



JBoss Enterprise Application Platform 4.3

Installation Guide

for Use with JBoss Enterprise Application Platform 4.3

Edition 4.3.10

JBoss Enterprise Application Platform 4.3 Installation Guide

for Use with JBoss Enterprise Application Platform 4.3
Edition 4.3.10

Red Hat Documentation Group

Legal Notice

Copyright © 2011 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This Installation Guide documents relevant information regarding the installation of JBoss Enterprise Application Platform 4.3 and its patch releases.

Table of Contents

CHAPTER 1. INTRODUCTION	3
1.1. OTHER MANUALS	3
CHAPTER 2. GETTING STARTED	4
2.1. PRE-REQUISITES	4
2.1.1. Hardware Requirements	4
2.1.2. Supported Installations	5
2.1.3. Configuring Your Java Environment	5
2.2. COMPONENTS OF JBOSS ENTERPRISE APPLICATION PLATFORM	5
CHAPTER 3. INSTALLATION ALTERNATIVES	6
CHAPTER 4. INSTALLATION USING THE GRAPHICAL INSTALLER	7
CHAPTER 5. INSTALLATION WITH ZIP DOWNLOAD	11
5.1. DOWNLOAD	11
5.2. INSTALLATION	11
CHAPTER 6. RPM INSTALLATION VIA RED HAT NETWORK	12
6.1. RED HAT NETWORK	12
6.2. INSTALL ON RED HAT ENTERPRISE LINUX 4	12
6.3. INSTALL ON RED HAT ENTERPRISE LINUX 5	13
CHAPTER 7. POST INSTALLATION CONFIGURATION	15
7.1. SET JBOSS_HOME ENVIRONMENT VARIABLE	15
On a Linux Platform	15
On Microsoft Windows	15
7.2. ADJUST MEMORY SETTINGS	15
7.3. POST INSTALLATION SECURITY CONFIGURATION	15
7.3.1. Security Configuration: JMX Console, Admin Console, HttpInvoker	16
7.3.2. Securing the HTTPInvoker	17
7.3.3. Security Configuration: JBoss Messaging	18
7.4. POST INSTALLATION SECURITY CONFIGURATION	18
Set up Accounts for jmx-console and the invokers by modifying:	18
Set up Accounts for web-console users by modifying:	19
Set SuckerPassword for JBoss Messaging:	19
7.5. RUN THE APPLICATION SERVER AS A SERVICE	19
7.6. DISABLING AUTHENTICATION	22
7.7. THE PRODUCTION CONFIGURATION AND CLUSTERING	24
CHAPTER 8. UNINSTALL JBOSS	26
CHAPTER 9. TEST YOUR INSTALLATION	27
APPENDIX A. INSTALLING A JAVA DEVELOPMENT KIT ON RED HAT ENTERPRISE LINUX	29
A.1. OPENJDK ON RED HAT ENTERPRISE LINUX 5	29
A.2. IBM JAVA DEVELOPMENT KIT ON RED HAT ENTERPRISE LINUX 5	29
A.3. IBM JDK ON RED HAT ENTERPRISE LINUX AS/ES 4	30
A.4. SETTING THE DEFAULT JDK WITH THE /USR/SBIN/ALTERNATIVES UTILITY	30
APPENDIX B. REVISION HISTORY	32

CHAPTER 1. INTRODUCTION

JBoss Enterprise Application Platform is the open source implementation of the Java EE suite of services. 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. It's 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 Application Platform is cross-platform, 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 Application 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>.

CHAPTER 2. GETTING STARTED

2.1. PRE-REQUISITES

The following sections discuss the hardware and software requirements to run the JBoss Application Server.

2.1.1. Hardware Requirements

When considering the minimum hardware required to run the JBoss Application Server, it is necessary to consider both the hardware required to successfully install the application and the additional requirements to support an operational server which varies depending on the demand and the types of applications being served.

Minimum Installation Requirements

The minimum hardware required to support the installation of the JBoss Application Server is a 240MB hard disk drive. Additional space is required for the installation of the JDK upon which the the JBoss Application Server depends. The JDK installation size is currently up to 150MB.

Minimum Operational Requirements

The minimum hardware required to support an operational JBoss Application Server varies depending on the following:

- the size and complexity of the applications being served;
- the demand placed on the server by the number and frequency of client requests;
- the server configuration including the selected log files, their designated size and general server tuning.

The following discussion relates to the deployment of a simple application on a server experiencing minimal demand. In view of this, the absolute minimum requirements for an operational server are:

- Disk Space: 1GB
 - The default server log file storage configuration is 500MB.
 - The remaining 500MB is allocated to the server installation (240MB), the required JDK (150MB) and some additional space for applications (110MB).
- CPU: Intel Pentium Processor @ 1GHz
 - Core 2 Duo, Core 2 Quad and Intel Xeon chips will improve the performance of servers which experience high demand.
- RAM: 128 MB
 - RAM installations of 1GB or more will be required to run a server upon which small to medium applications are deployed. 4GB or more is preferable for larger applications or to run a GUI server interface.

**NOTE**

Tests were performed to establish the minimum memory requirements of 128MB based on JMX-Console, which is a small web application packaged with the standard distribution, and a small sub-set of tests from the test suite. These tests simulate small applications deployed on the server. For these tests, the server's configuration was modified via the JAVA_OPTS parameter "-Xms(MEMORY)m -Xmx(MEMORY)m".

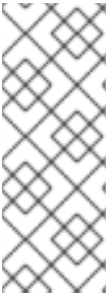
**IMPORTANT**

A server's performance must be viewed in light of the applications deployed on the server, the demand placed on the server by client requests and any post-installation server configuration or tuning.

2.1.2. Supported Installations

For the latest information on supported Operating System / JVM combinations and supported Database platforms, refer to

<http://www.jboss.com/products/platforms/application/testedconfigurations>.

