



JBoss Enterprise Web Platform 5 Installation Guide

for use with JBoss Enterprise Web Platform 5
Edition 5.1.1

Isaac Rooskov
Elspeth Thorne

Laura Bailey

Joshua Wulf

JBoss Enterprise Web Platform 5 Installation Guide

for use with JBoss Enterprise Web Platform 5
Edition 5.1.1

Isaac Rooskov

Laura Bailey

Joshua Wulf

Elsbeth Thorne

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, MetaMatrix, 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 Web Platform 5 and its patch releases.

Table of Contents

Preface	3
1. Document Conventions	3
1.1. Typographic Conventions	3
1.2. Pull-quote Conventions	4
1.3. Notes and Warnings	5
2. Getting Help and Giving Feedback	5
2.1. Do You Need Help?	5
2.2. We Need Feedback!	6
Chapter 1. Introduction	7
1.1. Feedback	7
1.2. Other Manuals	7
Chapter 2. Migrating to Enterprise Application Platform 5	8
2.1. What's New in Enterprise Application Platform 5	8
2.1.1. JBoss Application Server 5.1.0.GA	8
2.1.1.1. ProfileService-based Deployment Configuration	8
2.1.2. Enterprise Java Beans (EJB) 3.0	8
2.1.3. Java Enterprise Edition 5 Compliance	9
2.1.4. Seam 2.2.0.GA	9
2.1.5. RESTEasy 1.1.GA	9
2.1.6. Enhanced Enterprise GUI Installer	9
2.1.7. Enterprise Application Platform Admin Console	9
2.1.8. JBoss Transactions includes Java Transaction Service	9
2.1.9. Distribution with Red Hat Signed JARs	9
2.2. What's Different in Enterprise Application Server 5	10
2.2.1. Differences in the Distribution Layout	10
2.2.2. Standard and Web Configuration	12
2.2.3. Differences in Application Server Configuration Files	13
2.2.3.1. General	13
2.2.3.2. J2EE Connector Architecture	13
2.2.3.3. Web	14
2.2.3.4. Clustering	15
2.2.3.5. Transactions	15
2.2.3.6. Logging	15
2.2.3.7. Security	16
2.2.3.8. Enterprise JavaBeans	16
2.3. Admin Console	16
2.4. Applications	17
2.4.1. Classloading	18
2.4.2. EAR Scoping	19
Chapter 3. New Installation	20
3.1. Pre-Requisites	20
3.1.1. Hardware, Operating System, and JVM Requirements	20
3.1.2. Configuring Your Java Environment	20
Chapter 4. Installation Options	21
4.1. Web Services Stack	21
4.2. PicketLink Federation	21
4.3. Installation Methods	21
Chapter 5. ZIP Installation from the Red Hat Customer Portal	23

5.1. HornetQ	23
Chapter 6. RPM Installation via Red Hat Network	25
6.1. Red Hat Network	25
6.2. Install on Red Hat Enterprise Linux 4	25
6.3. Install on Red Hat Enterprise Linux 5	26
6.4. Install on Red Hat Enterprise Linux 6	27
Chapter 7. Installation using the Graphical Installer	28
Chapter 8. Install Native Components	30
8.1. Red Hat Enterprise Linux-specific notes	30
8.2. Solaris-specific notes	31
8.3. Native Components Installation	31
Chapter 9. Post Installation Configuration	33
9.1. Post Installation Security Configuration	33
9.1.1. Security Configuration: JMX Console, Admin Console, HttpInvoker	33
9.1.2. Securing the HTTPInvoker	34
9.1.3. Security Configuration: Web Console	35
9.1.4. Security Configuration: JBoss Messaging	36
9.2. Default Database	37
9.3. Memory Settings for the Enterprise Web Platform	37
9.4. Run the Application Server as a Service	38
Chapter 10. Test your Installation	40
Chapter 11. Uninstall JBoss Enterprise Web Platform	41
Disabling Authentication	42
The Red Hat Customer Portal	45
Installing a Java Development Kit on Red Hat Enterprise Linux	46
C.1. OpenJDK on Red Hat Enterprise Linux 5	46
C.2. Sun Java Development Kit on Red Hat Enterprise Linux 5	46
C.3. Sun JDK on Red Hat Enterprise Linux AS/ES 4	47
C.4. Setting the default JDK with the /usr/sbin/alternatives Utility	48
Installing the Sun JDK on Microsoft Windows	50
Installing Apache Ant	51
Revision History	53

Preface

1. Document Conventions

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

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 keys 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, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from an individual key by the plus sign that connects each part of a key combination. For example:

Press **Enter** to execute the command.

