

Pre-requisites

1. Apache Active MQ. Required for examples 3.2.7 and chapter 10's Apache Synapse examples. Download at: <http://activemq.apache.org/activemq-520-release.html>.
2. Apache Tomcat (test w/6.0.18). Required for examples 4.4.1. Optionally, the JBoss 4.2.3 app server installed for Drools and jBPM may be used.
3. Java JDK 1.5. Issues will be encountered when using 1.6 with the Apache Synapse examples in chapters 9 & 10.

I. Steps for Setting up the Code Samples

Eclipse Ganymede w/SOA Developer Plugins Setup

1. Download Eclipse Ganymede, version "Eclipse IDE for Java Developers" from <http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/ganymede/SR2/eclipse-java-ganymede-SR2-win32.zip>. Unzip to desired location and run the executable to launch. We recommend creating a separate workspace just for the code sample project.
2. Install SOA Development plugins. When running Eclipse,
 - a. Select *Help->Available Software*.
 - b. Select checkboxes *Ganymede Update Site -> SOA Development*. All items should be selected. Confirm when prompted to install all of the associated plugins.

Import Source Code

1. Download the source zip file <http://jdavis.open-soa.info/wordpress/>.
2. There are two options available, "Source with all libraries" and "Source with no libraries". The distinction between the two is that the "Source with all libraries" contains all JAR files necessary. The "Source with no libraries" option will require you to run the ant target copy.libs which will fetch all of the required libraries from various Maven repositories.

NOTE: We recommend the all libraries option. Although it's a bigger download initially, the no libs option can occasionally encounter some issues if some of the Maven repositories is unavailable.

3. In Eclipse, select *File->Import*. In the Wizard pane, open *General->Existing Projects into Workspace*. Select *Next*, then choose the *Select archive file* option, then browse and load the OpenSOA.zip file you downloaded in the previous step. Click *Finish* (when prompted for whether to overwrite certain files, select *Yes to All*).

4. Update the settings found in the file *resources/build.properties*. Set the `project.home` property so that the directory location points to location where the OpenSOA-Book project resides. This file is used by the various ant scripts that correspond to each chapter.

II. Steps for Setting up JBoss jBPM (Chapters 5-7)

While JBoss jBPM can be run in a standalone, embedded fashion, for the examples in chapters 6 & 7, it's advisable to have jBPM Console available, since that is a fairly important part of the overall solution (ant scripts are available to run the examples in embedded fashion, so it's not completely required, just suggested). The jBPM Console can be run under a variety of app servers, but the only "officially" supported one is JBoss. To make things as straightforward as possible, we recommend that you actually download and install the JBoss Drools release "Drools 5.0.0.CR1 Guvnor Standalone". This is available at: <http://www.jboss.org/drools/downloads.html>. This includes JBoss 4.2.3, which is also supported by JBoss jBPM. So, with that in mind, do:

Install JBoss 4.2.3 App Server Along w/JBoss Drools

1. Download Drools with JBoss 4.2.3 at:
<http://download.jboss.org/drools/release/5.0.0.25561.CR1/drools-5.0.0.CR1-guvnor-standalone.zip>.
2. Unzip to an appropriate destination directory on your box. A subdirectory called "jboss-4.2.3.GA" is where JBoss app server is installed, and we'll call this `JBOSS_HOME`.

Install JBoss jBPM

1. Download JBoss jBPM at:
http://sourceforge.net/project/showfiles.php?group_id=70542&package_id=145174&release_id=667500
2. Change directory to the download location, and in a command window, type "`java -jar jbpms-installer-3.2.6.SP1.jar`". When prompted:
 - a. Enter directory location where you want jBPM installed. We will refer to this as `JBPM_HOME`.
 - b. Select the default options for which items to install ("jBPM3 JBoss Integration").
 - c. When prompted for target server, select "JBoss-4.2.3". Leave the server name as "default".
 - d. When prompted for a database, select "Hypersonic".
 - e. When prompted for the location of the JBoss Home, specify the location where you installed JBoss in step 2 above. However, note that the JBoss app server is installed as a subdirectory within Drools 5.0 standalone-guvnor directory, so that is what you want to specify.
 - f. Start the JBoss apps server by running (`JBOSS_HOME/bin/run.bat`). Once running, you can access the jBPM Console at: <http://localhost:8080/jbpm-console>. You can specify

