The page contains the build procedure for the OpenJBI components located under DSF-DI\vendor\openjbi in the ESV-M SVN repository.

# Pre-reqs

## Software

* Java 1.5.0\_22
* Maven 2.2.1

## Environment Variables

The following environment variables need to be set prior to attempting the build.

* JV\_SRCROOT - The directory where the full OpenJBI component source tree is located; this source tree is located in the zip file under DSF-DI\vendor\openjbi\drop
* JV\_GFBASE - The directory where Glassfish ESB v2.2 is installed. The installer for this application is located under DSF-DI\vendor\glassfish\bin

Examples:

{code}

set JV\_SRCROOT=c:\bit9prog\dev\openjbi-legacy

set JV\_GFBASE=c:\bit9prog\dev\GlassFishESBv22\glassfish

{code}

## Maven

The %USERPROFILE%\.m2\settings.xml file needs to be modified to include additional repositories. Example settings.xml file containing the additional repositories:

{code}

<?xml version="1.0" encoding="UTF-8"?>

<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0 http://maven.apache.org/xsd/settings-1.0.0.xsd">

<mirrors>

<mirror>

<id>servicemix-repo</id>

<name>Apache ServiceMix Repository</name>

<url>http://svn.apache.org/repos/asf/servicemix/m2-repo/</url>

<mirrorOf>servicemix-repo,servicemix-m2-repo</mirrorOf>

</mirror>

<mirror>

<id>ibiblio.org</id>

<name>ibiblio Mirror of http://repo1.maven.org/maven2/</name>

<url>http://mirrors.ibiblio.org/pub/mirrors/maven2</url>

<mirrorOf>central</mirrorOf>

</mirror>

</mirrors>

<profiles>

<profile>

<id>private-profile</id>

<activation>

<activeByDefault>true</activeByDefault>

</activation>

<repositories>

<repository>

<id>java-net-legacy-repository</id>

<name>java.net Maven 1.x repository</name>

<url>https://maven-repository.dev.java.net/nonav/repository</url>

<layout>legacy</layout>

<releases>

<enabled>true</enabled>

</releases>

<snapshots>

<enabled>true</enabled>

</snapshots>

</repository>

</repositories>

</profile>

<profile>

<id>extraPublicRepositories</id>

<repositories>

<repository>

<id>jboss.org</id>

<name>JBoss repository</name>

<url>http://repository.jboss.org/maven2/</url>

</repository>

<repository>

<id>java.net-maven2-repo</id>

<name>Java.net Repository for Maven 2</name>

<url>http://download.java.net/maven/2/</url>

<snapshots>

<enabled>true</enabled>

<updatePolicy>never</updatePolicy>

</snapshots>

</repository>

<repository>

<id>codehaus.org-maven2-snapshots</id>

<name>Codehaus Snapshot Development Repository</name>

<url>http://snapshots.repository.codehaus.org</url>

<snapshots>

<enabled>true</enabled>

<updatePolicy>never</updatePolicy>

</snapshots>

</repository>

<repository>

<id>osgeo-geotools</id>

<name>OSGeo's GeoTools repository</name>

<url>http://download.osgeo.org/webdav/geotools/</url>

<snapshots>

<enabled>true</enabled>

<updatePolicy>never</updatePolicy>

</snapshots>

</repository>

</repositories>

<pluginRepositories>

<pluginRepository>

<id>codehaus.org-maven2-plugin-repo</id>

<name>Codehaus Maven2 Plugin Repository</name>

<url>https://nexus.codehaus.org/content/repositories/releases/</url>

</pluginRepository>

<pluginRepository>

<id>codehaus.org-maven2-plugin-snapshots</id>

<name>Codehaus Snapshot Development Repository</name>

<url>https://nexus.codehaus.org/content/repositories/snapshots/</url>

<snapshots>

<enabled>true</enabled>

</snapshots>

</pluginRepository>

<pluginRepository>

<id>codehaus.org-maven2-plugin-snapshots-alt</id>

<name>Codehaus Snapshot Development Repository</name>

<url>https://nexus.codehaus.org/content/groups/snapshots-group/org/codehaus/mojo/</url>

<snapshots>

<enabled>true</enabled>

</snapshots>

</pluginRepository>

<pluginRepository>

<id>java.net-maven2-plugins-repo</id>

<name>Java.net Repository for Maven</name>

<url>http://download.java.net/maven/2/</url>

</pluginRepository>

</pluginRepositories>

</profile>

</profiles>

<activeProfiles>

<activeProfile>extraPublicRepositories</activeProfile>

</activeProfiles>

</settings>

{code}

The Oracle JDBC driver then needs to be installed (adjust file path as needed):

{code}

mvn -Dmaven.repo.local=%JV\_SRCROOT%/m2/repository install:install-file -DgroupId=oracle.jdbc \