Press **Ctrl+Alt+F2** to switch to a virtual terminal.

The first example highlights a particular key to press. The second example highlights a key combination: a set of three keys 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; dialog-box text; labeled buttons; check-box and radio-button labels; menu titles and submenu titles. For example:

Choose **System** → **Preferences** → **Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, select 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.

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:

Publican is a *DocBook* publishing system.

1.2. Pull-quote Conventions

Terminal output and source code listings 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 add syntax highlighting as follows:

```
static int kvm_vm_ioctl_deassign_device(struct kvm *kvm,
                                       struct kvm_assigned_pci_dev *assigned_dev)
{
    int r = 0;
    struct kvm_assigned_dev_kernel *match;

    mutex_lock(&kvm->lock);

    match = kvm_find_assigned_dev(&kvm->arch.assigned_dev_head,
                                assigned_dev->assigned_dev_id);
    if (!match) {
        printk(KERN_INFO "%s: device hasn't been assigned
```

```

before, "
            "so cannot be deassigned\n", __func__);
    r = -EINVAL;
    goto out;
}

kvm_deassign_device(kvm, match);

kvm_free_assigned_device(kvm, match);

out:
    mutex_unlock(&kvm->lock);
    return r;
}

```

1.3. Notes and Warnings

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



Note

Notes are tips, shortcuts or alternative approaches 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 a box labeled “Important” will not cause data loss but may cause irritation and frustration.



Warning

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

2. Getting Help and Giving Feedback

2.1. Do You Need Help?

If you experience difficulty with a procedure described in this documentation, visit the Red Hat Customer Portal at <http://access.redhat.com>. Through the customer portal, you can:

- ✧ search or browse through a knowledgebase of technical support articles about Red Hat products.
- ✧ submit a support case to Red Hat Global Support Services (GSS).
- ✧ access other product documentation.

Red Hat also hosts a large number of electronic mailing lists for discussion of Red Hat software and technology. You can find a list of publicly available mailing lists at <https://www.redhat.com/mailman/listinfo>. Click on the name of any mailing list to subscribe to that list or to access the list archives.

2.2. 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! Please submit a report in Bugzilla: <http://bugzilla.redhat.com/> against the product **JBoss Enterprise Application Platform 5**.

When submitting a bug report, be sure to mention the manual's identifier: *Installation 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.

Chapter 1. Introduction

JBoss Enterprise Web 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. 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 Web 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 Web 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. Feedback

If you spot a typographical error in this guide, 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](#) against the Product: JBoss Enterprise Application Platform, Version: *EWP 5.1.0*, Component: *Documentation*. If you have a suggestion for improving the documentation, try to be as specific as possible. If you have found an error, include the section number and some of the surrounding text so we can find it easily.

1.2. Other Manuals

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

Chapter 2. Migrating to Enterprise Application Platform 5

This chapter provides information for administrators who plan to move their enterprise servers from JBoss Enterprise Application Platform 4.2 or 4.3 to the new Enterprise Application Platform 5. The first section covers new features available in Enterprise Application Platform 5. The second section covers the changes to configuration, administration, and application deployment between Enterprise Application Platform 4.x and Enterprise Application Platform 5. If you require further information, refer to the relevant guides provided in this release.

2.1. What's New in Enterprise Application Platform 5

This section provides an overview of the components of Enterprise Application Platform 5, and the changes to each component between version 4.x and 5.

