

JBoss At Work, Part 1: Installing and Configuring JBoss

by [Tom Marris](#) and [Scott Davis](#)

03/01/2005

Editor's note: Tom Marris and Scott Davis are currently finishing up their book JBoss At Work: A Practical Guide. The book will be available this summer from O'Reilly. In the meantime, they have written this series of articles that will make it easy to get up and running with JBoss if you are new to writing J2EE 1.4 applications, new to JBoss, or both.

[JBoss](#) is a popular J2EE application server. Several [recent surveys](#) put its market share at above 30 percent and single it out as the only application server whose market share is on the rise. JBoss is standards-compliant and a free download, making it an attractive choice whether you are running a production-quality site or looking for a learning tool.

The title of this article series, "JBoss At Work: A Practical Guide," is inspired by the authors' experience as back-to-back presidents of the [Denver Java Users Group](#). In our time running the group, the most common refrain we heard was, "I don't want to be an expert in it--I just want to make it work." It didn't matter what technology or API we were talking about at the time; the comment was always the same.

This sentiment reflects the busy schedules and tight deadlines that most working professionals face. Digging through a 1,000-page book on each new technology that comes down the pike is, quite simply, a luxury that few of us can afford. Our goal is to get you up and running in JBoss as quickly as possible.

In this, the first of three articles, we show you how to download and install JBoss. We explore the directory structure and show you how to add and remove services. Finally, we show you how to deploy an application to JBoss.

The second article in the series introduces the different parts of a J2EE application. We show you how JSPs and servlets fit into the application. We show you how they interact with EJBs. We show you how to use XDoclet and Ant to generate key configuration files and create an EAR (Enterprise ARchive) suitable for deployment.

The third article focuses on web services. We take the application from the second article and wrap the existing functionality in web services. XDoclet once again comes into play, making it almost trivial to expose your application's functions as web services. We create SOAP (Simple Object Access Protocol) clients to consume the web services from both a servlet and the command line.

Contents

[What is an Application Server?](#)

[Installing JBoss](#)

[Running JBoss](#)

[Exploring the Directory Structure](#)

[Server Configurations](#)

[Configuring JBoss Services](#)

[Deploying Applications to JBoss](#)

[Summary](#)

But before we can get to any of those advanced topics, we need to start at the very beginning.

What is an Application Server?

As you well know, compiled Java code cannot run on its own--it must be run inside of a JVM. J2EE classes require an additional step: they must be run inside of a J2EE container (which in turn runs inside of a JVM). The container provides an infrastructure and a framework in which your code runs. It handles the mundane low-level "plumbing" aspects of the application, freeing you to focus on the higher-level business development. An application server is nothing more than a loosely coupled collection of containers (or "services") that correspond to the various parts of the J2EE API.

The beauty of the J2EE is that it is a specification, not an implementation. In other words, Sun provides the technical guidelines for how each service should behave, but leaves it open for anyone to write a server that can host a J2EE application. This gives you, the developer, a wide variety of choices when it comes to selecting an application server. While the details of how to deploy a J2EE application may vary from server to server, you can rest assured that your compiled code will behave the same way regardless of the application server it is running in.

JBoss is an open source J2EE application server. It is J2EE 1.4 spec-compliant, which means that it offers the same level of functionality as its more expensive commercial counterparts. What this means to you is that the lessons learned here will be largely useful, regardless of the application server you are currently using. The most important thing to keep in mind is that cost does not equal value or reliability. The open source Apache web server runs [more than two-thirds](#) of all web sites today. JBoss isn't quite at that level of pervasiveness, but it shouldn't be ignored when considering enterprise-quality application servers just because it is free.

Installing JBoss

The latest JBoss release, 4.0.1, requires Java 1.4 or higher. Be sure that you have the 1.4 SDK (Software Development Kit) installed and not just the JRE (Java Runtime Environment). JBoss uses the compiler in the SDK to compile your JSP pages on the fly, among other things. To verify the version of Java you have installed, type `java -version` at a command prompt. Make sure that you have the SDK installed by typing `javac` at a command prompt. If necessary, go to [Sun's web site](#) for the latest JDK and installation instructions.

Once your Java environment is in place, you can go to the [JBoss web site](#) to download the application server. As you can see, there are a number of projects hosted at this site. Our focus is on the first one in the list, JBoss AS.

