

Running JBoss in Eclipse on a Developer's Workstation

Table of contents

1 Audience.....	2
2 Purpose.....	2
3 Prerequisites.....	2
4 Running JBoss in Eclipse, allowing localhost-access only.....	2
4.1 Running a generated server from within Eclipse.....	3
4.2 Create a "localhost-only" server configuration in Eclipse.....	3
4.3 Test the server.....	4

1. Audience

Developers using the Eclipse IDE who wish to run JBoss on their development workstation.

2. Purpose

To deploy applications to JBoss for developer-testing on a local workstation, where JBoss is configured to respond only to requests from the local workstation.

3. Prerequisites

- Installed Eclipse IDE (cf. [Dev Box Setup](#) on this site)
- Installed Eclipse Web Tools Platform, an element of the [Callisto](#) Projects (cf. [Eclipse IDE Installation Instructions for Windows](#) on this site; "Eclipse Web Tools Platform" is listed as "Web and J2EE Development" in the Callisto installation/update dialogs)
- The appropriate version of JBoss checked-out from UC Berkeley's third-party repository into the local workspace (the directory in which JBoss is checked out is referred to in this documentation as `${JBOSS_HOME}`).
- One or more generated JBoss servers *configured for localhost access only*, which requires either:
 - Access to the UC Berkeley repository from which the current `servant` project can be checked-out (automated JBoss server generation) **AND** access to file(s) of encrypted passwords necessary to run `servant`; **AND** understanding of how `servant` is run to generate servers *that listen for requests only from localhost* (cf. [Generating JBoss 4 Server Instances](#) on this site); or,
 - Access to a known-good and up-to-date set of servers generated previously *and configured for localhost access only*, to be copied into `${JBOSS_HOME}/servers`; or,
 - Use no generated servers; rather, use a server that is part of the JBoss distribution (i.e., `all`, `default`, or `minimal`).

4. Running JBoss in Eclipse, allowing localhost-access only

This HowTo expands on the "Using the server tools" section of the [Web Tools Project documentation](#) in Eclipse Help. The instructions deviate from those provided by Eclipse in order to explain how to:

- run a locally- and script-generated server from within Eclipse; and,
- restrict JBoss so that it listens only for requests from the local machine (localhost, IP = 127.0.0.1)

4.1. Running a generated server from within Eclipse

If you intend to simply run the distributed default server (listening only on localhost), you can skip this section and go on to the next.

As of December 2006, the easiest way to run a generated JBoss server from within Eclipse is to copy it into the `${JBOSS_HOME}/server/default/` directory. It is noteworthy that [Eclipse Bug #128444](#) identifies users' interest to have a less clumsy way of accomplishing this goal.

To effect the copy (using your preferred file-manipulation tool, e.g., Windows Explorer):

1. Move (rename) the default server distributed with JBoss from its original location, `${JBOSS_HOME}/server/default/`. The suggested new location is `${JBOSS_HOME}/server/default-dist/`. *[The original default server must reside in the `${JBOSS_HOME}/server/default/` directory when servant is used to generate servers. An alternative to renaming the directory - and restoring it to the original path when it is necessary to run servant again - is to repeat the check-out of the original default server from the code repository if and when that becomes necessary.]*
2. Copy the generated server to be run from within Eclipse into the directory `${JBOSS_HOME}/server/default/`

4.2. Create a "localhost-only" server configuration in Eclipse

This set of steps configures a launcher for the server in `${JBOSS_HOME}/server/default/` that, when run, will listen for requests coming from localhost only, and ignore requests that originate on other machines (i.e., over the network).

Begin by opening the J2EE perspective in Eclipse (Window : Open Perspective : J2EE - click Other . . . if J2EE does not appear immediately upon selection of Open Perspective). Next, Click the Servers tab (unless you've changed how your J2EE perspective is organised, you'll find this tab among the views grouped at the bottom of the Eclipse window). Create a new server by following the steps and screenshots below:

Right click in the empty space on the Servers tab, and choose New : Server. Step through the JBoss 4.0 wizard, making sure to specify the path of your `${JBOSS_HOME}` (in the example, `d:\workspace\jboss-4.0.3`); defaults should suffice for all other field values:

server config wizard, 1 of 4

server config wizard, 2 of 4

server config wizard, 3 of 4

server config wizard, 3 of 4

The new server will be named something like "JBoss 4.0"; open the new server (double-click) and edit the name to something more specific and recognizable, such as "JBoss localhost-only":

open and edit (rename) the new server

Last, click the `Open launch configuration` link, and select the `Arguments` tab. In the "Program arguments" field, add this new argument: `-b 127.0.0.1`. The `-b` argument sets the bind address for all JBoss services. The complete program argument set will read:

```
-c default -b 127.0.0.1
```

open launch configuration and add argument

Click OK, and save the server configuration window (e.g., from the Eclipse menu, choose `File : Save`).

4.3. Test the server

Start the server to test it (right click and choose `Start` from the context menu). The server's console log will display in Eclipse's console window. When the server has fully launched, the end of the log will show the ports on which JBoss is listening. The ports (`http` and `https`, respectively) are expected to be:

- 127.0.0.1:8080 (will display in log as: "INFO [Http11Protocol] Starting Coyote HTTP/1.1 on http-127.0.0.1-8080")
- 127.0.0.1:8443 (will display in log as: "INFO [Http11Protocol] Starting Coyote HTTP/1.1 on http-127.0.0.1-8443")

Point a browser at the server, e.g., at `https://localhost:8443/`. You should see the JBoss Application Server answer your browser's request:

JBoss server in browser window

You are now ready to use the local instance of JBoss to deploy and test applications.