2.1.1. JBoss Application Server 5.1.0.GA

JBoss Application Server 5 is the next generation of the JBoss Application Server built on top of a new kernel architecture, the **JBoss Microcontainer**. The JBoss Microcontainer is a lightweight container for managing the deployment, configuration and lifecycle of Plain Old Java Objects (POJOs). While remaining compatible with the 4.x-based JMX kernel, the Microcontainer integrates with the JBoss framework for Aspect Oriented Programming, JBoss AOP. JMX support remains strong in JBoss AS 5, and MBean services written against the old Microkernel work as expected. Further, it lays the groundwork for Java EE 6 profile-oriented configurations and embedded JBoss AS, which will allow for fine grained selection of services for both unit testing and embedded scenarios.

2.1.1.1. ProfileService-based Deployment Configuration

Definitions for both non-kernel deployers and their deployment are now contained in a Profile obtained from the ProfileService. The **ProfileService** replaces JBoss AS 4.x *server configuration*. In JBoss AS 4.x, a server configuration was a collection of services and applications loaded from the **deploy** directory by the deployment scanner service. Enterprise Application Platform 5 uses more active profiles, which may depend on other *sub-profiles*.

The main profile is the *server profile*, which is based on the `${jboss.server.name}`. This profile has three sub-profiles:

- *bootstrap* — representing `conf/jboss-service.xml`
- *deployers* — the **deployers/** directory
- *applications* — a hot-deployment profile for the **deploy/** and additional user directories

A profile generally represents a named collection of deployments on a server. A profile can also apply certain behaviors to the deployments that it manages. Some profiles, such as the **application** profile, provide hot-deployment checks and allow remote distribution of deployed applications via the **DeploymentManager**. Other profiles can provide a farming service to distribute deployments over a cluster. The ProfileService also provides the ManagementView for ManagedDeployments/ManagedObjects used by the Enterprise Application Admin Console (admin-console).

2.1.2. Enterprise Java Beans (EJB) 3.0

JBoss EJB 3.0, an implementation of the latest revision of the EJB specification, is a deep overhaul and simplification of earlier versions of the EJB specification. It simplifies development, facilitates a test driven approach, and focuses more on writing POJOs rather than coding against complex EJB

APIs.

2.1.3. Java Enterprise Edition 5 Compliance

JBoss Enterprise Web Platform 5 is a fully-certified Java EE 5 implementation. It uses the microcontainer to integrate enterprise services with a Servlet/JSP container, EJB container, deployers and management utilities, providing a standard Java EE environment with the flexibility to deploy additional services on top of Java EE to give you the functionality you need. For further compatibility details, read <http://java.sun.com/javaee/overview/compatibility.jsp> page.

2.1.4. Seam 2.2.0.GA

Seam is an application framework for Java Enterprise Edition. It integrates technologies such as Asynchronous JavaScript and XML (AJAX), JavaServer Faces (JSF), Java Persistence (JPA), Enterprise JavaBeans 3.0 (EJB) and Business Process Management (BPM). Seam enables developers to assemble complex web applications using simple annotated Java classes, a rich set of UI components, and very little XML.

2.1.5. RESTEasy 1.1.GA

RESTEasy provides several frameworks to help you build RESTful Web Services and RESTful Java applications. It is a fully-certified, portable implementation of the [JAX-RS](#) specification, which defines a Java API for RESTful Web Services over the Hypertext Transfer Protocol (HTTP).

2.1.6. Enhanced Enterprise GUI Installer

The Enterprise Installer retains the familiar Enterprise Application Platform 4.3 interface but includes enhancements to provide you with a complete Enterprise Application Platform 5 installation. The installer is localized and provides you with secure JMX, Web and Admin Consoles.

The new Enterprise Installer also presents users with the opportunity to install the optional Native package, which includes **JBoss Native** and **mod_jk**. The Native package helps users who wish to use Tomcat or JBoss Web with the HTTP daemon.

2.1.7. Enterprise Application Platform Admin Console

A new Admin Console is being introduced in this Enterprise Application Platform release. The admin-console enables configuration and management of a single Enterprise Application Platform server instance. See [Section 2.3, “Admin Console”](#) for more information about this new management console.