Since JBoss is written in Java, it can be run on any platform that has a working JVM. The download page has a long list of files archived in a variety of formats (.zip, .tar.gz, .bz2). The only difference between them is the utility used to bundle them up--there isn't a platform-specific JBoss distribution. Whichever one you download will work on Windows, Linux, Mac OS X, or any of the various flavors of Unix.

The current release is about 65MB, so make sure that you have a fast Internet connection or a lot of time on your hands. Don't download the source ("src") or the release candidate ("RC") bundles--look for the ones named `jboss-4.0.1.*`. Download one of them and unzip it to the directory of your choice. (For best results, avoid directory names with spaces in them like Program Files.) Create an environment variable called `JBOSS_HOME` that points to the directory.

There are no installation scripts to run. At this point, you are ready to fire up JBoss and take a look around.

Running JBoss

To start JBoss, change to the `JBOSS_HOME/bin/` directory and type `run` (`run.bat` on Windows, `run.sh` on Linux/Mac OS X/Unix). You should see a long stream of output similar to the abbreviated version shown in Figure 1.

```
rosencrantz:/Library/jboss/bin sdavis$ ./run.sh
=====

JBoss Bootstrap Environment
JBOSS_HOME: /Library/jboss
JAVA: /System/Library/Frameworks/JavaVM.framework/home/bin/java
JAVA_OPTS: -server -Xms128m -Xmx128m -Dprogram.name=run.sh
CLASSPATH: /Library/jboss/bin/run.jar:/System/Library/Frameworks/JavaVM.
framework/home/lib/tools.jar

=====

22:26:55,421 INFO [Server] Starting JBoss (MX MicroKernel)...
22:26:55,431 INFO [Server] Release ID: JBoss [Zion] 4.0.1 (build: CVSTag=
JBoss_4_0_1 date=200412230944)
22:26:55,432 INFO [Server] Home Dir: /Library/jboss-4.0.1
22:26:55,432 INFO [Server] Home URL: file:/Library/jboss-4.0.1/
22:26:55,433 INFO [Server] Library URL: file:/Library/jboss-4.0.1/lib/

[many lines deleted for clarity...]

22:26:56,299 INFO [ServerInfo] Java version: 1.4.2_05,Apple Computer, Inc.
22:26:56,299 INFO [ServerInfo] Java VM: Java HotSpot(TM) Client VM 1.4.2-38,
"Apple Computer, Inc."
22:26:56,299 INFO [ServerInfo] OS-System: Mac OS X 10.3.7,ppc
22:26:57,543 INFO [Server] Core system initialized
22:27:28,360 INFO [JkMain] Jk running ID=0 time=0/104 config=null
22:27:28,373 INFO [Server] JBoss (MX MicroKernel) [4.0.1 (build: CVSTag=
JBoss_4_0_1 date=200412230944)] Started in 32s:508ms
```

Figure 1. JBoss initialization output

Not surprisingly, the output shows the step-by-step process JBoss goes through when starting up. The first thing JBoss does is look for key environment variables such as `JBOSS_HOME`. It then begins booting the microkernel, the core JBoss process. Once the core is in place, it begins loading the individual J2EE services. After the services are loaded, JBoss deploys all WAR

(Web ARchive) and EAR files it finds in the *deploy/* directory. (More on this in just a moment.) JBoss is fully up and running when it reports *Started in XXs:YYms*.

Any service that can't be started will throw an exception during this process. One of the most common problems is trying to use default port assignments that conflict with applications already running on the system. For example, Tomcat comes bundled with JBoss. If you already have a running instance of Tomcat using the default ports (8005 and 8080), you will see a stream of errors in the console output from the embedded Tomcat instance, and it will fail to start. You'll learn how to change those ports after we get to know the directory structure.

Exploring the Directory Structure

There are a number of subdirectories under `JBOSS_HOME` (see Figure 2), but the two most important to the typical J2EE developer are the *bin/* and *server/* directories. The *bin/* directory contains the startup and shutdown scripts for JBoss, and *server/* contains the directories where we will eventually deploy our applications. The others are of secondary importance to us.