**NOTE**

The JBoss Application Server requires a working installation of Java and will work on any Operating System / Platform that supports Java. However, there are a few issues relating to Operating Systems which should be noted. Please refer to the following link for more information:

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

2.1.3. Configuring Your Java Environment

You must correctly configure your Java environment before installing JBoss Enterprise Application Platform to ensure the platform installs correctly. Follow the procedures in [Appendix A, *Installing a Java Development Kit on Red Hat Enterprise Linux*](#) prior to proceeding to [Chapter 6, *RPM Installation via Red Hat Network*](#).

2.2. COMPONENTS OF JBOSS ENTERPRISE APPLICATION PLATFORM

For current information on the revision level of included components please refer to http://www.redhat.com/docs/en-US/JBoss_Enterprise_Application_Platform/4.3.cp07/html-single/Release_Notes/index.html.

CHAPTER 3. INSTALLATION ALTERNATIVES

You can install the JBoss Enterprise Application Platform in one of these three modes:

- *Graphical Installer*

Using the Graphical Installer can simplify the installation and configuration process for non-sophisticated users. In addition to the basic installation, the installer provides you with basic configuration capabilities, allows you to control whether the JMX interfaces are secured and the ability to enter console username/password.

- *ZIP download*

In this form of installation, simply unzip the downloaded zip file to the directory of your choice. You can unzip the platform on any operating system that supports the zip format.

- *RPM download*

In this form of installation, you can automatically install the platform on a Red Hat Enterprise Linux system using Red Hat Network.

When you install from the installer, you get a smaller install image that is more tuned for your environment. However, the directory structure will be slightly different than when using the rpm/zip archive.

Four types of server configurations will be included in your installation - *minimal*, *default*, *production* and *all*. The Getting Started Guide explains in detail the different server configuration file sets.

CHAPTER 4. INSTALLATION USING THE GRAPHICAL INSTALLER

Launching the Graphical Installer

- The installer is an executable JAR file named `enterprise-installer-<release>.jar`
- On many operating systems, you can run executable JARs by double-clicking them. If your system doesn't support that, you can run the installer directly from the command line:

```
[vsr]$ java -jar enterprise-installer-<release>.jar
```

- The installer will then guide you through a series of installation steps explained in detail in the following section. You can quit the installation process any time before you confirm the final installation.

Installation Steps

- Choose the language to be used for the installation instructions and press the OK button.
- Read the License Agreement carefully. If you agree to the terms of the agreement select "I accept the terms of this license agreement" option and press the Next button. If you do not accept to the terms then choose "I do not accept to the terms of this license agreement" option. If you choose the second option you will not be able to proceed with the installation.
- Select the Installation Path where you would like JBoss Enterprise Application Platform to be installed. You can either type the complete path or browse for it. If the directory corresponding to the path you entered does not exist, the installer will create the target directory in the specified path. If the directory corresponding to the path you entered exists already, the installer will overwrite the contents of the directory. In either case the installer will prompt you to confirm the action.



NOTE

It does not matter where on your system you install JBoss Enterprise Application Platform, however note that installing JBoss Enterprise Application Platform into a directory that has a name containing spaces causes problems in some situations with Sun-based VMs. This is caused by bugs with file URLs not correctly escaping the spaces in the resulting URL.

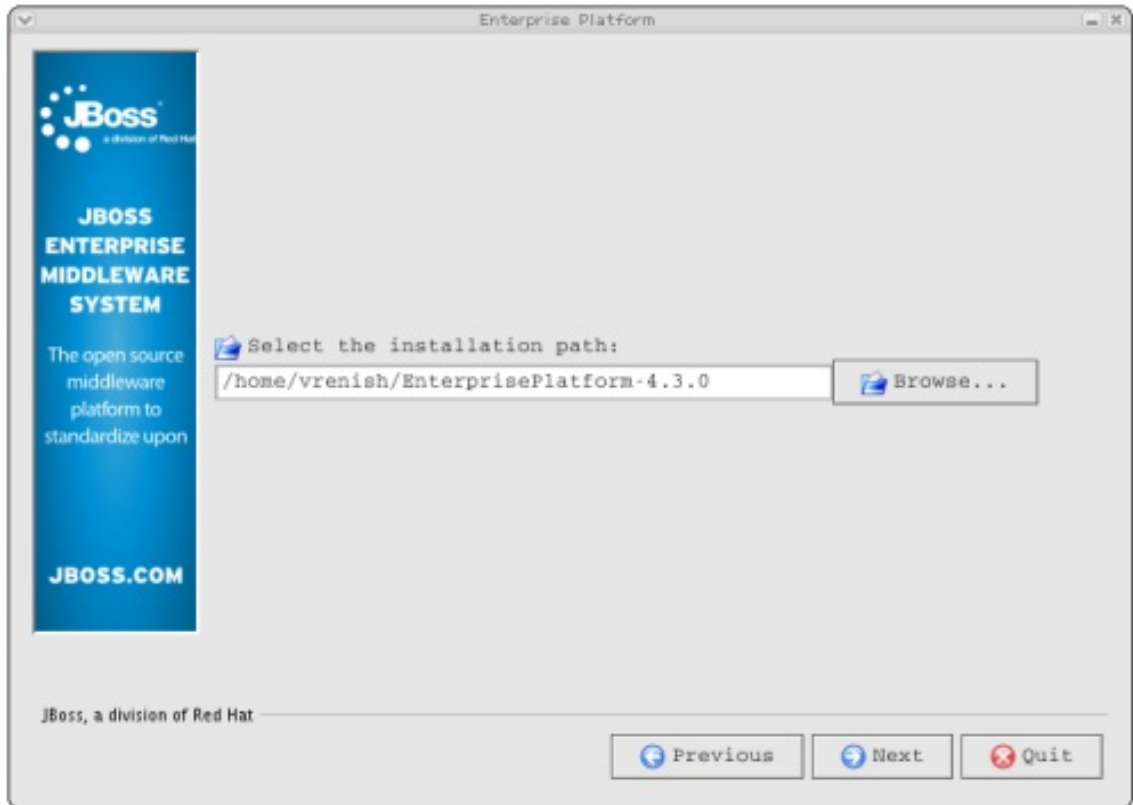


Figure 4.1. Select Installation Path

- **Configure JMX Security** - In this section you can control the security settings for the JMX interfaces. You can choose to secure the following services:

jmx-console.war , web-console.war , jmx-invoker-services , http invoker

It is recommended that you click to enable security for all services and change the username/password from the default **admin/admin** values.