2.1.8. JBoss Transactions includes Java Transaction Service

JBoss Transactions now includes the Java Transaction Service and the XML Transaction Service. The Java Transaction Service handles distributed, interoperable transactions between Enterprise JavaBean containers. The XML Transaction Service handles transactions for Web Services.

2.1.9. Distribution with Red Hat Signed JARs

JAR files included with JBoss Enterprise Web Platform are digitally signed by Red Hat. This gives you an additional level of security about the source and identity of the code executing on your systems.

For the complete technology matrix and information on the revision level of included components please refer to the Release Notes.

2.2. What's Different in Enterprise Application Server 5

The distribution layout and configuration information in the Enterprise Application Platform 5 distribution are similar to the Enterprise Application Platform 4.x series with some notable differences. This section highlights the differences at a glance.

2.2.1. Differences in the Distribution Layout

The directory structure of **jboss-as** directory is summarized below.

- ✧ **/bin** — contains start scripts and **run.jar**
- ✧ **/client** — contains client JARs.



Note

Previously, JBoss client libraries were bundled in **jbossall-client.jar**. Rather than including them, **jbossall-client.jar** now references them through a Classpath manifest entry. This enables granular updating of libraries without requiring replacement of all libraries. It requires that you have the **jbossall-client.jar**, which now acts as a map or index, as well as the actual **client/*.jar** libraries.

- ✧ **/common/lib** — contains shared libraries common to various configurations have been moved to this new shared location. This eliminates the need for multiple copies of the same library in the distribution.

The location of the common library directory is controlled with the following properties:

- **jboss.common.base.url** — the default value is **\${jboss.home.url}/common**
- **jboss.common.lib.url** — the default value is **\${jboss.common.base.url}/lib**

You can set these properties in **run.conf** under **JAVA_OPTS** with the **-D** flag:

```
JAVA_OPTS="[...] -Djboss.common.base.url=$URL1 -
Djboss.common.lib.url=$URL2"
```

The common library directory is shared by all configuration types except for the **minimal** configuration. The common library is referenced at the beginning of every configuration's **conf/jboss-service.xml**

```
<classpath codebase="${jboss.server.lib.url}" archives="*" />
```

The **library** directory of the individual directory remains in place, although in some cases (as in **\$JBOSS_HOME/server/default/lib/**) it is an empty directory.

- ✧ **/docs** — contains schemas, document type declarations, examples and licenses. Most deployment descriptors now use XML Schema Definitions (XSDs). One exception is **jboss-app**, which uses **jboss-app_5_0.dtd**. JBoss Web uses **jboss-web_5_1.xsd**. For Enterprise JavaBeans 3.0 deployments, **jboss_5_1.xsd** is the recommended schema. Enterprise JavaBeans 2.0 deployments must use **jboss_x_x.dtd**.
- ✧ **/lib** — contains the core bootstrap JARs. These have been changed slightly to accommodate the Microcontainer and the division of **jboss-common**.

- ✧ **/server** — contains directories for configuring the server:
 - **\$PROFILE** — contains the configuration details of a particular server profile
 - **/conf**
 - **bootstrap.xml** — a new kernel bootstrap configuration that refers to other configuration files containing the beans to set up each individual subsystem.
 - **bindingservice.beans**
 - **/META-INF**
 - **bindings-jboss-beans.xml** — contains required port bindings.
 - **jboss-bindingservice.jar**
 - **/bootstrap**
 - **vfs.xml** — initializes the virtual file system
 - **classloader.xml**
 - **aop.xml**
 - **jmx.xml** — legacy JMX support.
 - **deployers.xml**
 - **profile-repository.xml** — the ProfileService enabled deployment repository.
 - **jax-ws-catalog.xml** — an Oasis Catalog-driven Schema/DTD namespace configuration file.
 - **jbossts-properties.xml** — contains new JBossTS properties.
 - **jboss-service.xml** — contains legacy static managed beans to retain compatibility.
 - **jndi.properties** — contains JNDI configuration properties.
 - **log4j.xml** — contains log4j configuration information.
 - **login-config.xml** — contains JAAS login configuration information.
 - **/props** — contains default JAAS login properties files.
 - **standardjbosscmp-jdbc.xml** — contains CMP2 configuration information.
 - **standardjboss.xml** — contains Enterprise JavaBean 2.0 configuration information.
 - **/xmdesc** — contains legacy XML managed bean descriptors.
 - **/deploy**
 - **jca-jboss-beans.xml**
 - **hdscanner-jboss-beans.xml** — contains the hot-deployment scanner.
 - **legacy-invokers-service.xml**
 - **profiles-service-jboss-beans.xml**
 - **remoting-jboss-beans.xml**