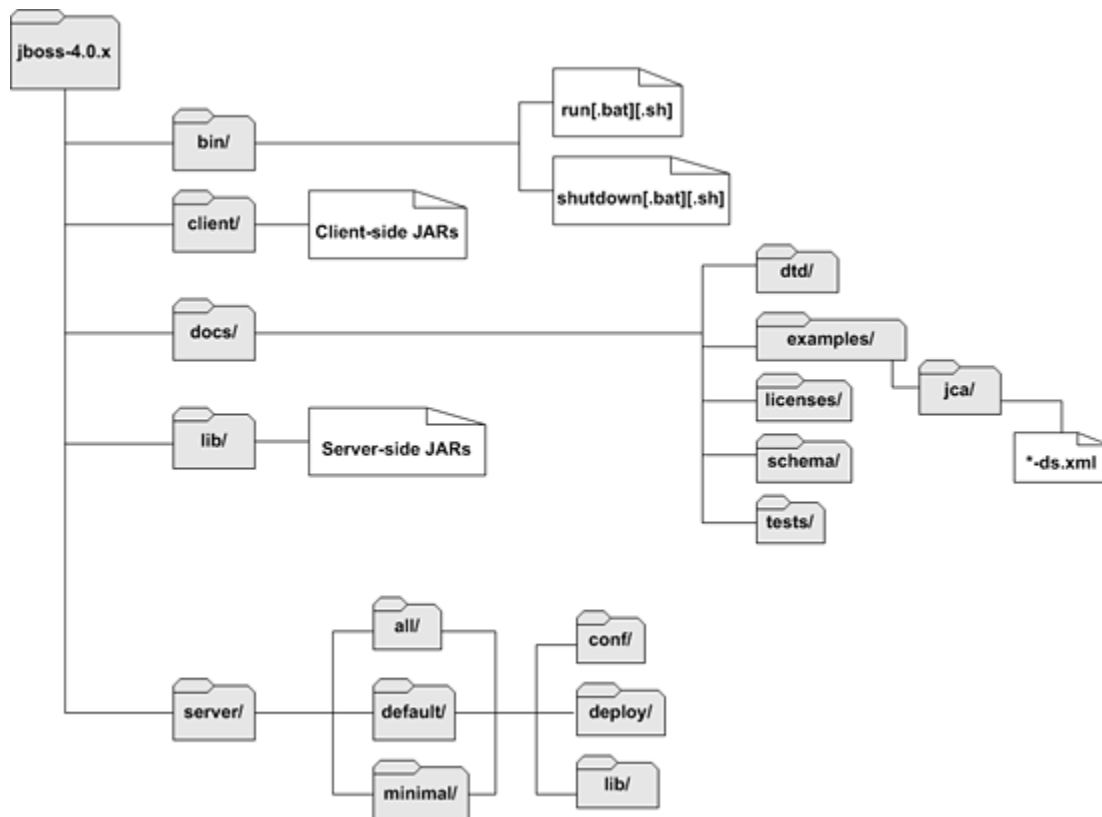


Figure 2. The JBoss directory structure

The *client/* directory contains JARs used by remote clients. (We discuss remote clients in the third article of this series.) The *docs/* directory contains various license and example files. The *lib/* directory contains the core JBoss JARs.

You've already had a chance to look around the *bin/* directory. The rest of this article focuses on the contents of the *server/* directory.

Server Configurations

We said earlier that an application server is a collection of J2EE services. JBoss makes it easy to mix and match these services by grouping them in a directory structure called a *Server Configuration*. If you look in the `JBOSS_HOME/server/` directory, you should see three named Server Configurations: `all`, `default`, and `minimal`.

If you don't specify a configuration on boot, JBoss uses the `default` configuration. The `default` configuration is the one that fulfills the J2EE 1.4 spec, so it is a reasonable place for most developers to start. The `minimal` configuration is bare dirt--nothing but a JNDI and a Log4j service. `All` is at the opposite end of the spectrum--everything in the `minimal` and `default` configurations, plus advanced services like clustering.

Let's shut down JBoss and start it with a different configuration. Press `Ctrl-C` in the JBoss console window to shut it down. Now type `run -c minimal`. You should see far fewer services launch and a dramatically shorter startup time.

It is easy to create your own Server Configuration. Simply copy one of the existing directory trees and give it a new name. For example, copy the *default/* directory to a new one named *myapp/*. Now start JBoss using the new configuration--press `Ctrl-C` to stop the currently running configuration, and then type `run -c myapp`.

