

JBoss Enterprise Portal Platform 4.3 CP01 Installation Guide



Red Hat

Publication date: April, 2009

JBoss Enterprise Portal Platform 4.3 CP01 Installation Guide

Author Red Hat

Copyright © 2008 Red Hat, Inc

Copyright © 2008 Red Hat, Inc. This material may only be distributed subject to the terms and conditions set forth in the Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported License (which is presently available at <http://creativecommons.org/licenses/by-nc-sa/3.0/>).

Red Hat and the Red Hat "Shadow Man" logo are registered trademarks of Red Hat, Inc. in the United States and other countries.

All other trademarks referenced herein are the property of their respective owners.

The GPG fingerprint of the security@redhat.com key is:

CA 20 86 86 2B D6 9D FC 65 F6 EC C4 21 91 80 CD DB 42 A6 0E

1801 Varsity Drive
Raleigh, NC 27606-2072USA
Phone: +1 919 754 3700
Phone: 888 733 4281
Fax: +1 919 754 3701
PO Box 13588 Research Triangle Park, NC 27709USA

This Installation Guide documents relevant information regarding the installation of JBoss Enterprise Portal Platform

Preface	v
1. Document Conventions	v
1.1. Typographic Conventions	v
1.2. Pull-quote Conventions	vi
1.3. Notes and Warnings	vii
2. We Need Feedback!	viii
1. Introduction	1
1.1. Other Manuals	1
2. Getting Started	3
2.1. Pre-Requisites	3
2.1.1. Hardware and Operating System Requirements	3
2.1.2. Configuring Your Java Environment	3
2.2. Components of JBoss Enterprise Portal Platform	6
3. Installation	7
3.1. Download	7
3.2. Installing	7
3.3. Choosing a server configuration	7
4. Post Installation Configuration	9
4.1. Set JBOSS_HOME Environment Variable	9
4.2. Adjust memory settings	9
4.3. Post Installation Security Configuration	9
4.4. Disabling Authentication	11
5. Database Configuration	15
5.1. Installing an RDBMS JDBC connector	15
5.2. Datasource Descriptors	15
6. Databases	17
6.1. Using a MySQL Database	17
7. Test your Installation	21
8. Uninstall JBoss	23
A. Revision History	25

Preface

1. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

In PDF and paper editions, this manual uses typefaces drawn from the *Liberation Fonts*¹ set. The Liberation Fonts set is also used in HTML editions if the set is installed on your system. If not, alternative but equivalent typefaces are displayed. Note: Red Hat Enterprise Linux 5 and later includes the Liberation Fonts set by default.

1.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

Mono-spaced Bold

Used to highlight system input, including shell commands, file names and paths. Also used to highlight key caps and key-combinations. For example:

To see the contents of the file **my_next_bestselling_novel** in your current working directory, enter the **cat my_next_bestselling_novel** command at the shell prompt and press **Enter** to execute the command.

The above includes a file name, a shell command and a key cap, all presented in Mono-spaced Bold and all distinguishable thanks to context.

Key-combinations can be distinguished from key caps by the hyphen connecting each part of a key-combination. For example:

Press **Enter** to execute the command.

Press **Ctrl+Alt+F1** to switch to the first virtual terminal. Press **Ctrl+Alt+F7** to return to your X-Windows session.

The first sentence highlights the particular key cap to press. The second highlights two sets of three key caps, each set pressed simultaneously.

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **Mono-spaced Bold**. For example:

File-related classes include **filesystem** for file systems, **file** for files, and **dir** for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialogue box text; labelled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

¹ <https://fedorahosted.org/liberation-fonts/>

Choose **System > Preferences > Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, click the **Left-handed mouse** check box and click **Close** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).

To insert a special character into a **gedit** file, choose **Applications > Accessories > Character Map** from the main menu bar. Next, choose **Search > Find...** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit > Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in Proportional Bold and all distinguishable by context.

Note the **>** shorthand used to indicate traversal through a menu and its sub-menus. This is to avoid the difficult-to-follow 'Select **Mouse** from the **Preferences** sub-menu in the **System** menu of the main menu bar' approach.

Mono-spaced Bold Italic or ***Proportional Bold Italic***

Whether Mono-spaced Bold or Proportional Bold, the addition of Italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

To connect to a remote machine using ssh, type **ssh *username@domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh *john@example.com***.

The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount */home***.