- **transaction-jboss-beans.xml**
- **vfs-jboss-beans.xml**
- **/deployers** — contains new VDF deployers.
 - **/bsh-deployer** — contains the beanshell deployer.
 - **ejb3.deployer** — contains Enterprise JavaBean 3.0 deployers.
 - **jboss-aop-jboss5.deployer** — contains the aspect deployer.
 - **jboss-jca.deployer** — contains the JCA deployers.
 - **jbossweb.deployer** — contains the WAR deployers.
 - **jbossws.deployer** — contains the web service deployers.
 - **seam.deployer** — contains the Seam deployer.
 - **clustering-deployers-jboss-beans.xml**
 - **dependency-deployers-jboss-beans.xml**
 - **directory-deployer-jboss-beans.xml**
 - **ear-deployer-jboss-beans.xml**
 - **ejb-deployer-jboss-beans.xml**
 - **hibernate-deployer-jboss-beans.xml**
 - **logbridge-boss-beans.xml**
 - **jsr77-deployers-jboss-beans.xml** — contains JSR-77 (J2EE Management) support.
 - **metadata-deployer-jboss-beans.xml** — contains the metadata handlers.
 - **messaging-definitions-jboss-beans.xml** — contains data required to map JMS destinations to managed objects.
 - **security-deployer-jboss-beans.xml** — contains the security deployers.
 - **xnio.deployer**
 - **jboss-threads.deployer**
- **/lib** — contains static library JARs. Some JARs that were previously located in this directory have been moved into the top-level **common/lib** directory.

2.2.2. Standard and Web Configuration

Two additional server configurations are distributed with Enterprise Application Platform 5: **standard** and **web**.

The **standard** configuration is certified for Java EE 5 compliance. This configuration enables both call-by-value and deployment isolation by default. Support for RMI-IIOP (Remote Method Invocation over the Internet Inter-Orb Protocol) and Java UDDI (Universal Description, Discovery and Integration), as in the **all** configuration type, is also enabled.

The **web** configuration is lightweight. It was created around JBoss Web and provides the services required for web application deployment and only a subset of Java EE technologies. This profile does not include JBoss Transaction JTS or XTS, Enterprise Java Bean 1.x or 2.x capabilities, JBoss Messaging, JCA, or JBoss IIOP.

2.2.3. Differences in Application Server Configuration Files

2.2.3.1. General

- ✧ A reminder that the RPM and ZIP distributions of the Enterprise Application Platform are shipped with authentication enabled for the JMX Console, Web Console, JMX Invoker, Admin Console, HTTP Invoker and Profile Service. No user accounts are active by default to assist in preventing default user and password-based attacks.

- ✧ **shutdown.sh** now accepts a JNDI URL, as follows:

```
shutdown.sh -s http://localhost:8080/invoker/JNDIFactory -S
```

Where **-s** defines the server name to perform an operation on; **-S** specifies the shutdown operation.