Configuring JBoss Services

A server configuration has three main subdirectories: *conf/*, *deploy/*, and *lib/*. The *conf/* directory, as you might guess, contains configuration files. The main config file is *jboss-service.xml*. This is one place (but not the only place) where settings like port numbers can be found. The *lib/* directory contains the JAR files that make up the JBoss services. The *deploy/* directory contains the J2EE services, as well as any WAR and EAR files that should be deployed.

Services can be added and removed from a running JBoss instance through the magic of JMX (Java Management eXtensions). JMX is a framework that allows you to interact with live, running code. You can start and stop services, gather metrics, and even change parameters on the fly.

Services that implement the JMX interface are called "managed beans," or *MBeans*. Each of the J2EE services that run inside of JBoss is an MBean. (For more details on JMX, see [this tutorial](#).)

To see JMX in action, open a new terminal window next to the JBoss console, change to `JBOSS_HOME/server/myapp/deploy`, and create an *undeploy/* directory. (It can be named anything you'd like, but *undeploy/* is a common choice.) Move *hsqldb-ds.xml* to the *undeploy/* directory and watch the JBoss console. You should see the Hypersonic database service shut down and all

of the related services reconfigure themselves. Now move the file back from the *undeploy/* directory to the *deploy/* directory. Once again, JBoss recognizes the change in configuration and adjusts itself accordingly.

The XML file defining the MBean is another good place to look for port numbers. For instance, if you look at *hsqldb-ds.xml*, line 16 shows the port number that Hypersonic is configured to use.

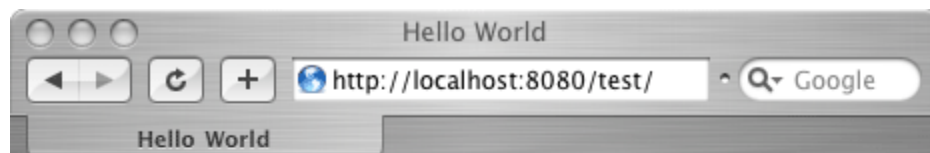
Complex MBeans that require more than a simple XML file and a JAR in the *lib/* directory can be deployed as a SAR (Service ARchive). Tomcat is a good example of this. Change to the *jbossweb-tomcat50.sar/* directory, and you'll find the traditional *server.xml* file used to configure Tomcat, along with all of its dependent libraries. (Tomcat ports can be changed here.)

Deploying Applications to JBoss

Your applications are deployed to the same directory as the MBeans. JBoss treats them as equals.

J2EE applications are generally bundled up as EARs or WARs. While each of these is technically nothing more than a simple JAR file, they have special internal directory structures and configuration files that must be present for the sake of the application server. (We discuss EARs and WARs in the next article in this series.)

Download *test.war* from [here](#) and copy it to the *deploy/* directory. In the JBoss console window, you should see the test application being deployed. Open a web browser and go to *http://localhost:8080/test* (see Figure 3).



Hello World

It's Tue Feb 01 23:01:35 MST 2005. Do you know where your dependent libraries are?

Figure 3. A custom application deployed to JBoss

If you move *test.war* to the *undeploy/* directory, JBoss will dynamically unload it, just as it did Hypersonic earlier. You don't have to do anything special to your application to gain this level of functionality--JBoss does it for you automatically.

Summary

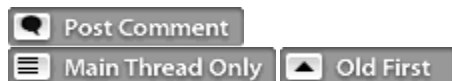
This was a whirlwind introduction to J2EE and JBoss. We downloaded and installed JBoss. We described the JBoss directory structure and how to start and stop the server. Finally, we deployed and undeployed both JBoss services and custom applications.

Next time, we'll introduce you to the basic pieces of a J2EE application and show you how to stick it in your EAR. (EAR **file**, that is.) Until then, have a great time using JBoss at work or on your open source project.

[Tom Marrs](#) is the principal and senior software architect at Vertical Slice, a consulting firm that designs and implements mission-critical business applications using the latest J2EE and open source technologies, along with providing architecture evaluation and developer training and mentoring services.

[Scott Davis](#) is a senior software engineer and instructor in the Denver, Colorado area.

