

# EASIE JBoss Plugin Installation

## JBoss Server Development With Eclipse

Genuitec, L.L.C.  
March, 2002

This article introduces practical J2EE server-side development techniques using Eclipse with JBoss Server 2.4.4 integrated with Tomcat 4.0.1 (JBoss). We'll demonstrate how to install, configure, and run the EASIE JBoss Plugin for Eclipse in preparation to performing J2EE server-side development. Additionally, use of this plugin with JBoss 3.0 Alpha and Beta will also be discussed.

### Introduction

Genuitec recently released the EASIE JBoss Plugin. It is the second in a series of application and web server plugins for the Eclipse 2.0 IDE. This product extends the Eclipse IDE to include user controls for managing a JBoss instance from within Eclipse runtime and debug contexts. The plugin is built on top of Genuitec's Eclipse Application Server Integration Environment, (EASIE™), a framework for interfacing Eclipse with Java/J2EE application servers. In this article, we demonstrate how to install, configure, and run the EASIE JBoss Plugin for Eclipse.

### Getting Started – The Software Environment

The following software environment was used as the basis for this article. For similar results, it is highly recommended that you use the same or later versions of all listed software. When defining directory paths remember to compensate for differences between the environment presented in this article and your development environment.

Software	Version
Java JDK	Sun JDK 1.4 rc 0 or greater
JBoss Server	Version 2.4.4 with Tomcat Integration Version 3.0 Alpha/Beta with Tomcat Integration
Eclipse IDE	Version 2.0 (pre-release) Build 20020125

### EASIE JBoss Plugin Installation

Download the latest EASIE JBoss Plugin from [Genuitec](http://www.genuitec.com). The plugin is distributed in ZIP format, therefore, using your ZIP file utility, unzip the `EASIEJBoss.zip` file into the `<Eclipse_Home>` directory. This will create a new plugin directory named `com.genuitec.eclipse.easie.jboss`.

### Step-1, Enable the EASIE JBoss Plugin

Once the plugin has been installed it must be activated. From the main menu navigate to '**Perspective > Customize...**', expand the '**Other**' node on Available Action Sets and select '**EASIE JBoss**'.

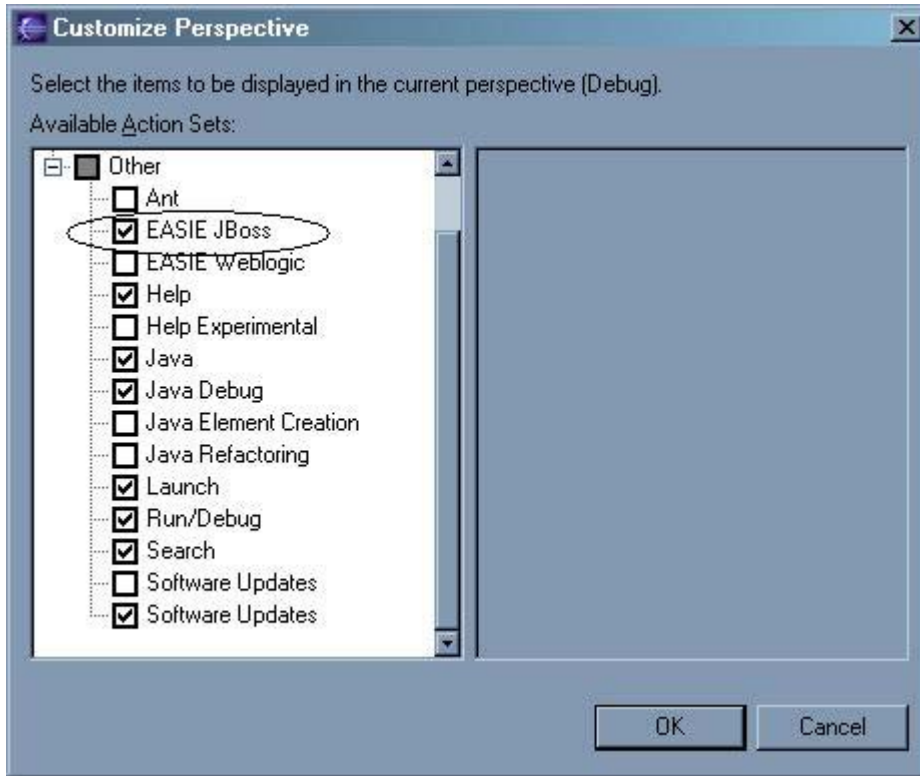


Figure 1

Select '**OK**' to activate the plugin. When activated, the plugin installs a **JBoss** menu to the Eclipse menubar and UML node-based start and stop icons to the Eclipse toolbar.