- ✧ If a user omits the **-c** option when starting an instance of JBoss Application Server in Enterprise Application Platform 4.x, the **production** configuration was started by default. In JBoss Enterprise Web Platform 5, **default** configuration is used when a user omits the **-c** option.
- ✧ **bin/run.conf** now uses a Java heap size of 1303 MB. This is consistent across all configurations.
- ✧ Document Type and Schema Declarations have been updated.
- ✧ The **production** server profile provided with Enterprise Application Platform 5 restricts the classes served on port 8083. If Remote Method Invocation (RMI) is being used, you may need to make this port available to clients. This option can be set in **production/conf/jboss-service.xml**:

```
<!-- Should non-EJB .class files be downloadable -->
<attribute name="DownloadServerClasses">false</attribute>
```

- ✧ The cluster-safe UUID generator can now be used from **server/production/deploy/uuid-key-generator.sar/META-INF/jboss-service.xml**.
- ✧ The delay period for **server/production/deploy/hdscanner-jboss-beans.xml** to rescan for deployment changes has been increased to 60 seconds from the previous 5 second delay period.

```
<!-- Frequency in milliseconds to rescan the URLs for changes-->
<property name="scanPeriod">60000</property>
```

2.2.3.2. J2EE Connector Architecture

- ✧ **jboss-ra.xml** can now be used to override the properties specified in ***-ra.xml**.

The **jboss-ra.xml** file should be in the **META-INF** directory of the resource adapter whose properties you wish to override, alongside the ***-ra.xml** file.

Specify a corresponding **<ra-config-property>** in the **jboss-ra.xml** file for each property

you wish to override. An example follows:

Example 2.1. Representative excerpt from resource adapter *-ra.xml file

```
<config-property>
  <config-property-name>StringRAR</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value>StringFromRARProperties</config-property-
value>
</config-property>
```

Example 2.2. Representative excerpt from a corresponding jboss-ra.xml file

```
<ra-config-property>
  <ra-config-property-name>StringRAR</ra-config-property-name>
  <ra-config-property-type>java.lang.String</ra-config-property-
type>
  <ra-config-property-value>XMLEVERRIDE</ra-config-property-value>
</ra-config-property>
```

The complete source for a working example can be viewed in the test case for this feature at <https://anonsvn.jboss.org/repos/jbossas/trunk/testsuite/src/resources/jcaprops/xmloverride/META-INF/>.

- ✦ Support has been added for defining dependencies in J2EE Connector Architecture (JCA) adapters.
- ✦ **server/production/deploy/jca-jboss-beans.xml** disables debug monitoring of JCA and database connections:

```
<!-- Whether to track unclosed connections and close them -->
<property name="debug">false</property>
```

This disables the application server's debug support. Disabling this means that the origin of obtained database connections and connection leaks cannot be tracked. Unclosed managed database connections are still returned to the connection pool, regardless of this attribute's value.

2.2.3.3. Web

- ✦ For JavaServer Pages-based pages, the default setting for **DeleteWorkDirOnContextDestroy** is **false**. Set this to **true** to enable a faster, simpler page recompilation check, or if you are using JSP settings that require recompilation.
- ✦ **emptySessionPath="true"** no longer sets the cookie path / by default. Instead, the cookie path is set via the **<SessionCookie path="/" />** in the **Context** element. Session cookies are now scoped to the context by default.
- ✦ **emptySessionPath** no longer affects whether Session IDs are recycled. This is now handled by the **org.apache.catalina.connector.Request.SESSION_ID_CHECK** system property. If set to **true**, the Servlet container verifies that a Session ID does not yet exist in a particular context before creating a session with that ID. You can set this property in the **jboss-as/bin/run.conf** file using the **-D** switch.

2.2.3.4. Clustering

- ✦ Clustering configurations have been moved to a new **/deploy/cluster** directory.

```
cluster
|-- deploy-hasingleton-jboss-beans.xml
|-- farm-deployment-jboss-beans.xml
|-- ha-legacy-jboss-beans.xml
|-- hajndi-jboss-beans.xml
|-- hapartition-jboss-beans.xml
|-- jboss-cache-manager.sar
| `-- META-INF
|   |-- jboss-cache-configs.xml
|   `-- jboss-cache-manager-jboss-beans.xml
|-- jbossweb-cluster.aop
|-- jgroups-channelfactory.sar
| `-- META-INF
|   |-- jgroups-channelfactory-jboss-beans.xml
|   `-- jgroups-channelfactory-stacks.xml
`-- timestamps-jboss-beans.xml
```