View all [java.net Articles](#).



Showing messages 1 through 32 of 32.

- **Why!!!**
2007-06-28 04:44:51 sheilaleticia [[Reply](#) | [View](#)]

You say to not use .src why you say that?

We are trying to install and configure JBoss on Linux Red Hat Enterprise but we need to use src, we try but we are having a problem when we execute the command run.sh he say that we dont have the run.jar.

- **jboss don't start**
2007-03-09 16:40:13 torleif67 [[Reply](#) | [View](#)]

I am running jboss 4.0.5.GA on my MAC (10.4.8) system. When i start eclipse and try to run my server i get this message from the console and my server don't start.

```
07:00:34,274 INFO [Server] Starting JBoss (MX MicroKernel)...
07:00:34,282 INFO [Server] Release ID: JBoss [Zion] 4.0.5.GA (build:
CVSTag=Branch_4_0 date=200610162339)
07:00:34,305 INFO [Server] Home Dir: /Applications/jboss-4.0.5.GA
07:00:34,305 INFO [Server] Home URL: file:/Applications/jboss-4.0.5.GA/
07:00:34,307 INFO [Server] Patch URL: null
```

```

07:00:34,308 INFO [Server] Server Name: default
07:00:34,308 INFO [Server] Server Home Dir: /Applications/jboss-
4.0.5.GA/server/default
07:00:34,310 INFO [Server] Server Home URL: file:/Applications/jboss-
4.0.5.GA/server/default/
07:00:34,311 INFO [Server] Server Log Dir: /Applications/jboss-
4.0.5.GA/server/default/log
07:00:34,311 INFO [Server] Server Temp Dir: /Applications/jboss-
4.0.5.GA/server/default/tmp
07:00:34,312 INFO [Server] Root Deployment Filename: jboss-service.xml
07:00:37,556 INFO [ServerInfo] Java version: 1.5.0_07,Apple Computer, Inc.
07:00:37,562 INFO [ServerInfo] Java VM: Java HotSpot(TM) Client VM 1.5.0_07-
87,"Apple Computer, Inc."
07:00:37,609 INFO [ServerInfo] OS-System: Mac OS X 10.4.8,ppc
07:00:39,110 INFO [Server] Core system initialized
07:01:00,113 INFO [WebService] Using RMI server codebase: http://jesper-nykvists-
ibook-g4.local:8083/
07:01:00,236 INFO [Log4jService$URLWatchTimerTask] Configuring from URL:
resource:log4j.xml
07:01:06,034 INFO [SocketServerInvoker] Invoker started for locator: InvokerLocator
[socket://10.60.128.231:3873/]
07:01:18,931 INFO [ServiceEndpointManager] WebServices: jbossws-1.0.3.SP1
(date=200609291417)
07:01:21,016 INFO [SnmpAgentService] SNMP agent going active
07:01:22,969 INFO [CorbaNamingService] Naming:
[IOR:0000000000000002B49444C3A6F6D672E6F72672F436F734E616D696E672F4E6
16D696E67436F6E746578744578743A312E3000000000000020000000000000E800010
2000000000E31302E36302E3132382E323331000DC8000000114A426F73732F4E616D
696E672F726F6F74000000000000050000000000000080000000004A41430000000001
0000001C0000000000001000100000001050100010001010900000001050100010000002
100000060000000000000000010000000000000024000000200000007E0000000000000
0100000000E31302E36302E3132382E323331000DC90040000000000000000000000000100401
0008060667810201010100000000000000000000000000000000000000000000000000200000
00040000000000000001F000000040000000300000001000000200000000000000000002000
0002000000004000000000000001F0000000400000003]

```

I am fairly new to this so i ask for some help with this please.
Anybody knows what i am doing wrong here with my jboss?

Torleif

- o **jboss don't start**
2008-05-05 05:30:58 blastoffinc [[Reply](#) | [View](#)]

Will somebody please/edit or delete this comment? It makes the page unbearably wide.

- **Regarding Part 2**

2007-02-20 21:37:34 mknandhu [[Reply](#) | [View](#)]

hi

Please let me know where the rest of the document of jboss work can be found. ie part 2 and part 3

- **problem with installation**

2007-02-01 11:04:59 xgaffadi [[Reply](#) | [View](#)]

I am a brand new to jboss. I followed the process exactly, but i didn't get the webpage on <http://localhost:8080/test>. What could be the reason?