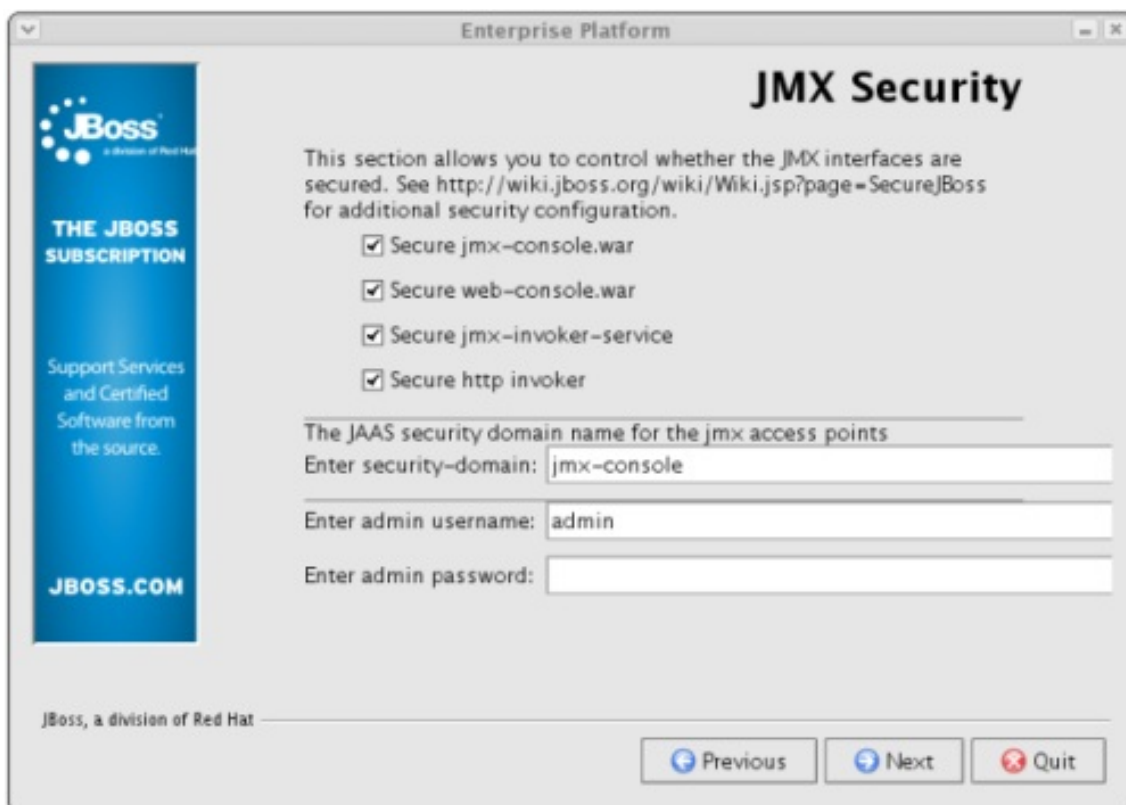


Figure 4.2. Configure JMX Security



NOTE

The JMX and web console would ask for your password if you install from the GUI installer.

- You can find the latest release notes available here: <http://www.redhat.com/docs/manuals/jboss>. Read the release notes information carefully to know about important compatibility and configuration issues, library updates, feature requests and bugs, links to additional documentation and license information.
- JBoss Enterprise Application Platform is now ready to install. Verify the installation path displayed in the summary screen before you hit the Next button. Pressing the Next button will begin the installation.
- You can add the JBoss Platform menu in the XDG menu and also choose to create shortcuts on the desktop. If you select the "Create additional shortcuts on the desktop" checkbox, the installer will create the following shortcuts:
 - Shortcut to Start the application server
 - Shortcut to Stop the application server
 - Shortcut to the documentation
 - Shortcut to the JMX Console
 - Shortcut to start the demo application

If you run the installer as the root user you can choose to create these shortcuts for all users.



Figure 4.3. Create Desktop Shortcuts

- You are done with the installation! You should now have a directory called **EnterprisePlatform-*<release>***, or whatever other name you specified. To explore the Platform directory structure and to understand the layout in detail, refer to the Getting Started Guide.

CHAPTER 5. INSTALLATION WITH ZIP DOWNLOAD

5.1. DOWNLOAD

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

5.2. INSTALLATION

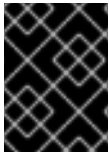
In this form of installation, simply unzip the downloaded zip file to the directory of your choice. You can unzip the platform on any operating system that supports the zip format.