- ✦ A separate cache is now used for Clustered Single Sign-On (SSO).
- ✦ UseJK, snapshot mode and snapshot interval can now be configured on a per-application basis. The default value for UseJK depends upon whether the **jvmRoute** is set.
- ✦ The default setting for session replication is now **total** replication instead of **buddy** replication.
- ✦ **loopback** is now set to **true** for all JGroups User Datagram Protocol stacks.
- ✦ The **jboss.jgroups.udp.mcast_port** property is now used to configure the multicast port. The **-m** option to the **run.sh** or **run.bat** script now sets **jboss.jgroups.udp.mcast_port** instead of **jgroups.udp.mcast_port**.

jgroups.udp.mcast_port is checked internally by JGroups, and is used to override any XML-based configuration. If this parameter is set, two channels with non-shared transports cannot use different ports. The **jboss.jgroups.udp.mcast_port** property substitutes system properties in the default UDP channel configurations.

2.2.3.5. Transactions

The transaction manager configuration information has moved from **conf/jboss-service.xml** to **deploy/transaction-service.xml**.

2.2.3.6. Logging

- ✦ The default **conf/jboss-log4j.xml** configuration now includes the thread name for **log/server.log** entries.
- ✦ The new **jboss.server.log.threshold** system property can be used to control the **log/server.log** threshold. The default value is **INFO**.
- ✦ **server.log** is appended, rather than truncated, after a server is restarted.
- ✦ The following changes apply only to **server/production/conf/jboss-log4j.xml**:
 - the console logger has been commented out by default.

- the async logger is enabled by default.
- a **cluster.log** file has been added to store cluster output.

2.2.3.7. Security

Security-related configuration files are now found in the **deploy/security** directory:

```
security/  
|-- security-jboss-beans.xml  
`-- security-policies-jboss-beans.xml
```

2.2.3.8. Enterprise JavaBeans

- ✧ Enterprise JavaBean configuration information is now located in **deployers/ejb3.deployer/META-INF/ejb3-deployers-jboss-beans.xml**.
- ✧ Java Persistence API configuration information is now located in **deployers/ejb3.deployer/META-INF/jpa-deployers-jboss-beans.xml**.

2.3. Admin Console

The first release of the JBoss Enterprise Web Platform Admin Console (**admin-console**) provides the following administrative features:

- ✧ configuration information about the system on which the Enterprise Application Platform is running.
- ✧ configuration information about the Service Binding Manager.
- ✧ deploy, undeploy and update Enterprise Applications, including:
 - Java EE Enterprise Applications (EARs)
 - Web Applications (WARs)
 - Resource Adapters (RARs)
 - Enterprise JavaBean 2 and 3 (JARs)
- ✧ persistent configuration changes for the following resources:
 - data sources
 - connection factories
 - JMS queues and topics (based on JBoss Messaging)
- ✧ Control Operations:
 - execute scripts to perform tasks against a running instance of the application server
 - stop, start, and restart applications
 - view resource statistics
 - view resource metric information

The new **admin-console** provided with JBoss Enterprise Web Platform retains the JMX and web consoles. **admin-console** supports the **production**, **all**, **web** and **default** configurations out of the box. It has also been tested with **standard** server profile, but is not included in standard by default. To use **admin-console** in a **standard** profile, copy the **admin-console.war** from one of the supported server profiles.



Note

The Admin Console is not intended for use with the **minimal** configuration provided with the distribution. Custom configurations based on this configuration should not be used with the Admin Console, either.

When the server has been started, you can use the **admin-console** to perform administrative tasks for your application server. To use the **admin-console**, navigate to **http://\${hostname}:8080/admin-console**.

Refer to the *Administration Console User Guide* for more information on the Admin Console.