This is the error mssg:

HTTP Status 404 - /test

type Status report

message /test

description The requested resource (/test) is not available.

Apache Tomcat/5.5.9

- **problem with installation**

2007-02-01 11:42:45 sub_bh [[Reply](#) | [View](#)]

Hi ,

after downloading test.war, it will be saved as test.zip.

rename this file to test.war and place this war file under C:\jboss-4.0.5.GA\server\default\deploy\

now see the Jboss console, it will accept this test file and

now you can run the application by using this

<http://localhost:8080/test/>

- **problem with installation**

2007-02-01 11:18:51 xgaffadi [[Reply](#) | [View](#)]

well..found the solution myself..the link to download test file actually downloads test.zip NOT test.war.

I changed my test file to test.war, and it worked.

- **problem with installation**

2007-02-01 11:13:56 xgaffadi [[Reply](#) | [View](#)]

Actually here is more error info:

```
--- Packages waiting for a deployer ---
org.jboss.deployment.DeploymentInfo@a1ca0916 { url=file:/C:/jboss-
4.0.2/server/myapp/deploy/test.zip }
deployer: null
status: null
state: INIT_WAITING_DEPLOYER
watch: file:/C:/jboss-4.0.2/server/myapp/deploy/test.zip
altDD: null
lastDeployed: 1170357136448
lastModified: 1170357136448
mbeans:
```

```
--- Incompletely deployed packages ---
org.jboss.deployment.DeploymentInfo@a1ca0916 { url=file:/C:/jboss-
4.0.2/server/myapp/deploy/test.zip }
deployer: null
status: null
state: INIT_WAITING_DEPLOYER
watch: file:/C:/jboss-4.0.2/server/myapp/deploy/test.zip
altDD: null
lastDeployed: 1170357136448
lastModified: 1170357136448
mbeans:
```

- **hi...i am unable to use the bean class in my JSP using useBean...**
2006-10-29 22:36:10 sriramsharma [[Reply](#) | [View](#)]

I try to use the useBean attribute in my JSP with JBOSS appserver.

I get an error message on the explorer screen stating as follows

"The value for the useBean class attribute com.projects.beans.sampleBean is invalid."

The jsp tag that i used for usebean was as follows

```
<jsp:useBean id="newBean" class="com.projects.beans.sampleBean" scope="session"/>
```

Pls help me out to resolve this issue...

Thanks.

Sriram.

- **Thanks a lot...**
2006-10-29 08:18:03 sriramsharma [[Reply](#) | [View](#)]

as your part2 published?

I am eager to see it.
Thanks again.

- **wen do we gwt thw article on EAR'S**
2005-12-28 21:20:27 abhisek_chandra [[Reply](#) | [View](#)]

hi
the article is really inforamative
eagerly waiting for next article on EAR'S
anyways i hav a query:
How to have two insatnces of jboss
running on same m/c

- **wen do we gwt thw article on EAR'S**
2005-12-30 09:05:25 tmarrs [[Reply](#) | [View](#)]

We're glad you liked the article. Here's a link to the JBoss Wiki showing you how to run multiple JBoss instances on one machine:

<http://wiki.jboss.org/wiki/Wiki.jsp?page=ConfiguringMultipleJBossInstancesOnOneMachine>

- **next artical**
2005-11-16 23:13:19 sunju [[Reply](#) | [View](#)]

when would we supposed to see the next article

- **JBoss Portal**
2005-10-13 11:02:39 joelmerrin [[Reply](#) | [View](#)]

Very useful information; I look forward to purchasing your book. Will you include any information on JBoss Portal? If not, any plans for a book on this?
Thanks.

- **JBoss Portal**
2005-10-13 11:30:19 scottdavis99 [[Reply](#) | [View](#)]

Joel,

I'm glad you enjoyed this article. We're just about ready to release the next one in the series. (I know that we've been saying that for months, but *this* time we're serious. Honest!) The book should be shipping any day now, so we'll have more time to dedicate to the article series.

We don't talk about Portal in JBoss at Work. For more info on what we cover, see <http://www.jbossatwork.com>. To see the full table of contents and download a sample chapter, see [the O'Reilly catalog](#). Of course, you could [order the book](#) as well.