To see the version of a currently installed package, use the **rpm -q *package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above — *username*, *domain.name*, *file-system*, *package*, *version* and *release*. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

When the Apache HTTP Server accepts requests, it dispatches child processes or threads to handle them. This group of child processes or threads is known as a *server-pool*. Under Apache HTTP Server 2.0, the responsibility for creating and maintaining these server-pools has been abstracted to a group of modules called *Multi-Processing Modules (MPMs)*. Unlike other modules, only one module from the MPM group can be loaded by the Apache HTTP Server.

1.2. Pull-quote Conventions

Two, commonly multi-line, data types are set off visually from the surrounding text.

Output sent to a terminal is set in Mono-spaced Roman and presented thus:

```
books      Desktop  documentation  drafts  mss    photos  stuff  svn
books_tests Desktop1  downloads      images  notes  scripts svgs
```

Source-code listings are also set in Mono-spaced Roman but are presented and highlighted as follows:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
    public static void main(String args[])
        throws Exception
    {
        InitialContext iniCtx = new InitialContext();
        Object          ref    = iniCtx.lookup("EchoBean");
        EchoHome        home   = (EchoHome) ref;
        Echo            echo   = home.create();

        System.out.println("Created Echo");

        System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
    }
}
```

1.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.



Note

A note is a tip or shortcut or alternative approach to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.



Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring Important boxes won't cause data loss but may cause irritation and frustration.



Warning

A Warning should not be ignored. Ignoring warnings will most likely cause data loss.

2. We Need Feedback!

If you find a typographical error in this manual, or if you have thought of a way to make this manual better, we would love to hear from you! Submit a report in [JIRA](#)² against the Version: 4.3, Component: *Documentation*. When submitting a bug report, be sure to mention the manual's identifier: Portal_User_Guide If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

² <http://jira.jboss.com/jira/browse/JBEPP>

Introduction

JBoss Enterprise Portal Platform is the open source implementation of the Java EE suite of services and Portal services running atop JBoss Enterprise Application Platform. It comprises a set of offerings for enterprise customers who are looking for preconfigured profiles of JBoss Enterprise Middleware components that have been tested and certified together to provide an integrated experience. Its easy-to-use server architecture and high flexibility makes JBoss the ideal choice for users just starting out with J2EE, as well as senior architects looking for a customizable middleware platform.

Because it is Java-based, JBoss Enterprise Portal Platform is cross-platform and easy to install and use on any operating system that supports Java. The readily available source code is a powerful learning tool to debug the server and understand it. It also gives you the flexibility to create customized versions for your personal or business use.

Installing JBoss Enterprise Portal Platform is simple and easy. You can have it installed and running in no time. This guide will teach you to install and uninstall JBoss.

1.1. Other Manuals

If you are looking for detailed product information refer to the manuals available online at <http://www.redhat.com/docs/manuals/jboss>.

Getting Started

2.1. Pre-Requisites

You must have adequate disk space to install JDK and JBoss Enterprise Portal Platform while also allowing enough space for your applications. Before installing JBoss Enterprise Portal Platform you must have a working installation of Java. Since JBoss is 100% pure Java you can have it working on any Operating System / Platform that supports Java. However, there are a few Operating System-specific issues that you should be aware of. Refer to <http://www.jboss.org/community/docs/DOC-10687> for more information.

2.1.1. Hardware and Operating System Requirements

For the latest information on supported Operating System / JVM combinations and supported Database platforms, please refer to <http://www.jboss.com/products/platforms/application/testedconfigurations>.

2.1.2. Configuring Your Java Environment

You must have a working installation of **JDK 1.5** before you install JBoss Enterprise Portal Platform. You can install the 32-bit or 64-bit JVM as per your requirements. In this guide we will show you how to install a 32-bit Sun JDK 5.0 using RHN, on a generic Linux Platform and Microsoft Windows Platform. But before we do that let's take a look at some of the benefits of using a 64-bit JVM.

Benefits of 64-bit JVM on 64-bit OS and Hardware:

- Wider datapath: The pipe between RAM and CPU is doubled, which improves the performance of memory-bound applications.
- 64-bit memory addressing gives virtually unlimited (1 exabyte) heap allocation. However large heaps affect garbage collection.
- Applications that run with more than 1.5GB of RAM (including free space for garbage collection optimization) should utilize the 64-bit JVM.
- Applications that run on a 32-bit JVM and do not require more than minimal heap sizes will gain nothing from a 64-bit JVM. Barring memory issues, 64-bit hardware with the same relative clock speed and architecture is not likely to run Java applications faster than their 32-bit cousin.

Installing and Configuring 32-bit Sun JDK 5.0 using RHN

- Java SDKs are provided by the Red Hat Enterprise Linux 5 Supplementary channel for your Linux variant and architecture. Depending on the architecture, the channel names are:

`rhel-i386-server-supplementary-5` and `rhel-x86_64-server-supplementary-5`

The `java-1.5.0-bea`, `java-1.5.0-ibm` and `java-1.5.0-sun` SDKs are available from these channels. Make sure you also install the `-devel` subpackages.

- Selecting `alternatives` for `java`, `javac` and `java_sdk_1.5.0` (setting `java_sdk_1.5.0` is optional).

This is only needed if you want to use the SysV service script and/or want this installed SDK to be the default java and javac in the system. This choice can often be overridden by setting the **JAVA_HOME** environment variable.

The **alternatives** system allows different versions of Java, from different sources, to co-exist on your system. You should make sure the desired one is selected so that the service script uses the one you want.

As root, issue the following command:

```
/usr/sbin/alternatives --config java
```

and make sure the desired one is selected (marked with a '+'), or select it by entering its number as prompted.

Make sure you do the same for **javac** and **java_sdk_1.5.0**. We recommend that all point to the same manufacturer and version.

Installing and Configuring 32-bit Sun JDK 5.0 on a generic Linux platform

- Download the Sun JDK 5.0 (Java 2 Development Kit) from Sun's website: http://java.sun.com/javase/downloads/index_jdk5.jsp. Choose "JDK 5.0 Update <x>" (where x is the latest update number) for download and then select "RPM in self-extracting" file for Linux¹. Read the instructions on Sun's website for installing the JDK.
- If you don't want to use SysV service scripts you can install the "self-extracting file" for Linux instead of choosing the "RPM in self-extracting" file. In that case you can skip the next step mentioned here. But it is recommended to use the SysV service scripts for production servers.
- Download and install the appropriate **-compat RPM** from **JPackage** [here](#)². Please ensure you choose a matching version of the **-compat** package to the JDK you installed.
- Create an environment variable that points to the JDK installation directory and call it **JAVA_HOME**. Add **\$JAVA_HOME/bin** to the system path to be able to run **java** from the command line. You can do this by adding the following lines to the **.bashrc** file in your home directory.

```
#In this example /usr/java/jdk1.5.0_11 is the JDK installation
directory.
export JAVA_HOME=/usr/java/jdk1.5.0_11
export PATH=$PATH:$JAVA_HOME/bin
```

Set this variable for the user account doing the installation and also for the user account that will run the server.

- If you have more than one version of JVM installed in your machine, make sure you are using the JDK1.5 installation as the default **java** and **javac**. You can do this using the alternatives system. The alternatives system allows different versions of Java, from different sources to co-exist on your system.

Select alternatives for java, javac and java_sdk_1.5.0

- As root, type the following command at the shell prompt and you should see something like this:

```
[root@vsr ~]$ /usr/sbin/alternatives --config java
There are 2 programs which provide 'java'.
Selection    Command
-----
   1          /usr/lib/jvm/jre-1.4.2-gcj/bin/java
*+ 2          /usr/lib/jvm/jre-1.5.0-sun/bin/java
Enter to keep the current selection[+], or type selection number:
```

Make sure the Sun version [**jre-1.5.0-sun** in this case] is selected (marked with a '+' in the output), or select it by entering its number as prompted.

- Repeat the same for javac and java_sdk_1.5.0.

```
[root@vsr ~]$ /usr/sbin/alternatives --config javac
There are 1 programs which provide 'javac'.
Selection    Command
-----
*+ 1          /usr/lib/jvm/java-1.5.0-sun/bin/javac
Enter to keep the current selection[+], or type selection number:
```

```
[root@vsr ~]$ /usr/sbin/alternatives --config java_sdk_1.5.0
There are 1 programs which provide 'java_sdk_1.5.0'.
Selection    Command
-----
*+ 1          /usr/lib/jvm/java-1.5.0-sun
Enter to keep the current selection[+], or type selection number:
```

You should verify that java, javac and java_sdk_1.5.0 all point to the same manufacturer and version.

**Note**

You can always override this step by setting the **JAVA_HOME** environment variable as explained in the previous step.

- Make sure that the **java** executable is in your path and that you are using an appropriate version. To verify your Java environment, type **java -version** at the shell prompt and you should see something like this:

```
[root@vsr ~]$ java -version
```

```
java version "1.5.0_11"  
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_11-b03)  
Java HotSpot(TM) Client VM (build 1.5.0_11-b03, mixed mode, sharing)
```

Installing and Configuring 32-bit Sun JDK 5.0 on Microsoft Windows

- Download the Sun JDK 5.0 (Java 2 Development Kit) from Sun's website: http://java.sun.com/javase/downloads/index_jdk5.jsp. Choose "JDK 5.0 Update <x>" (where x is the latest update number) for download and then select your Windows Platform options to perform the installation.
- Create an environment variable called **JAVA_HOME** that points to the JDK installation directory, for example: **C:\Program Files\Java\jdk1.5.0_11**. In order to run java from the command line add the **jre\bin** directory to your path, for example: **C:\Program Files\Java\jdk1.5.0_11\jre\bin**. To do this, open the Control Panel from the Start Menu, switch to Classic View if necessary, open the System Control Panel applet, select the Advanced Tab, and click on the Environment Variables button.

2.2. Components of JBoss Enterprise Portal Platform

For current information on the revision level of included components please refer to <http://www.jboss.com/products/platforms/portals/components>.

Installation

3.1. Download

You can download the zip file from the JBoss Customer Service Portal (CSP), located at <https://network.jboss.com>.

3.2. Installing

To install from a zip file, simply unzip the downloaded file to a directory of your choice. You can unzip the platform on any operating system that supports the zip format. Four types of server configurations will be included in your installation - minimal, default, production and all. The Getting Started Guide of JBoss Enterprise Application Platform as well as <http://www.jboss.org/community/docs/DOC-12942> explain in detail the different server configuration file sets.

- Unzip **jboss-epp-<release>.zip** to extract the archive contents into the location of your choice. You can do this using the JDK **jar** tool (or any other ZIP extraction tool).

```
[vsr]$ cd jbeppinstallationdir
[vsr]$ jar -xvf jboss-epp-<release>.zip
```

- You are done with the installation! You should now have a directory called **jboss-epp-<release>**. Refer to the Getting Started Guide to understand and explore the Platform Directory Structure.

3.3. Choosing a server configuration

The important differences as far as Portal is concerned are:

- **minimal** server configuration has no Portal bits. It's there for the sake of consistency with JBoss Enterprise Application Platform
- **default** server configuration has non-clustered Portal bits **jboss-portal.sar** in **deploy** folder.
- **all** server configuration has clustered Portal bits **jboss-portal-ha.sar** in **deploy** folder.
- **production** server configuration has clustered Portal bits **jboss-portal-ha.sar** and this is the sever that gets started by default. This server configuration has several optimizations such as logging, memory size etc. If you do not need clustered portal version and want to use this optimized server configuration, you simply need to copy the directory **jboss-portal.sar** from **default/deploy** to **production/deploy**. Don't forget to remove **jboss-portal-ha.sar** though.

Post Installation Configuration

4.1. Set JBOSS_HOME Environment Variable

On a Linux Platform

Create an environment variable that points to the installation directory (**JBOSS_DIST/jboss-as**) and call it **JBOSS_HOME**. Add **\$JBOSS_HOME/bin** to the system path to be able to run the server from the command line. You can do this by adding the following lines to the **.bashrc** file in your home directory.

```
#In this example /home/user/jboss-epp-4.3/jboss-as is the installation
directory.
export JBOSS_HOME=/home/user/jboss-epp-4.3/jboss-as
export PATH=$PATH:$JBOSS_HOME/bin
```

Set this variable for the user account(s) that will run the server.

On Microsoft Windows

Create an environment variable called **JBOSS_HOME** that points to the installation directory, for example: C:\Program Files\jboss-epp-4.3\jboss-as\. In order to run the server from the command line add the **bin** directory to your path, for example: C:\Program Files\jboss-epp-4.3\jboss-as\bin. To do this, open the Control Panel from the Start Menu, switch to Classic View if necessary, open the System Control Panel applet, select the Advanced Tab, and click on the Environment Variables button.

4.2. Adjust memory settings

The default configuration for the server to start with, if no other configuration is specified, is the production configuration. It is recommended to run the example Seam applications that are included with the documentation using the production configuration.

To avoid memory issues, adjust the memory settings before deploying the applications. You can do this by updating **JAVA_OPTS** settings in the file **JBOSS_DIST/jboss-as/server/production/run.conf** with these recommended values:

```
-Xms1303m -Xmx1303m -XX:PermSize=256m -XX:MaxPermSize=256m
```

4.3. Post Installation Security Configuration

When installed from the zip archive, all JBoss services require authentication to access most JBoss services, including administrative services. Additionally no user accounts are set up. This is to stop default user/password-based attacks.

Set up Accounts for jmx-console and the invokers by modifying:

```
$JBOSS_HOME/server/$CONFIG/conf/props/jmx-console-users.properties
```

Set up Accounts for web-console users by modifying:

```
$JBOSS_HOME/server/$CONFIG/deploy/management/console-mgr.sar/  
web-console.war/WEB-INF/classes/web-console-users.properties
```

Where **\$JBOSS_HOME** is the install directory and **\$CONFIG** is the server configuration being used.

Set SuckerPassword for JBoss Messaging:

JBoss Messaging makes internal connections between nodes in order to redistribute messages between clustered destinations. These connections are made with the user name of a special reserved user whose password is specified by this parameter **SuckerPassword** in the Server Peer configuration file:

```
$JBOSS_HOME/server/$CONFIG/deploy/jboss-messaging.sar/messaging-  
service.xml
```

Where **\$JBOSS_HOME** is the install directory and **\$CONFIG** is the server configuration being used. To avoid a security risk, you **MUST** specify the value of the attribute **SuckerPassword**, failing which the default value will be used. Any one who knows the default password will be able to gain access to any destinations on the server. The following fragment should be uncommented and modified:

```
<mbean code="org.jboss.jms.server.ServerPeer"  
  name="jboss.messaging:service=ServerPeer"  
  xmbean-dd="xmdesc/ServerPeer-xmbean.xml">  
  ...  
  ...  
  ...  
  ...  
  <!-- The password used by the message sucker connections to create  
connections.  
      THIS SHOULD ALWAYS BE CHANGED AT INSTALL TIME TO SECURE SYSTEM  
-->  
  <attribute name="SuckerPassword"></attribute>  
  
  ...  
  ...  
  ...  
</mbean>
```

4.4. Disabling Authentication

It is also possible to disable authentication on specific services. All specified paths in the sections below are relative to `$JBOSS_HOME`.

Disabling Authentication for JMX Console:

To disable authentication for the JMX console, edit the following file and comment out the security-constraint section:

```
server/$CONFIG/deploy/jmx-console.war/WEB-INF/web.xml
```

The following fragment should be commented out:

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>HtmlAdaptor</web-resource-name>
    <description>An example security config that only allows
users with the
role JBossAdmin to access the HTML JMX console web application
    </description>
    <url-pattern>/*</url-pattern>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
  </web-resource-collection>
  <auth-constraint>
    <role-name>JBossAdmin</role-name>
  </auth-constraint>
</security-constraint>
```

Disabling Authentication for Web Console:

To disable authentication for the Web console, edit the following file to comment out the security-constraint section:

```
server/$CONFIG/deploy/management/console-mgr.sar/web-console.war/WEB-INF/
web.xml
```

The following fragment should be commented out:

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>HtmlAdaptor</web-resource-name>
    <description>An example security config that only allows
users with the role JBossAdmin to access the HTML JMX console web
application
```

```
</description>
<url-pattern>/*</url-pattern>
<http-method>GET</http-method>
<http-method>POST</http-method>
</web-resource-collection>
<auth-constraint>
  <role-name>JBossAdmin</role-name>
</auth-constraint>
</security-constraint>
```

Disabling Authentication for HTTP Invoker:

To disable authentication for the http invoker, **JNDIFactory**, **EJBInvokerServlet**, and **JMXInvokerServlet** need to be removed from the security realm in the file:

```
server/$CONFIG/deploy/httpa-invoker.sar/invoker.war/WEB-INF/web.xml
```

For example, the security-constraint element should look as follows:

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>HttpInvokers</web-resource-name>
    <description>An example security config that only allows
users with the role HttpInvoker to access the HTTP invoker servlets
    </description>
    <url-pattern>/restricted/*</url-pattern>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
  </web-resource-collection>
  <auth-constraint>
    <role-name>HttpInvoker</role-name>
  </auth-constraint>
</security-constraint>
```

Disabling Authentication for JMX Invoker:

To disable authentication for the JMX invoker, edit the following file to comment out the security interceptor passthrough:

```
server/$CONFIG/deploy/jmx-invoker-service.xml
```

Locate the mbean section with the class **org.jboss.jmx.connector.invoker.InvokerAdaptorService**. In that section comment out the line that relates to authenticated users:

```
<descriptors>
```

```
<interceptors>
  <!-- Uncomment to require authenticated users -->
  <interceptor
code="org.jboss.jmx.connector.invoker.AuthenticationInterceptor"
    securityDomain="java:/jaas/jmx-console"/>
  <!-- Interceptor that deals with non-serializable results -->
  <interceptor
code="org.jboss.jmx.connector.invoker.SerializableInterceptor"
    policyClass="StripModelMBeanInfoPolicy"/>
</interceptors>
</descriptors>
```