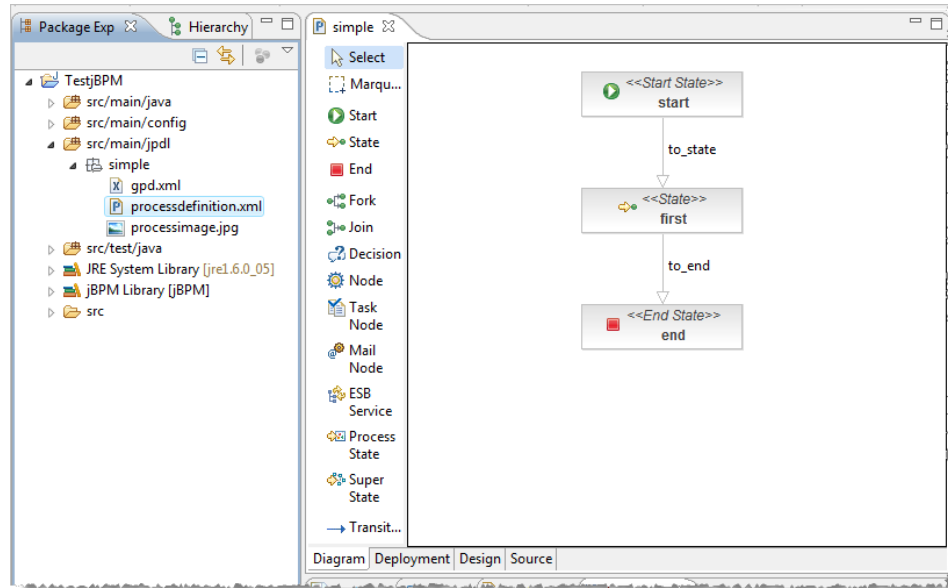
any of the credentials listed on the main splash page that appears (such as manager/manager).

JBoss jBPM Eclipse Setup (jBPM Designer)

1. The jBPM plugin requires the jBPM runtime engine. This is used for creating the visual process flows.
2. When running Eclipse:
 - a. Select *Help->Software Updates*.
 - b. Select the "Add Site..." button.
 - c. In the pop-up that appears, select "Archive...". Then, navigate to the JBPM_HOME/designer and select the file: jbpdm-jpdl-designer-site.zip. (see step 2 in the section above to identify the location of your JBPM_HOME).
 - d. When the pop-up disappears, "jBPM jPDL GPD" will appear. Click on checkbox adjacent to that, and then click on the "Install..." button. This will install the plugin.
 - e. Click "Next" when the install confirmation page appears. Agree to the license terms and click "Finish". Restart Eclipse, when prompted.
3. With Eclipse running, verify that the jBPM plugin installed correctly by:
 - a. Select *Window -> jBoss jBPM*. Expand the submenu, and select "Runtime Locations". Click on the "Add..." button. In the "Name:" field enter "jbpm", and in the "Location:" field, click on the "Search..." button and enter the location of your JBPM_HOME directory. Click "OK" to save the changes.
 - b. Select *File->New->Other*, then scroll through Wizard list and *JBoss jBPM* should appear as a folder. Expand and select *Process Project*. Click *Next*.
 - c. When prompted for a project name, give it something like "TestjBPM".

NOTE: It appears as though the 3.2.6.SP1 release is missing the *activation.jar* file. This can be copied from the OpenSOA project's lib directory (rename *activation-1.1.jar* to *activation.jar*). You will need to refresh the test jBPM project for the change to appear.

- d. This will result in a new project being created. Now, if you expand the project and open the *src/main/jpdl* folder and click on the *processdefinition.xml*, you should see something resembling:



This means that jBPM is properly installed. You can now right click on the project name and remove, if you so desire.

Specific instructions on how to run the examples can then be found in the jBPM chapter source subdirectories (look for the README.txt files).