Thanks again,
Scott

- **very good, part 2?**
2005-10-05 03:59:43 elabidi [[Reply](#) | [View](#)]

Very good tutorial, congratulations

I am asking me too for part 2?

I have by the way a questio? which environment do you use to work with Jboss? The best I mean

- **very good, part 2?**
2005-10-05 06:14:58 tmarrs [[Reply](#) | [View](#)]

Thanks - we're glad you liked the article. Yes, we're coming out with a 2nd article soon - we were delayed because we had to finish the book. We expect to have article 2 available in about 2 weeks.

Which environment do I use to work with JBoss? It comes down to a good IDE and a good build tool.

I usually use Eclipse (a freely available Open Source IDE) or a commercial-grade IDE to write my code. I prefer Eclipse because it's just as good as the commercial products, it's free, and it has a great user/developer community behind it.

For the articles and the book (and for the last 4 years of my development projects :-), I've used Ant and XDoclet. Ant does the build, and XDoclet automates the generation of the J2EE and JBoss deployment descriptors. We'll cover our build environment in our third article and go into greater detail in the book.

But, I've recently (after we finished the book) started using Maven because it's at a higher level of abstraction than Ant. With Ant, you write a fair bit of code and you use low-level tasks <copy>, <delete>, <javac>, and so on. With Ant, you also have to come up with your own directory structure (which varies from project to project) and your third-party JARs (like log4j, and so on).

Maven imposes a standard directory structure, which is a good thing. Although some may argue that this limits creativity, I would argue that a standard structure alleviates a lot of pain due to all the poor build environments I've seen. Maven goes out and downloads all the third-party JARs you need, so you don't have to save them in your version control system. Maven's biggest feature is its extensive list of plug-ins that do all the build work for you. So, rather than writing lots of low-level Ant code to to run JUnit tests, you just invoke Maven's test plug-in to run your JUnit tests.

Anyway, Ant isn't "bad", but Maven takes care of most of the tasks I used to do

by hand.

Ant, Maven, and XDoclet aren't specific to JBoss - they work well in any Open Source Java and/or J2EE environment.

- **How to change the default deploy directory?**

2005-07-06 15:17:08 txie [[Reply](#) | [View](#)]

Good article! I should have read this long time ago.

My question is how to change the default deploy directory - \$SERVER_NAME/deploy?

Basically, our build&release engineer prefer a seperate location for our products deployable locations.

Thanks.

- **How to change the default deploy directory?**

2005-07-07 10:00:51 tmarrs [[Reply](#) | [View](#)]

I'm glad you liked our article.

You're right about the production deployment environment - it's generally a good idea to put your application-specific deployments (EAR/WAR/EJB JAR) into a separate server configuration.

Here's how to do it:

1) Copy the \$JBOSS_HOME/server/default directory (that shipped with JBoss) to something like "default.orig" so you always have a backup.

2) Copy the \$JBOSS_HOME/server/default directory (that shipped with JBoss) to <MyApp> - where MyApp is the name of your application.

3) Copy the following to \$JBOSS_HOME/server/<MyApp>/deploy:

a) Your EAR/WAR/EJB JAR

b) Any -ds.xml files (for your DataSources).

c) Any -service.xml files (for your JMS Destinations, and so on).

4) Start JBoss with your new server configuration:

a) CD to \$JBOSS_HOME/bin

b) type the following command: run.bat -c MyApp

If you're running on UNIX, the command would be: sh ./run.sh -c MyApp

You can post-fix the UNIX command with "&" to run JBoss in the background.

In order to run your new server configuration alongside the default server configuration, you must setup JBoss to run multiple instances on the same host

computer. You'll need to change port numbers in the configuration and - service.xml files for the new server configuration.

- **Good Article - Mine has more pictures though**

2005-05-05 16:40:09 blbrown [[Reply](#) | [View](#)]

[Another Jboss Setup introduction](#)

- **Good Article - Mine has more pictures though**

2005-05-06 09:39:36 tmarrs [[Reply](#) | [View](#)]

I liked your article. It's good to see more quality documentation on JBoss. More information on how to use JBoss only improves things for everyone.