-DartifactId=ojdbc14 -Dversion=10.1.0.5 -Dpackaging=jar \

-Dfile=c:/bit9prog/dev/DSF-DI/DSF-DI\_trunk/vendor/openjbi/lib/ojdbc14.jar

{code}

The Saxon jar then needs to be installed (adjust file path as needed):

{code}

mvn -Dmaven.repo.local=%JV\_SRCROOT%/m2/repository install:install-file -DgroupId=net.sf.saxon \

-DartifactId=saxon -Dversion=9.1.0.3 -Dpackaging=jar \

-Dfile=c:/bit9prog/dev/DSF-DI/DSF-DI\_trunk/vendor/openjbi/lib/saxon-9.1.0.3.jar

{code}

The SBME jar then needs to be installed (adjust file path as needed):

{code}

mvn -Dmaven.repo.local=%JV\_SRCROOT%/m2/repository install:install-file -DgroupId=com.sun \

-DartifactId=sbme -Dversion=1.0 -Dpackaging=jar \

-Dfile=c:/bit9prog/dev/DSF-DI/DSF-DI\_trunk/vendor/openjbi/lib/sbme-1.0.jar

{code}

The HSQLDB jar then needs to be installed (adjust file path as needed):

{code}

mvn -Dmaven.repo.local=%JV\_SRCROOT%/m2/repository install:install-file -DgroupId=hsqldb \

-DartifactId=hsqldb -Dversion=1.8.0.2 -Dpackaging=jar \

-Dfile=c:/bit9prog/dev/DSF-DI/DSF-DI\_trunk/vendor/openjbi/lib/hsqldb-1.8.0.2.jar

{code}

The Glassfish Extras jars then need to be installed (adjust file paths as needed):

{code}

mvn -Dmaven.repo.local=%JV\_SRCROOT%/m2/repository install:install-file -DgroupId=org.glassfish.extras \

-DartifactId=appserv-rt -Dversion=2.1.1 -Dpackaging=jar \

-Dfile=c:/bit9prog/dev/DSF-DI/DSF-DI\_trunk/vendor/openjbi/lib/appserv-rt.jar

mvn -Dmaven.repo.local=%JV\_SRCROOT%/m2/repository install:install-file -DgroupId=org.glassfish.extras \

-DartifactId=appserv-admin -Dversion=2.1.1 -Dpackaging=jar \

-Dfile=c:/bit9prog/dev/DSF-DI/DSF-DI\_trunk/vendor/openjbi/lib/appserv-admin.jar

mvn -Dmaven.repo.local=%JV\_SRCROOT%/m2/repository install:install-file -DgroupId=org.glassfish.extras \

-DartifactId=appserv-deployment-client -Dversion=2.1.1 -Dpackaging=jar \

-Dfile=c:/bit9prog/dev/DSF-DI/DSF-DI\_trunk/vendor/openjbi/lib/appserv-deployment-client.jar

mvn -Dmaven.repo.local=%JV\_SRCROOT%/m2/repository install:install-file -DgroupId=org.glassfish.extras \

-DartifactId=appserv-ext -Dversion=2.1.1 -Dpackaging=jar \

-Dfile=c:/bit9prog/dev/DSF-DI/DSF-DI\_trunk/vendor/openjbi/lib/appserv-ext.jar

mvn -Dmaven.repo.local=%JV\_SRCROOT%/m2/repository install:install-file -DgroupId=org.glassfish.extras \

-DartifactId=appserv-se -Dversion=2.1.1 -Dpackaging=jar \

-Dfile=c:/bit9prog/dev/DSF-DI/DSF-DI\_trunk/vendor/openjbi/lib/appserv-se.jar

mvn -Dmaven.repo.local=%JV\_SRCROOT%/m2/repository install:install-file -DgroupId=org.glassfish.extras \

-DartifactId=webservices-rt -Dversion=2.1.1 -Dpackaging=jar \

-Dfile=c:/bit9prog/dev/DSF-DI/DSF-DI\_trunk/vendor/openjbi/lib/webservices-rt.jar

{code}

# Build Procedures

Before the the OpenJBI components that have been modified for DSF-DI can be built, the unmodified version of the OpenJBI components must first be built to populate the maven repository with the applicable dependencies. This only needs to be done once per dev machine / JV\_SRCROOT location.

Run the following command from the JV\_SRCROOT directory:

{code}

mvn -Dmaven.test.skip=true -Dmaven.repo.local=%JV\_SRCROOT%\m2\repository

{code}

After the initial build is complete, run the following command from DSF-DI\vendor\openjbi\src :

{code}

mvn -Dmaven.repo.local=%JV\_SRCROOT%\m2\repository

{code}