- Unzip `jboss-eap-<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 jbeapinstallationdir
[vsr]$ jar -xvf jboss-eap-<release>.zip
```

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

CHAPTER 6. RPM INSTALLATION VIA RED HAT NETWORK



IMPORTANT

Ensure you have followed the instructions in [Appendix A, *Installing a Java Development Kit on Red Hat Enterprise Linux*](#) before proceeding with RPM installation.

6.1. RED HAT NETWORK

Red Hat Network (<http://rhn.redhat.com>) is a complete systems management platform for Red Hat Enterprise Linux. RHN provides update, management, and provisioning functionality to Red Hat Enterprise Linux Customers. Red Hat Network is the primary delivery mechanism for subscription software in RPM format.

Prerequisite:

To perform the installation from Red Hat Network, you must have a Red Hat Network account with a valid entitlement for JBoss Enterprise Application Platform.

6.2. INSTALL ON RED HAT ENTERPRISE LINUX 4

Procedure 6.1. Install on Red Hat Enterprise Linux 4

This procedure installs the latest version of JBoss Enterprise Application Platform 4.3 on a Red Hat Enterprise Linux 4 machine, or upgrades a previous version of JBoss Enterprise Application Platform to the latest version.

1. **Subscribe the system to the correct channel in the Red Hat Network.**

For instructions to subscribe a system to a channel refer to: "[How do I subscribe a system to a sub-channel or a child channel using Red Hat Network \(RHN\)?](#)" in the Red Hat Knowledgebase.

Red Hat Enterprise Linux 4 channel names

32-bit ES

`jbappplatform-4.3.0-i386-es-4-rpm`

`rhel-i386-es-4-extras`

32-bit AS

`jbappplatform-4.3.0-i386-as-4-rpm`

`rhel-i386-as-4-extras`

64-bit ES

`jbappplatform-4.3.0-x86_64-es-4-rpm`

`rhel-x86_64-es-4-extras`

64-bit AS

`jbappplatform-4.3.0-x86_64-as-4-rpm`

`rhel-x86_64-as-4-extras`

2. Install JBoss Enterprise Application Platform

Execute the following commands to install JBoss Enterprise Application Platform on Red Hat Enterprise Linux 4, where no previous version of the application server exists:

```
up2date jbossas jboss-seam2 jboss-seam rh-eap-docs jboss-profiler
```

3. Remove Obsolete GlassFish Dependencies

Before executing upgrade commands for JBoss Enterprise Application Platform, execute the following commands to remove obsolete GlassFish packages:

```
rpm -e classpathx-jaf
rpm -e --nodeps glassfish-jaf
up2date glassfish-jaf
```

4. Upgrade JBoss Enterprise Application Platform

Execute the following commands to upgrade from a previous version of JBoss Enterprise Application Platform:

```
up2date jbossas jboss-seam2 jboss-seam rh-eap-docs jboss-profiler
up2date -u
```

5. Perform post-installation configuration

Refer to [Chapter 7, Post Installation Configuration](#) for post-installation configuration instructions.

6.3. INSTALL ON RED HAT ENTERPRISE LINUX 5

Procedure 6.2. Install on Red Hat Enterprise Linux 5

This procedure installs the latest version of JBoss Enterprise Application Platform 4.3 on a Red Hat Enterprise Linux 5 machine, or upgrades a previous version of JBoss Enterprise Application Platform to the latest version.

1. Subscribe the system to the correct channel in the Red Hat Network.

For instructions to subscribe a system to a channel refer to: "[How do I subscribe a system to a sub-channel or a child channel using Red Hat Network \(RHN\)?](#)" in the Red Hat Knowledgebase.

Red Hat Enterprise Linux 5 channel names

32-bit

```
jbappplatform-4.3.0-i386-server-5-rpm
```

```
rhel-i386-server-supplementary-5
```

64-bit

```
jbappplatform-4.3.0-x86_64-server-5-rpm
```

```
rhel-x86_64-server-supplementary-5
```

2. Install JBoss Enterprise Application Platform

Available options are:

- o *CURRENT_REPO*: for 32-bit, use `rhel-i386-server-5`; for 64-bit, use `rhel-x86_64-server-5`

Execute the following commands to install JBoss Enterprise Application Platform on Red Hat Enterprise Linux 5, where no previous version of the application server exists. Run these commands with the chosen value for *CURRENT_REPO*.

```
yum remove classpathx-jaf
yum upgrade --disablerepo=CURRENT_REPO
yum install jbossas jboss-seam2 jboss-seam rh-eap-docs jboss-
profiler
```

3. Upgrade JBoss Enterprise Application Platform

Available options are:

- o *CURRENT_REPO*: for 32-bit, use `rhel-i386-server-5`; for 64-bit, use `rhel-x86_64-server-5`

Execute the following commands to upgrade JBoss Enterprise Application Platform on Red Hat Enterprise Linux 5, where a previous version of the application server exists.

```
yum remove classpathx-jaf
yum install jbossas jboss-seam2 jboss-seam rh-eap-docs jboss-
profiler
yum upgrade --disablerepo=CURRENT_REPO
```

4. Perform post-installation configuration

Refer to [Chapter 7, Post Installation Configuration](#) for post-installation configuration instructions.

CHAPTER 7. POST INSTALLATION CONFIGURATION

7.1. 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/vrenish/EnterprisePlatform-4.3.0/jboss-as is the
installation directory.
export JBOSS_HOME=/home/vrenish/EnterprisePlatform-4.3.0/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\EnterprisePlatform-4.3.0\jboss-as**. In order to run the server from the command line add the **bin** directory to your path, for example: **C:\Program Files\EnterprisePlatform-4.3.0\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.

7.2. 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.

On a Linux Platform

Memory settings can be adjusted on a Linux platform 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
```

On Microsoft Windows

To adjust the memory settings on Microsoft Windows, locate the **run.bat** file in the **bin** sub-directory of the **JBOSS_HOME** environment variable (**%JBOSS_HOME%\bin\run.bat**). Edit this file at the appropriate **set JAVA_OPTS** line corresponding to the comment "JVM memory allocation pool parameters". Modify this line according to the following recommended values:

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

7.3. POST INSTALLATION SECURITY CONFIGURATION

When installed from the zip archive, authentication is required to access the majority of JBoss services, including administrative services. Consoles are secured by the JAAS security domain "jmx-console". At installation this security domain has no user accounts. This is to eliminate the possibility

of default username/password based attacks. Refer to [Procedure 7.1, “Create jmx-console, admin-console, and http invoker user account”](#) to create a user account to access the consoles.

To *disable* authentication (useful for development, but not recommended for production), refer to [Section 7.6, “Disabling Authentication”](#).

When installed via the graphical installer, a JAAS security domain and a user account is created as part of the install process. Even if you change the name of the JAAS security domain during installation, the users are stored in the same place. Follow the instructions in [Procedure 7.1, “Create jmx-console, admin-console, and http invoker user account”](#) to edit your user account, or create a new one.