- **Thanks Fellas!**

2005-04-21 08:03:36 romm [[Reply](#) | [View](#)]

The company I work with has recently moved over to JBoss for internal development. I really appreciate your article! It always helps to learn to the fundamentals to fill in the cracks where needed...

I do have a quick question (thought I'd ask, I might actually get lucky)...:

I'm attempting to configure SSL with JBoss and am not entirely clear on the best route to take with it -

1) Use Apache as the primary Web server to negotiate all SSL-related functionality. Will this work when securing jsps and servlets? EJB's? In what architectures is it useful to use Apache in this regard?

2) Stick with JBoss as the primary server and configure accordingly.

I understand this isn't a support forum; therefore, my heart won't be broken if I don't get an answer :-)... Any assistance/direction is appreciated!

Thanks again... Looking forward to your other articles.

Ryan

- **Thanks Fellas!**

2005-04-25 23:37:44 scottdavis99 [[Reply](#) | [View](#)]

I'm glad you enjoyed the article.

As far as SSL is concerned, either approach is acceptable (Tomcat or Apache). Conventional wisdom says that you should set up an Apache/Tomcat environment if you have static elements that would benefit from Apache's caching (images, static html pages, etc.) and dynamic elements that would benefit from having a

servlet container like Tomcat available.

So your SSL question is a logical extension of that -- do you have static objects that would benefit from SSL encryption, or are your encryption needs limited to Servlets/JSPs/etc?

If you have already committed to the Apache/Tomcat setup, I'd go ahead and let Apache handle the SSL. If you're having Apache redirect * to Tomcat (true confessions: I've done that more than once), then setting up a SSL connector in conf/server.xml is probably just as easy.

On a side note, thanks to everyone for your continued interest in the next articles in the series. Tom and I are up to our eyeballs in getting the final draft completed. The promised articles *will* get published as soon as we can catch our breath. Thanks again for being so patient.

Cheers,
s

- **Great Article**

2005-04-13 08:46:25 rvipani [[Reply](#) | [View](#)]

Simple and very helpful. Thanks for it. Would you also have details on how apache and Jboss work together and the needed connector config. Is there a reference I can go to for this.

- **Great Article**

2005-04-13 09:21:05 tmarrs [[Reply](#) | [View](#)]

We're glad you liked the article.

You can find the reference for connector configuration in the JBoss Application Server Guide:

<http://docs.jboss.org/jbossas/jboss4guide/r2/html/>

Scroll down to Section 9.7 - Using Apache with Tomcat.

- **where is part 2?**

2005-04-01 08:14:11 yuhong [[Reply](#) | [View](#)]

A real helpful article, I'd like to read the aprt 2, where is it?

- **where is part 2?**

2005-04-01 09:49:59 scottdavis99 [[Reply](#) | [View](#)]

I'm glad you enjoyed it. Part 2 should be out in the next week or two. Stay tuned... (grin)

- **where is part 2?**

2005-04-19 11:41:06 ctellez69 [[Reply](#) | [View](#)]

Hi!

When is due to be published part 2 of this excellent article?

Regards

- **where is part 2?**

2005-04-19 17:58:25 tmarrs [[Reply](#) | [View](#)]

We're glad you liked it. We're pushing hard to finish the book right now, but we hope to have the 2nd article out in the next couple of weeks.

- **Tomcat ports**

2005-03-10 11:37:04 patc890 [[Reply](#) | [View](#)]

Hi, I also think this was nice introduction for someone like me who knows nothing about jboss.

You mentioned changing the ports Tomcat uses if you have another instance of it running. I have so can some one tell me where to do this?

Thanks.

Pat.

- **Tomcat ports**

2005-03-11 11:34:50 scottdavis99 [[Reply](#) | [View](#)]

Pat,

Thanks for the kind words. As mentioned in the article, Tomcat ports can be changed in the server.xml file located in
\$JBOSS_HOME/server/default/deploy/jbossweb-tomcat50.sar

That file is an XML file that can be edited with any standard text editor. If you nose around in there, the port settings should be pretty obvious.

For more information on configuring Tomcat, the documentation is available [online](#).

Cheers, Scott

- **Thanks**

2005-03-06 03:24:15 doronorbach [[Reply](#) | [View](#)]

Very nice introduction to jboss - thanks.

D. Orbach

[javasight - java news & books](#)