Figure 2

## Step-2, JBoss Plugin Configuration

With the JBoss Plugin activated, it must be configured before attempting to start JBoss. Select the **Workbench > Preferences** and choose the **JBoss** node to view and edit the plugin properties.

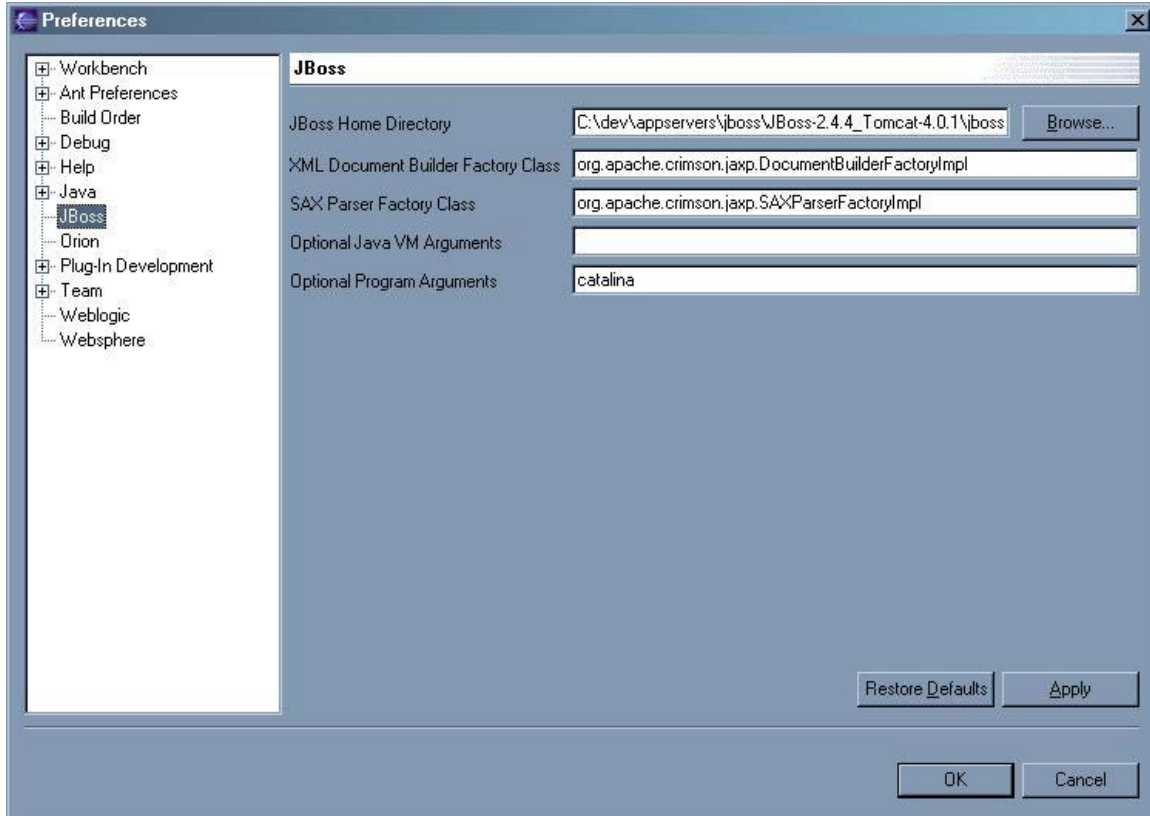



Figure 3

## JBoss Plugin Property Summary

Property	Default Value	Required	Description
JBoss Home Directory		Yes	Path to JBoss home directory. This is the directory named jboss directly containing the bin directory.
XML Document Builder Factory Class	org.apache.crimson.jaxp.DocumentBuilderFactoryImpl	Yes	Path to XML builder factory class
SAX Parser Factory Class	org.apache.crimson.jaxp.SAXParserFactoryImpl	Yes	The name of the SAX parser factory class
JVM Arguments		No	Additional commandline arguments to be passed to the JVM when launching JBoss.
Optional Program Arguments		No	Program arguments passed to JBoss. For version 2.4.4 this is often the configuration name as shown. For version 3.0, the configuration name can be set by using -c <configurationname>.

Please consult the resources at the end of this article for a comprehensive description of each configuration property.

## Starting JBoss

Once the JBoss Plugin has been configured, you should be able to start it from either the **JBoss > Start JBoss** menu or the green UML node icon  on the toolbar. Selecting either start option causes the JBoss Plugin to launch a JBoss Server instance within an Eclipse debug perspective. The plugin echoes the stdout and stderr of the JBoss instance to the Eclipse console. This allows you to monitor the JBoss startup and shutdown processes as well as informational and error messages output by the JBoss instance.