7.3.1. Security Configuration: JMX Console, Admin Console, HttpInvoker

Procedure 7.1. Create jmx-console, admin-console, and http invoker user account

This procedure creates user with access permissions to the admin and jmx consoles, and the http invoker

1. Create a user in the default JAAS security domain

- a. Edit the file `$JBOSS_HOME/server/$PROFILE/conf/props/jmx-console-users.properties`.
- b. Create a `username = password` pair.



IMPORTANT

The commented `admin=admin` username and password pair is an example of the username/password definition syntax. Do not use this for your user account.

2. Grant permissions to user

- a. Edit the file `$JBOSS_HOME/server/$PROFILE/conf/props/jmx-console-roles.properties`.
- b. Create an entry for the user of the form:

```
username=JBossAdmin,HttpInvoker
```

JBossAdmin

Grant the user permission to access the JMX Console and Admin Console.

HttpInvoker

Grant the user permission to access the httpinvoker



IMPORTANT

The authentication system applied to the JMX Console, Admin Console and Web Console does not block brute-force password attacks. It is recommended that in production environments, JBoss servers are protected by firewalls or reverse proxies that include measures to mitigate brute force attacks.

7.3.2. Securing the HTTPInvoker

The HTTP Invoker is a service that provides HTTP and Remote Method Invocation (RMI) access for EJBs and the JNDI Naming service. Secure this service to prevent unauthorized access.

Procedure 7.2. Secure the HTTP Invoker

1. Defining security constraints

The server/`$PROFILE/`deploy/http-invoker.sar/invoker.war/`WEB-INF/web.xml` or server/`$PROFILE/`deploy/httpa-invoker.sar/invoker.war/`WEB-INF/web.xml` file (depending on your server profile) must define a JNDIFactory, EJBInvokerServlet, and JMXInvokerServlet in the security realm. This means that the `security-constraint` element should be similar to:

```
<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>
    <url-pattern>/JNDIFactory/*</url-pattern>
    <url-pattern>/EJBInvokerServlet/*</url-pattern>
    <url-pattern>/JMXInvokerServlet/*</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>
```

2. Define an associated security domain

Add the following to fragment to `web.xml`:

```
<jboss-web>
  <security-domain>java:/jaas/jmx-console</security-domain>
</jboss-web>
```

3. Binding the jmx-invoker to localhost



NOTE

Binding the jmx-invoker to localhost is highly recommended for security, but makes it unavailable for use remotely.

Edit server/`$PROFILE/`conf/jboss-service.xml such that the `ServerAddress` of the RMI/JRMP invoker is `localhost`, as shown in the following code snippet:

```
<!-- RMI/JRMP invoker -->
<mbean code="org.jboss.invocation.jrmp.server.JRMPInvoker"
  name="jboss:service=invoker,type=jrmp">
```

```
<attribute name="RMIObjectPort">4444</attribute>
<attribute name="ServerAddress">localhost</attribute>
```

```
.....
```

4. Add the following lines to the **server** section of **server/\$PROFILE/deploy/jmx-invoker-service.xml**:

```
<!-- A pooled invoker bound to localhost -->
<mbean code="org.jboss.invocation.pooled.server.PooledInvoker"
      name="jboss:service=invoker,type=pooled,host=localhost">
  <attribute name="NumAcceptThreads">1</attribute>
  <attribute name="MaxPoolSize">300</attribute>
  <attribute name="ClientMaxPoolSize">300</attribute>
  <attribute name="SocketTimeout">60000</attribute>
  <attribute name="ServerBindAddress">localhost</attribute>
  <attribute name="ServerBindPort">4443</attribute>
  <attribute name="ClientConnectAddress">localhost</attribute>
  <attribute name="ClientConnectPort">0</attribute>
  <attribute name="ClientRetryCount">1</attribute>
  <attribute name="EnableTcpNoDelay">>false</attribute>
  <depends optional-attribute-
name="TransactionManagerService">jboss:service=TransactionManager</d
epends>
```

5. In the **<mbean code="org.jboss.invocation.jrmp.server.JRMPProxyFactory"** section, change **<depends optional-attribute-name="InvokerName">** to:

```
<depends optional-attribute-name="InvokerName">
jboss:service=invoker,type=pooled,host=localhost
</depends>
```

7.3.3. Security Configuration: 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 in the property **suckerPassword** in the configuration file:

Procedure 7.3. Set suckerPassword for JBoss Messaging:

This procedure sets the password used by JBoss Messaging in a clustered environment

1. Edit the file **jboss-as/server/\$PROFILE/deploy/messaging/messaging-jboss-beans.xml**.
2. Change the **suckerPassword** value.

7.4. 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>
```

7.5. RUN THE APPLICATION SERVER AS A SERVICE

Procedure 7.4. Run as a Service on Microsoft Windows

1. **Open a command prompt with elevated privileges.**
Navigate to `C:\Windows\System32` and right-click on `cmd.exe`. Select **Run as Administrator**.
2. **Change to the Enterprise Application Platform directory where the service installation script is located.**
`cd $JBOSS_DIST\native\sbin`

3. Optional: Edit `services.bat` to pass parameters to the Application Server at start-up.

Under `:cmdStart`, alter the following line:

```
call "%SVCPATH%\run.bat" < .r.lock >> run.log 2>&1
```

To run the 'default' profile binding to the 'localhost' address, change to the following: `call "%SVCPATH%\run.bat" -c default -b localhost < .r.lock >> run.log 2>&1`

For a full list of parameters to `run.bat` see the *Getting Started Guide*.

4. Run the service installation script.