Database Configuration

5.1. Installing an RDBMS JDBC connector

A relational database management system (RDBMS) Java Database Connectivity (JDBC) connector is required for JBoss Enterprise Portal Platform to communicate with a database. JBoss Enterprise Portal Platform comes bundled with Hypersonic SQL (HSQL) database so that users can get started up and running quickly without having to setup database initially. However, HSQL should not be used in production environment. Removing HSQL is simply a matter of removing **JBOSS_DIST/jboss-as/server/\$CONFIG/deploy/portal-hsql-ds.xml**. Test suites have been run against the Oracle® 9, Oracle® 10, Microsoft SQL Server®, MySQL 4, MySQL 5, and PostgreSQL 8 databases - these databases are recommended for production use. Please refer to your database documentation for the correct RDBMS JDBC connector. The MySQL RDBMS JDBC connector can be downloaded from <http://www.mysql.com/products/connector/j/>, and the PostgreSQL RDBMS JDBC connector can be downloaded from <http://jdbc.postgresql.org/download.html>.

Refer to the help files in each download for installation instructions. In most cases, the RDBMS JDBC connector comes as a .jar file. Copy this file into the **\$JBOSS_HOME/server/configuration/lib/** directory.

5.2. Datasource Descriptors

A Datasource descriptor is required for JBoss Portal to communicate with a database. Sample Datasource descriptor files are provided for the Hypersonic SQL, Microsoft SQL Server®, MySQL, Oracle®, and PostgreSQL databases. These can be found in the **/jboss-as/docs/portal/setup/** subdirectory where the EPP x.x.x binary distribution file was extracted to. For example, if EPP was extracted into the **/jboss/** directory, the sample Datasource descriptors would be in the **/jboss/jboss-epp-x.x.x/jboss-as/docs/portal/setup** directory.

The sample Datasource descriptors will work with default installations of external databases; however, you may need to configure the correct username and password. As well, the sample MySQL and PostgreSQL Datasource descriptors assume the name of the database to be **jbossportal**.

When copying a Datasource descriptor for MySQL, Microsoft SQL Server®, PostgreSQL, or Oracle® databases, change the `<user-name>portal</user-name>` and `<password>portalpassword</password>` sections to reflect the correct username and password for the user who has access to the **jbossportal** database.



Different Datasource Descriptors

Only have one Datasource descriptor in the **\$JBOSS_HOME/server/configuration/deploy/** directory. For example, if you want to use a MySQL database, remove the **portal-hsqldb-ds.xml** file, and other descriptors, if they exist.

The following is an example Datasource descriptor file for MySQL 5:

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

```
<local-tx-datasource>
  <jndi-name>PortalDS</jndi-name>
  <connection-url>jdbc:mysql://localhost:3306/jbossportal?
  useServerPrepStmts=false&amp;jdbcCompliantTruncation=false</
connection-url>
  <driver-class>com.mysql.jdbc.Driver</driver-class>
  <user-name><replaceable>portal</replaceable></user-name>
  <password><replaceable>portalpassword</replaceable></password>
</local-tx-datasource>
</datasources>
```


Databases

The following chapter contains examples for how to use MySQL databases to store JBoss Enterprise Portal user information.

6.1. Using a MySQL Database

Installing the MySQL RDBMS JDBC connector

The RDBMS JDBC MySQL connector is required for JBoss Portal to use a MySQL database:

1. Navigate to the [JDBC MySQL connector](#)¹ section of dev.mysql.com. Download the Source and Binaries (zip) file.
2. Unzip the `mysql-java-connector` into the desired directory. It does not matter which directory is used.
3. Change into the directory where the `mysql-java-connector` was extracted to. Copy the `mysql-connector-java-version-number-bin.jar` file into the `$JBOSS_HOME/server/configuration/lib/` directory.

Data Truncation Errors

If you are running MySQL 5, data truncation errors such as the following may occur:

```
ERROR JDBCExceptionReporter Data truncation: Data too long for column
'jbp_viewrealemail' at row 1
```

If you are running MySQL in strict mode, remove `STRICT_TRANS_TABLES` from the `sql-mode=` line, in the `my.cnf` file if you are running Linux, and from the `my.cfg` file if you are running Windows®. For further information, please see <http://wiki.jboss.org/wiki/Wiki.jsp?page=AvoidMySQL5DataTruncationErrors>.

Creating a MySQL Database

If you have just installed MySQL, make sure the MySQL server is running, and then run the following command to set the root user password:

```
mysqladmin -u root password 'new-root-user-password'
```

1. Run the following command to log in to MySQL. Enter the root user password when prompted:

```
mysql -u root -p
```

2. Use the **CREATE DATABASE** command to create a new `jbossportal` database. Note: the rest of this guide, and configuration examples following here after, assume the database name to

be `jbossportal`. If you change the database name, please change the name in all necessary configuration files:

```
mysql> CREATE DATABASE jbossportal;
```

The output should be similar to the following:

```
Query OK, 1 row affected (0.00 sec)
```

At this point, the **SHOW DATABASES;** command should display the **jbossportal** database:

```
mysql> SHOW DATABASES;
+-----+
| Database          |
+-----+
| information_schema |
| jbossportal       |
| mysql             |
+-----+
3 rows in set (0.00 sec)
```

It is safe to ignore the other databases.