When JBoss starts correctly, you should observe console output that reports the startup status of all the internal services. You will also notice a number of new threads have been created as children under for the `org.jboss.Main` program. `org.jboss.Main` is the Java startup class for running JBoss.

Pay close attention to the console output if JBoss fails to startup. The most common reason for startup failure is the port is already bound to another process. If this is the case, either reconfigure your JBoss configuration to launch JBoss on a different port or terminate the process using the specified port, which is most likely a forgotten instance of a prior execution of JBoss.

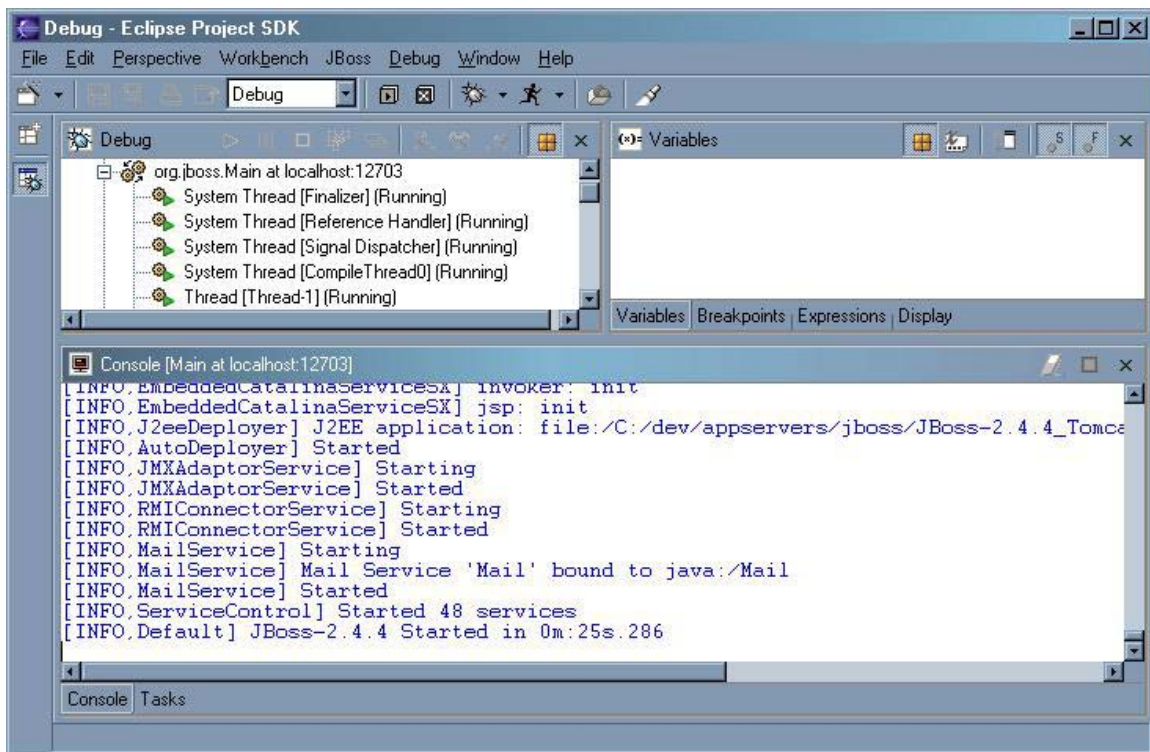



Figure 4

## Stopping JBoss

To stop the JBoss Server use either the **JBoss > Stop JBoss** menu action or the red UML node  on the toolbar. When the plugin receives a stop request it gracefully shuts down JBoss Server. This process usually takes between 5-10 seconds. You can monitor shutdown progress by observing log information JBoss sends to the Eclipse console. The shutdown process is complete when the JBoss Server thread group has been terminated.