III. Steps for Setting up JBoss Drools (Chapters 11-12)

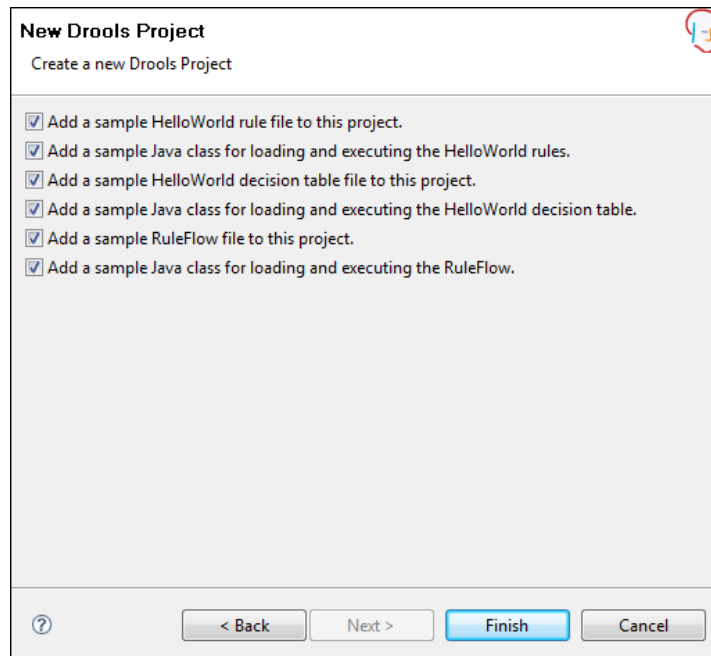
JBoss Drools Eclipse Binaries

1. Download Drools 5.0 Binaries at: <http://www.jboss.org/drools/downloads.html> (link should be titled something like "Drools 5.0 Binaries"). Unzip to a directory we will call DROOLS_HOME.

JBoss Drools Eclipse (IDE) Setup

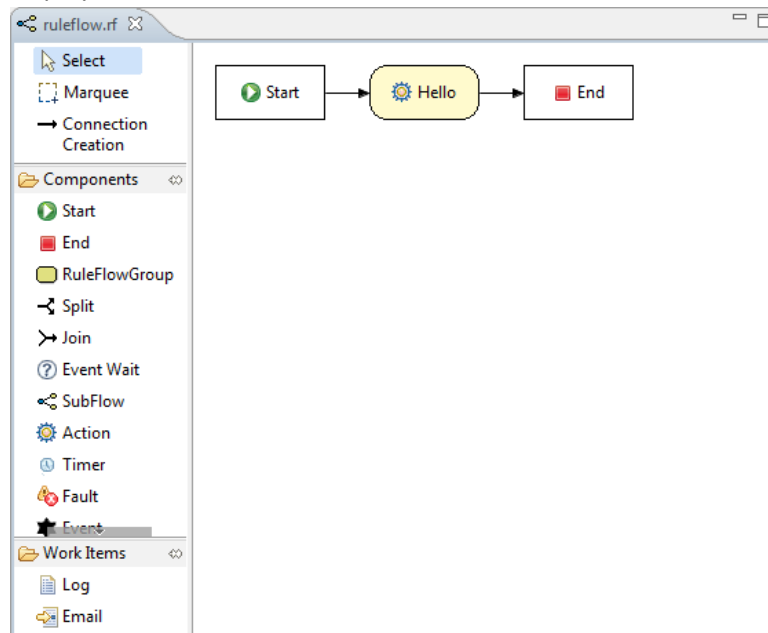
1. Download the Drools 5.0 Eclipse IDE plugin at: <http://www.jboss.org/drools/downloads.html> (download is titled Drools 5.0 IDE) . Once downloaded, then:
 - a. Exit Eclipse, if currently running.
 - b. Unzip contents to a temporary directory.
 - c. Copy contents of directory (which includes the *features* and *plugins* directories) to your Eclipse home. When prompted, selected Yes to overwrite the existing directories.
 - d. Launch Eclipse. Select *Window->Drools*. Expand the submenu and the choose "Installed Drools Runtime". Click "Add..", then specify the location of your DROOLS_HOME which was setup in the previous section. Click "OK" to save your changes.
 - e. Select *File->Project*. When the available Wizards window appears, expand the Drools folder and select *Drools Project*. Then, click Next and assign a project name such as

"TestDrools" (the purpose of the project is to just confirm the Drools plugin is setup correctly). Click *Next* (not *Finish*), then select all of the checkboxes that appear, such as:



Select *Finish* to complete the project setup.

- f. Now, verify the Drools plugin is actually working properly by expanding the tree node *src/main/rules* (this assumes you are in the Java Eclipse perspective, which can be accessed by choosing *Window->Open Perspective* and selecting *Java Browser*). Then, double click on *ruleflow.rf*. This should result in the following editor pane being displayed:



If the figure above is shown, the Eclipse Drools plugin has been successfully installed.cd

Installing Drools Guvnor

Drools Guvnor is covered in Chapter 12. Installation of Guvnor was actually performed in section II - "Steps for Setting up JBoss jBPM". As you recall, we setup Guvnor in that step since it includes JBoss 4.2.3, which was used for the jBPM examples. After starting the JBoss app server (JBOSS_HOME/run/run.bat), you can access Guvnor at: <http://localhost:8080/drools-guvnor>. When prompted for a username/password, you can use admin/admin, but actually anything will work, as, by default, no backend credentialing is setup (JAAS is used to configure security).