3. Add a new user and give that user access to the `jbossportal` database. The following example adds a new user named `portal`, with the password `portalpassword`, and gives the `portal` user access to the `jbossportal` database. It is recommended that you change the password. Change `portalpassword` to change the password:

```
mysql> GRANT ALL PRIVILEGES ON jbossportal.* TO 'portal'@'localhost'
IDENTIFIED BY 'portalpassword' WITH GRANT OPTION;
```

The output should be similar to the following:

```
Query OK, 0 rows affected (0.00 sec)
```

If you changed the `portal` user's password, also change the password in the MySQL Datasource descriptor.

MySQL Datasource Descriptor

To install the MySQL Datasource descriptor:

1. Change into the **setup** subdirectory where JBoss Portal was extracted to. Copy the MySQL Datasource descriptor into the **\$JBOSS_HOME/server/configuration/deploy/** directory.

If you are running MySQL 5, use **portal-mysql5-ds.xml**, otherwise use **portal-mysql-ds.xml**.

2. Update the `<user-name>portal</user-name>` and `<password>portalpassword</password>` sections in the Datasource descriptor to reflect the correct username and password for the user who has access to the jbossportal database.

MySQL Table Errors

Tables are set up automatically when JBoss Portal first starts. Initially, two users will be stored in the jbossportal MySQL database: a standard user with the username user, and the password user, and an administrative user with the username admin, and the password admin. Since the database tables are created during initial startup, errors such as the following may occur, but are safe to ignore:

```
WARN [JDBCExceptionReporter] SQL Error: 1146, SQLState: 42S02
ERROR [JDBCExceptionReporter] Table 'jbossportal.jbp cms_repositoryentry'
  doesn't exist
WARN [JDBCExceptionReporter] SQL Error: 1146, SQLState: 42S02
ERROR [JDBCExceptionReporter] Table 'jbossportal.jbp cms_version_refs'
  doesn't exist
WARN [JDBCExceptionReporter] SQL Error: 1146, SQLState: 42S02
ERROR [JDBCExceptionReporter] Table 'jbossportal.jbp cms_wsp_node'
  doesn't exist
WARN [JDBCExceptionReporter] SQL Error: 1146, SQLState: 42S02
ERROR [JDBCExceptionReporter] Table 'jbossportal.jbp cms_cmseentry'
  doesn't exist
WARN [JDBCExceptionReporter] SQL Error: 1146, SQLState: 42S02
```


Test your Installation

After you have installed the JBoss Enterprise Portal Platform, it is wise to perform a simple startup test to validate that there are no major problems with your Java VM/operating system combination. Make sure you have set the **JBOSS_HOME** environment variables as explained in [Post_Installation_Configuration](#). To test your installation, move to **JBOSS_DIST/jboss-as/bin** directory and execute the **run.bat** (for Windows) or **run.sh** (for Linux) script, as appropriate for your operating system. Your output should look like the following (accounting for installation directory differences) and contain no error or exception messages:

```
[user@localhost bin]$ ./run.sh
=====

JBoss Bootstrap Environment

JBOSS_HOME: /home/user/jboss-eap-4.3/jboss-as

JAVA: /usr/java/jdk1.5.0_11/bin/java

JAVA_OPTS: -Dprogram.name=run.sh -server -Xms1503m -Xmx1503m -
Dsun.rmi.dgc.cli ent.gcInterval=3600000
          -Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.net.prefer
IPv4Stack=true

CLASSPATH: /home/user/jboss-eap-4.3/jboss-as/bin/run.jar:/u sr/java/
jdk1.5.0_11/lib/tools.jar

=====

18:45:49,550 INFO  [Server] Starting JBoss (MX MicroKernel)...
.
.
.
.
18:45:50,449 INFO  [ServerInfo] Java version: 1.5.0_11,Sun Microsystems
Inc.
18:45:50,449 INFO  [ServerInfo] Java VM: Java HotSpot(TM) Server VM
1.5.0_11-b03 ,Sun Microsystems Inc.
18:45:50,449 INFO  [ServerInfo] OS-System: Linux 2.6.9-42.0.3.EL,i386
18:45:51,824 INFO  [Server] Core system initialized
18:45:59,622 INFO  [WebService] Using RMI server codebase:
http://127.0.0.1:8083 /
18:45:59,659 INFO  [Log4jService$URLWatchTimerTask] Configuring from URL:
resource:jboss-log4j.xml
```



Note

Note that there is no "Server Started" message shown at the console when the server is started using the **production** profile, which is the default profile used when no

other is specified. This message may be observed in the `server.log` file located in the `server/production/log` subdirectory.