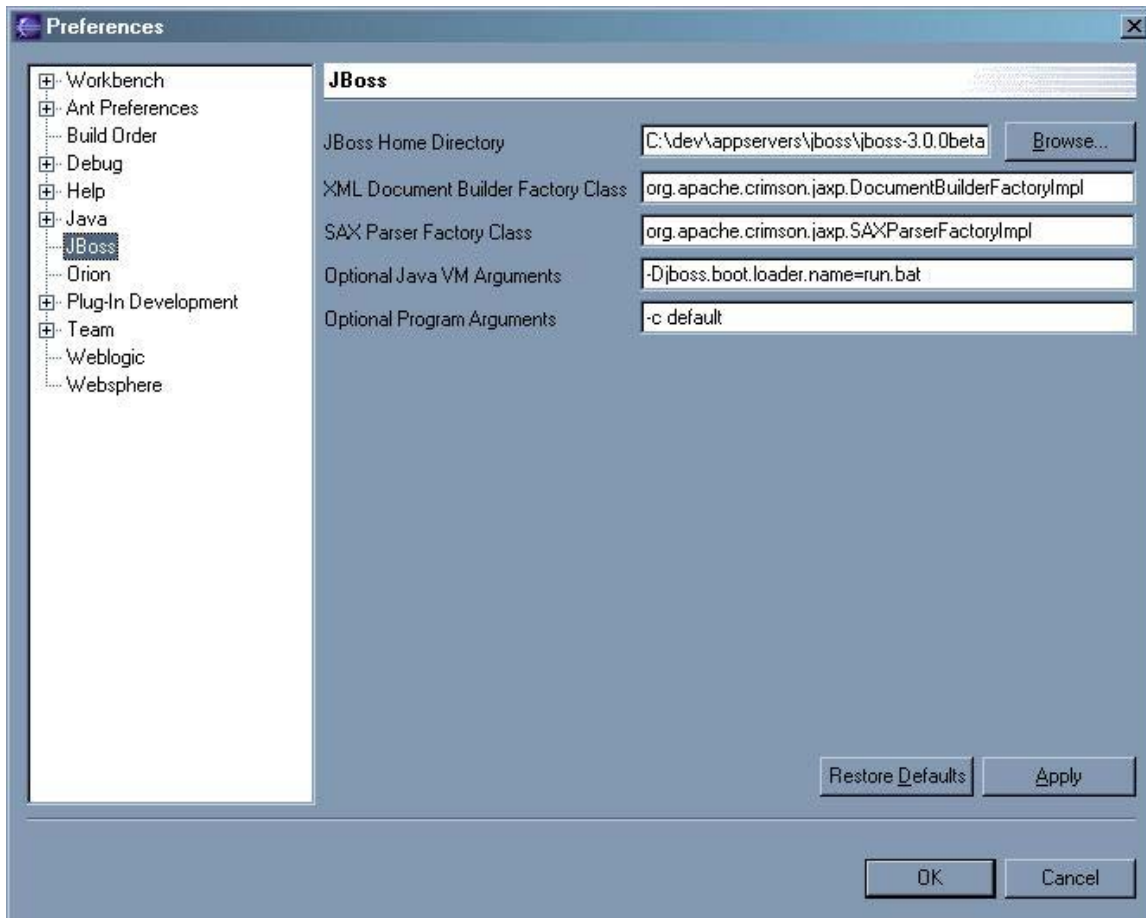
## JBoss 3.0 Alpha/Beta Notes

When launching the default JBoss 3.0 Alpha/Beta (JBoss) configuration from the commandline, outside of Eclipse, with JDK 1.4, multiple exceptions of the following form will be sent to the console and recorded in the log.

```
[2002-02-14 14:22:00,732,ServiceController,ERROR] Could not create MBean:
SingleJBoss:type=J2EEManagement,name=Manager
javax.management.NotCompliantMBeanException: Getter isStateManageable cannot co-exist
with getter getStateManageable
    at com.sun.management.jmx.Introspector.testConsistency(Introspector.java:328)
    at com.sun.management.jmx.Introspector.introspect(Introspector.java:264)
    ...
    at org.jboss.Main.<init>(Main.java:199)
    at org.jboss.Main$1.run(Main.java:411)
    at java.security.AccessController.doPrivileged(Native Method)
    at org.jboss.Main.main(Main.java:406)
```

When JBoss is run with JDK 1.3.1, there are no exceptions recorded. It appears that a couple of the MBeans that are deployed in the default configuration implement both a “getter” and a “boolean test” for the state manageable property and this violates the new JDK 1.4 MBean consistency compliance check. The same behavior can be seen when running JBoss through Eclipse, but obviously it is not the fault of Eclipse or the plugin.

With proper configuration, JBoss 3.0 can be run with the plugin just as successfully as JBoss 2.4.4. A sample of a configuration of the plugin properties, so they directly mimic the `run.bat` script settings, for this version of JBoss is depicted below. However, the VM argument setting, although currently set in the script, appears to be superfluous.



## Conclusion

If you followed these EASIE JBoss Plugin installation and configuration steps, you should have no problems starting and stopping JBoss Server from within Eclipse. If problems still persist, consult the resources section of this article then drop us a note at [EASIE support](#). Until next time, *happy programming*.

## Resources

- The [Eclipse.org](#) website to learn more about the Eclipse IDE.
- See the [JBoss website](#) for information on downloading a version of JBoss Server.
- The [EASIE JBoss Plugin](#) may be downloaded from the Genuitec website.
- The [EASIE Overview](#) introduces the Genuitec EASIE framework for Eclipse application Server plugins.