH=%SHLIB%\jetspeed-api-2.0-a1-dev.jar;%SHLIB%\jetspeed-commons-2.0-a1-dev.jar;%SHLIB%\portals-bridges-common-0.1.jar;%SHLIB%\portlet-api-1.0.jar;%SHLIB%\pluto-1.0.1-SNAPSHOT.jar;
set CLASSPATH=%SHARED_PATH%;%XMLPATH%;%WEBLOGIC_CLASSPATH%;%POINTBASE_CLASSPATH%;%JAVA_HOME%\jre\lib\rt.jar;%WL_HOME%\server\lib\webservices.jar
```

6. In jetspeed-spring.xml, locate the bean with the id="org.apache.jetspeed.tools.pamanager.servletcontainer.ApplicationServer_**Manager". Replace "Tomcat" with "Weblogic". Remove all arguments.

1. Don't forget the JDBC driver. Copy it into \${BEA}/shared/lib
2. Setup a JDBC Connection and Pool. Startup weblogic running the startWeblogic.cmd
Goto: <http://localhost:7001/console/> and login, goto Services->JDBC->Connection Pools. Create a new pool. Create a JDBC data source named JetspeedDB based on the Connection Pool. 7. Install Portlet Applications by dropping them into the \${BEA}/user_projects/domains/yourdomain/applications/ directory.
I've had pretty good luck dropping them in expanded, or using Jetspeed's Auto Deployer.
However I find that I need to prod it along by using the Admin console to deploy it once more:
<http://localhost:7001/console/>
And then go to Deployments -> Web Application Modules -> Deploy a new Web Application Module
Select the applications directory, you should see your portlet app there. Click the radio button next to the directory name and press Target Module and then Deploy.
NOTE: remember that you will always need a weblogic.xml file in your WEB-INF directory, even if its empty, it is absolutely required.
8. Ensure that Jetspeed is running: go to <http://localhost:7001/jetspeed/portal>
9. If you did not use the Auto Deployer, you will need to register the portlet application with Fusion using the Fusion JSR-168 Registration program (from the Admin/Security tab under Admin account)

Installing Fusion on Websphere

This procedure has been tested on Websphere Application Server Version 5 (the trial download).

1. Patch Websphere's Appserver/java/jre/lib/core.jar with 1.4.2's classes:
 - Preferences.class
 - Preferences\$.class

This requires expanding the 1.4.2 rt.jar, and IBM's core.jar Copy over the 2 files listed above into the expand core.jar and recreate the jar and drop it back into Appserver/java/jre/lib/

2. Copy Jetspeed's xalan*.jar into Websphere/Appserver/lib

3. Copy these 5 jars into Websphere/Appserver/lib

- pluto-1.0.1-SNAPSHOT.jar
- portals-bridges-common-0.1.jar
- jetspeed-commons-2.0-a1-dev.jar
- jetspeed-api-2.0-a1-dev.jar
- portlet-api-1.0.jar

4. Start Websphere

5. Create a MySQL (or other) JDBC Provider.

NOTE that Websphere does not support Hypersonic SQL

This procedure is pretty complicated. See instructions here for help:

<http://www.webspherepower.com/issuesprint/issue200403/00001236.html>

Create a Data Source named JetspeedDS. The JNDI name should be the same (JetspeedDS).

Make sure to use com.ibm.websphere.rsadapter.ConnectJDBCDataStoreHelper as the DataSource helper. Follow the guide in the link above for custom properties. The custom properties define your connection.

6. Install the Jetspeed War (jetspeed.war) file using the Websphere's application installation wizard.

Applications->Install New Application

Remember to associate Jetspeed with the data source defined above on installation.

7. Go to <http://localhost:9080/jetspeed/portal>

You should be ready to go

8. Install Portlet Applications using the Websphere application installation wizard.

9. Register the portlet application with Fusion using the Fusion JSR-168 Registration program (from the Admin/Security tab under Admin account)

NOTE: where as most app servers require the context to start with /, Websphere seems to require that it does not. So if your web app context name is "/mycontext", enter "mycontext"