```
service.bat install
```

5. Check that the service is installed.

Under the Windows services list you will find this listed by the short name `JBEAP5SVC` and the long name `JBoss EAP 5`.



NOTE

To uninstall the service, issue the following command from a command prompt with elevated privileges: `sc delete "JBEAP5SVC"`.

Procedure 7.5. Run as a Service on Red Hat Enterprise Linux

1. Add a JBoss User

With root privileges, use the `adduser` command with the `-r` parameter to create a system user account for use by the JBoss Enterprise Application Platform. Do this as the root user.

```
[localhost]$ su -
[localhost]# adduser -r jboss
```

2. Assign ownership of files

Use the `chown` and `chgrp` commands on the *installation directory* to change the **owner** and **group** of the JBoss Enterprise Application Platform files to the user created in the previous step.

```
[localhost]# chown -R jboss jboss-eap-5.1
[localhost]# chgrp -R jboss jboss-eap-5.1
```

3. Navigate to the `/bin` directory.

In a terminal, execute the following command to change into the directory containing the `jboss_init_*` scripts.

```
[home]$ cd $JBOSS_HOME/bin
[bin]$
```

4. Edit `jboss_init_redhat.sh`

The script `jboss_init_redhat.sh` in the `bin` directory is the script used to launch the server as a service. Append the following lines to the beginning of this script so that the first four lines below the initial comments look like those below.

```
#!/bin/sh
```



```
#chkconfig: 2345 85 15
#description: JBoss Enterprise Application Platform
#processname: jboss
```

These lines are needed by the `chkconfig` command.

Note that the `chkconfig` option specifies the runlevel, start priority, and stop priority. 2345 specifies the server will start only in runlevels 2, 3, 4, and 5.

5. Set values in `jboss_init_redhat.sh`

Edit `jboss_init_redhat.sh` in the `bin` directory so that the variables match the installation. The script variables are listed below.

JBOSS_HOME

This is the path of the JBoss Enterprise Application Platform's `jboss-as` directory. This value *must* be set here.

The example sets it to `/opt/jboss-eap-5.1/jboss-as`.

```
JBOSS_HOME=${JBOSS_HOME:-"/opt/jboss-eap-5.1/jboss-as"}
```

JBOSS_USER

This is the user created previously for running the JBoss Enterprise Application Platform.

The example sets it to the user name of `jboss`.

```
JBOSS_USER=${JBOSS_USER:-"jboss"}
```

JBOSS_CONF

This is the name of the server configuration that the server will be using.

The example sets it to `default`. The profiles available to use are contained in the `/jboss-as/server/` directory.

```
JBOSS_CONF=${JBOSS_CONF:-"default"}
```

JBOSS_HOST

JBOSS_HOST must be specified when binding the JBoss Enterprise Application Platform server to a specific IP address. This must be done before **JBOSS_HOST** is used by **JBOSS_BIND_ADDR**.

This must be configured to make the server available on the network. The default configuration binds the server to the IP address of `127.0.0.1`.

The example sets it to `10.1.1.83`

```
#if JBOSS_HOST specified, use -b to bind jboss services to that
address
JBOSS_HOST=10.1.1.83
JBOSS_BIND_ADDR=${JBOSS_HOST:+"-b $JBOSS_HOST"}
```

6. Link script into `init.d`

Create a symbolic link to `jboss_init_redhat.sh` in the directory `/etc/init.d/`. The name of the target of the symbolic link is the name of the new service.

The example uses the name `jboss_eap`.

```
[localhost]# ln -s /opt/jboss-eap-5.1/bin/jboss_init_redhat.sh
/etc/init.d/jboss_eap
```

7. Activate Service

Use the command `chkconfig` with the `--add` parameter to add the new service to the system configuration.

The example uses the service name of `jboss_eap`.

```
[localhost]# chkconfig --add jboss_eap
```

8. Configure Startup and Shutdown Behavior

Use the command `chkconfig` with the `on` parameter to start the service at boot time, and stop it gracefully when the server hosting the application server is shut down or restarted.

The example uses the service name of `jboss_eap`.

```
[localhost]# chkconfig jboss_eap on
```

9. Disable Service

Use the command `chkconfig` with the `off` parameter to disable the `jboss_eap` service from starting when the server starts.

```
[localhost]# chkconfig jboss_eap off
```

7.6. 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>
```



WARNING

Disabling authentication results in full administrator level access to the JBoss installation. A user connecting to a server with authentication disabled is permitted to run any code they choose on the server.

7.7. THE PRODUCTION CONFIGURATION AND CLUSTERING

The JBoss Enterprise Platform includes four server configurations which may be started by passing the `-c` parameter to the server startup script. A brief description of each configuration follows:

- the minimal configuration starts the core server container without any of the enterprise services. It is a good starting point if you want to build a customized version of JBoss AS that only contains the servers you need.
- the default configuration is the mostly common used configuration for application developers. It supports the standard J2EE 1.4 and most of the Java EE 5.0 programming APIs (e.g., JSF and EJB3);
- the all configuration supports clustering and other enterprise extensions;
- the production configuration is based on the all configuration with key parameters pre-tuned for production deployment.

The production configuration supports clustering by virtue of the fact that it is based on the all configuration. As such, it is important to understand how to separate clusters. This could be important, for instance, to separate test clusters from production clusters. Further details in regard to Cluster configurations can be found in the Server Configuration Guide located at http://www.redhat.com/docs/en-US/JBoss_Enterprise_Application_Platform.

CHAPTER 8. UNINSTALL JBOSS

If you used the GUI installer to install JBoss Enterprise Application Platform, then an automatic uninstaller is also installed. From the GUI the uninstaller can be selected from the JBoss program group, if one was created at installation time. A second option is to run this uninstaller from the command line. Within the JBoss Enterprise Application Platform Installation directory you will find a directory called **Uninstaller**. Inside the **Uninstaller** directory you will find a jar file named **uninstaller.jar**. Run the uninstaller from the command line using the `java` utility.

```
[vsr]$ java -jar uninstaller.jar
```

This command will launch the uninstaller program. If you wish to delete the installation directory and all its contents select the check box "Force the deletion of <installation_directory>" and hit the Uninstall button.