Now open `http://localhost:8080/portal` in your web browser. (Make sure you dont have anything else already on your machine using that port).¹ The contents of your page should look similar to this: *Figure 7.1, "Test your Installation"*.

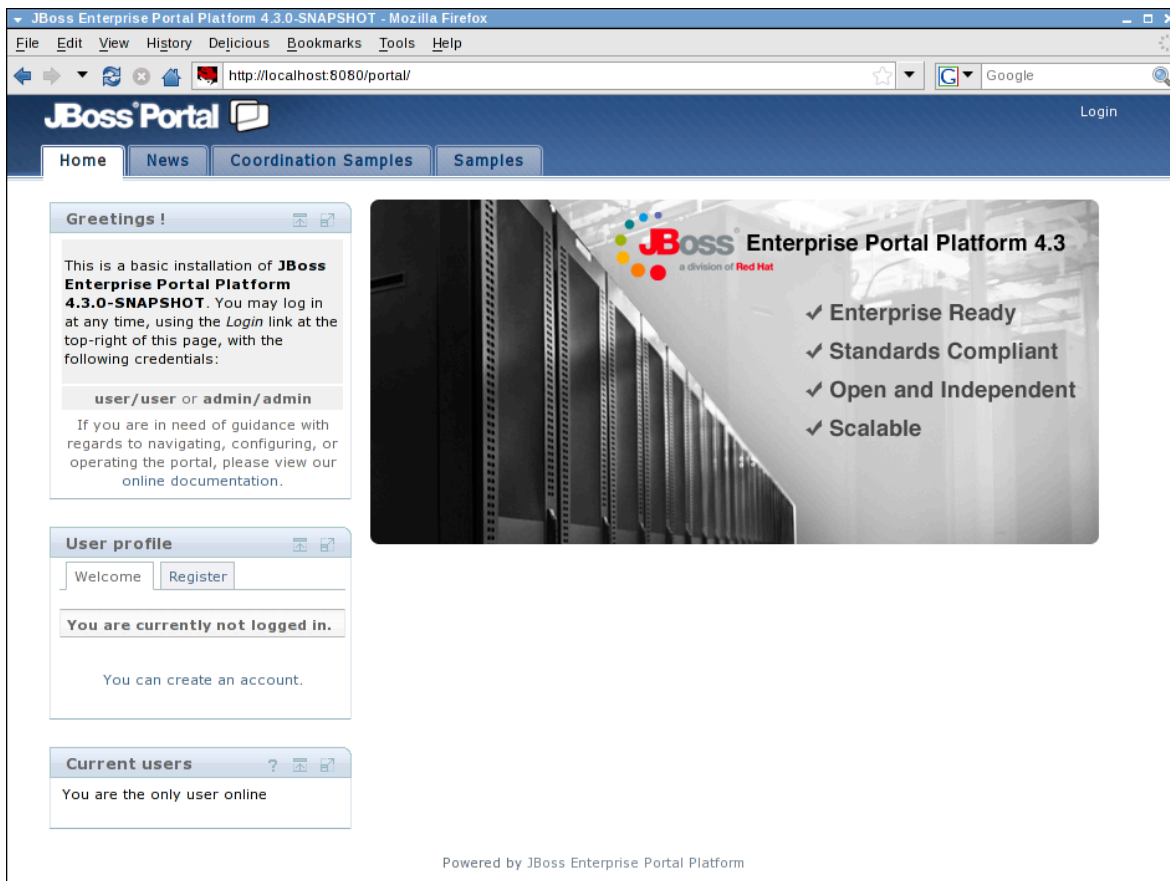


Figure 7.1. Test your Installation

You are now ready to use the JBoss Enterprise Portal Platform. Refer to the User Guide and Reference Guide for more information about the platform layout and example applications showcasing JBoss Portal in action.

¹ Note that on some machines, the name localhost won't resolve properly and you should use the local loopback address 127.0.0.1 instead.

Uninstall JBoss

As JBoss Enterprise Portal Platform was installed from a zipfile, you may uninstall it by simply deleting the JBoss Enterprise Portal Platform installed directory.

Appendix A. Revision History

Revision History
Revision 1.0