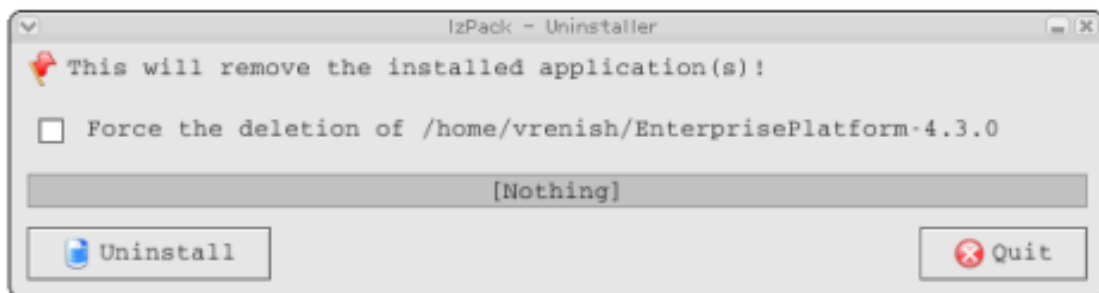


Figure 8.1. Uninstaller

If no uninstaller is available, and JBoss Enterprise Application Platform was installed using the zip file, it may be uninstalled by simply deleting the JBoss Enterprise Application Platform installed directory.

CHAPTER 9. TEST YOUR INSTALLATION

After you have installed the JBoss Enterprise Application 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:

```
[vrenish@vinux bin]$ ./run.sh
=====

JBoss Bootstrap Environment

JBOSS_HOME: /home/vrenish/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/vrenish/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` in your web browser. (Make sure you don't have anything else already on your machine using that port).^[1] The contents of your page should look similar to this:

Figure 9.1, “Test your Installation”.

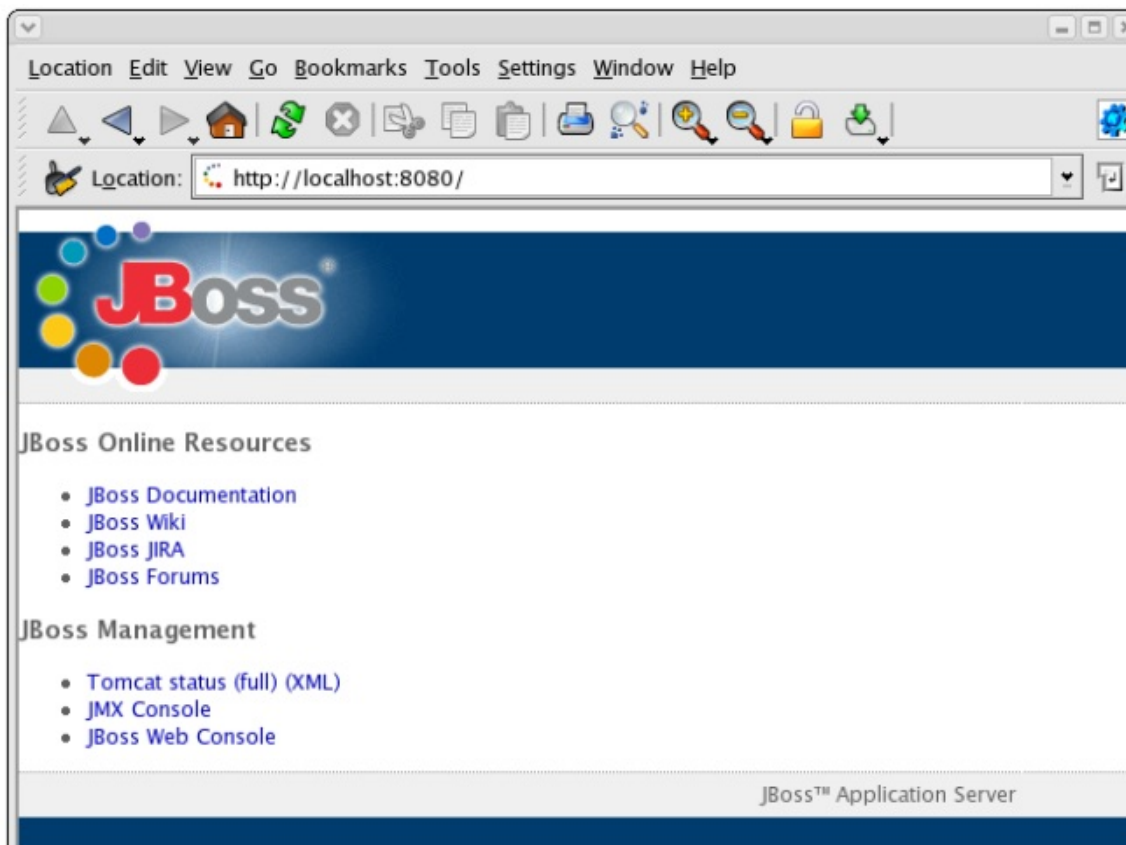


Figure 9.1. Test your Installation

You are now ready to use the JBoss Enterprise Application Platform. Refer to the Getting Started Guide for more information about the platform layout and example applications showcasing JBoss 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.

APPENDIX A. INSTALLING A JAVA DEVELOPMENT KIT ON RED HAT ENTERPRISE LINUX

Red Hat supports the JBoss Enterprise Application Platform when it is run on Red Hat Enterprise Linux version 4 or 5 in conjunction with the Sun Microsystems Java Development Kit (JDK) version 1.6, and the IBM JDK version 1.5.



NOTE

If you have difficulties subscribing to the correct software channels in Red Hat Network you should refer to the Red Hat Network Help Desk at <https://rhn.redhat.com/rhn/help/> or contact Red Hat Support via <http://access.redhat.com> directly for assistance.

A.1. OPENJDK ON RED HAT ENTERPRISE LINUX 5

Use this procedure to install OpenJDK on Red Hat Enterprise Linux 5.



IMPORTANT

The following commands must be run as **root**.

Procedure A.1. Installing OpenJDK on Red Hat Enterprise Linux 5

1. **Subscribe to the base channel.**

The OpenJDK is available in Red Hat Enterprise Linux's base channel.

2. **Install the package.**

To install OpenJDK, issue the following command:

```
yum install java-1.6.0-openjdk-devel
```

3. **Set OpenJDK as the system's default Java Development Kit.**

To ensure that the correct JDK is set as the system default, run the `alternatives` command as described in [Section A.4, "Setting the default JDK with the /usr/sbin/alternatives Utility"](#)

A.2. IBM JAVA DEVELOPMENT KIT ON RED HAT ENTERPRISE LINUX 5

Use this procedure to install the IBM Java Development Kit (JDK) on Red Hat Enterprise Linux 5.



IMPORTANT

The IBM JDK is a dependency of the platform installation. You must install this package for installation to succeed. You do not have to set the IBM JDK as the primary JDK the platform uses.

Procedure A.2. Installing the Sun Microsystems JDK on Red Hat Enterprise Linux 5

1. **Subscribe to Supplementary Server channel.**

The IBM Java Development Kit is available in the Supplementary Server channel.

2. Install the package.

To install the **Sun Microsystems Java Development Kit** package, become the root user and run this command:

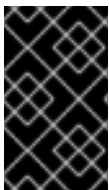
```
yum install java-1.5.0-ibm-devel
```

3. Set OpenJDK as the system's default Java Development Kit

To ensure that the intended JDK is set as the system default, run the `alternatives` command as described in [Section A.4, “Setting the default JDK with the `/usr/sbin/alternatives` Utility”](#)

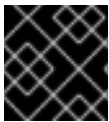
A.3. IBM JDK ON RED HAT ENTERPRISE LINUX AS/ES 4

Use this procedure to install the **IBM Java Development Kit** on **Red Hat Enterprise Linux AS** or **ES 4**.



IMPORTANT

The IBM JDK is a dependency of the platform installation. You must install this package for installation to succeed. You do not have to set the IBM JDK as the primary JDK the platform uses.



IMPORTANT

The following commands must be run as **root**.

Procedure A.3. Installing the IBM JDK on Red Hat Enterprise Linux AS/ES 4

1. Subscribe to the Extras channel.

The **IBM Java Development Kit** is available in the **Red Hat Extras** channel. Ensure that the machine is subscribed to this channel in order to install this package.

2. Install using the `up2date` command.

Run this command to install the package:

```
up2date java-1.5.0-ibm-devel
```

3. Set OpenJDK to the system's default Java Development Kit.

To ensure that the intended JDK is set as the system default, run the `alternatives` command as described in [Section A.4, “Setting the default JDK with the `/usr/sbin/alternatives` Utility”](#).

A.4. SETTING THE DEFAULT JDK WITH THE `/usr/sbin/alternatives` UTILITY

`/usr/sbin/alternatives` is a tool for managing different software packages that provide the same functionality. **Red Hat Enterprise Linux** uses `/usr/sbin/alternatives` to ensure that only one Java Development Kit is set as the system default at one time.



IMPORTANT

Installing a Java Development Kit from the Red Hat Network will normally result in an automatically configured system. However, if multiple JDKs are installed, it is possible that `/usr/sbin/alternatives` may contain conflicting configurations. Refer to [Procedure A.4, “Using `/usr/sbin/alternatives` to Set the Default JDK”](#) for syntax of the `/usr/sbin/alternatives` command.

Procedure A.4. Using `/usr/sbin/alternatives` to Set the Default JDK

1. **Become the root user.**

`/usr/sbin/alternatives` needs to be run with root privileges. Use the `su` command or other mechanism to gain these privileges.

2. **Set java.**

Input this command: `/usr/sbin/alternatives --config java`

Next, follow the on-screen directions to ensure that the correct version of `java` is selected.

[Table A.1, “java alternative commands”](#) shows the relevant command settings for each of the different JDKs.

Table A.1. java alternative commands

JDK	alternative command
OpenJDK 1.6	<code>/usr/lib/jvm/jre-1.6.0-openjdk/bin/java</code>
Sun Microsystems JDK 1.6	<code>/usr/lib/jvm/jre-1.6.0-sun/bin/java</code>

3. **Set javac.**

Enter this command: `/usr/sbin/alternatives --config javac`

Follow the on-screen directions to ensure that the correct version of `javac` is selected.

[Table A.2, “javac alternative commands”](#) shows the appropriate command settings for the different JDKs.

Table A.2. javac alternative commands

JDK	alternative command
OpenJDK 1.6	<code>/usr/lib/jvm/java-1.6.0-openjdk/bin/javac</code>
Sun Microsystems JDK 1.6	<code>/usr/lib/jvm/java-1.6.0-sun/bin/javac</code>

4. **Extra Step: Set `java_sdk_1.6.0`.**

The Sun Microsystems JDK 1.6 requires an additional command be run:

`/usr/sbin/alternatives --config java_sdk_1.6.0`

Follow the on-screen directions to ensure that the correct `java_sdk` is selected. It is `/usr/lib/jvm/java-1.6.0-sun`.

APPENDIX B. REVISION HISTORY

Revision 4.3.10-100.33.400 Rebuild with publican 4.0.0	2013-10-30	Rüdiger Landmann
Revision 4.3.10-100.33 Rebuild for Publican 3.0	July 24 2012	Ruediger Landmann
Revision 4.3.10-100 Incorporated changes for the Enterprise Application Platform 4.3.0CP10 release. For more information, refer to the Documentation Resolved Issues in the <i>Release Notes CP10</i> .	Mon Aug 29 2011	Jared Morgan
Revision 4.3.9-100 Incorporated changes for the Enterprise Application Platform 4.3.0CP09 release. For more information, refer to the Documentation Resolved Issues in the <i>Release Notes CP09</i> .	Tue Nov 30 2010	Jared Morgan