2.4. Applications

JBoss Enterprise Web Platform 5 is a fully-compliant implementation of the Java Enterprise Edition 5 (Java EE 5) Platform Specification. Java EE 5 defines the metadata associations of the Java language which can be used to annotate application code and eliminate the need for deployment descriptors wherever possible. Default behavior is also defined with the ability to override as needed. This is known as *configuration by exception*.

Portable Java EE applications running on Enterprise Application Platform 4.x can be deployed to Enterprise Application Platform 5 without any changes. However, runtime-specific deployment information may be required when migrating from another vendor's application server to JBoss Enterprise Web Platform 5.

Enterprise Application Platform 5 users can take advantage of the simplified packaging and deployment rules defined in the Java EE 5 Platform Specification, such as no longer requiring an **application.xml** file in Enterprise Archives (EARs). Additionally, a default library directory (**lib**) in the root directory of an EAR makes the JARs available to all components packaged within the EAR. If an **application.xml** file is included, the **library-directory** element can be used to specify the location of the **lib** directory.

Enterprise Application Platform 5 also introduces a new deployable unit: the *MCBeans archive*, after JBoss Microcontainer, which typically takes the **.beans** or **.deployer** suffix. MCBeans archives package a POJO deployment in a JAR file with a **META-INF/jboss-beans.xml** descriptor. This format is common in Enterprise Application Platform deployers.

Application verification for all file types is enabled by default, and can be configured in the **deployers/ear-deployer-jboss-beans.xml** file, specifically:

```
<!-- uncomment to disable xml validation
  <property name="useValidation">false</property -->
<!-- in case xml validation is disabled, it's also better to turn off
  schema validation
  <property name="useSchemaValidation">false</property -->
```

Enterprise JavaBean 2.0 archive verification remains the same between Enterprise Application Platform 4.x and Enterprise Application Platform 5. However, the properties that control verification have been moved from **deploy/ejb-deployer.xml** to **deployers/ejb-deployer-jboss-beans.xml**.

If an enterprise archive contains only an application client and refers to EJBs, you must also add the **</ignore-dependency>** element to the **ejb-ref** or **ejb-local-ref** definitions in the **jboss-client.xml** deployment descriptor. This informs the deployer to deploy the archive without resolving the referenced dependencies.

2.4.1. Classloading

The new **ClassLoader** is fully backwards compatible, with one exception that does not affect common use (<http://www.jboss.org/community/docs/DOC-12840>). All classloading configurations from JBoss AS 4.x will still work with the new implementation, and most default settings retain the behavior of the previous version.

The new **ClassLoader** shares many design and implementation details with the original **UnifiedClassLoader**, but makes the following improvements:

- ✧ the classloader no longer depends upon JMX, so it can be used in any environment as a standalone.
- ✧ it is much easier to implement your own classloader policy.
- ✧ increased control over which classloaders your classloader delegates to.
- ✧ increased control over which classes are visible to other classloaders.
- ✧ hierarchical repositories have been replaced by domains, and can now extend beyond a single level.



Note

useJBossWebClassLoader="true" is not used in JBoss Enterprise Web Platform 5. All WAR classloaders in Enterprise Application Platform 5 are JBoss **ClassLoader**s, so the **WarDeployer** no longer handles the configuration details for web applications.

There are several methods available to change the classloading configuration of a WAR:

Remove the WarClassLoaderDeployer

The **WarClassLoaderDeployer** automatically implements the defined classloading rules for WARs. Each WAR is assigned a scoped classloading domain. Its classes are not visible to other applications or to any parent EAR, and where possible the WAR's classes are called first. To remove this behavior and make WAR classloading behave like other deployers, comment out the **WarClassLoaderDeployer** in **deployers/jbossweb.deploy/META-INF/war-deployers-jboss-beans.xml**.

Define classloading rules explicitly for the WAR

Add a **WEB-INF/jboss-classloading.xml** with the following content to your WAR.

```
<?xml version="1.0" encoding="UTF-8"?>
<classloading xmlns="urn:jboss:classloading:1.0"
  name="mywar.war"
```

```
domain="DefaultDomain"  
export-all="NON_EMPTY"  
import-all="true">  
</classloading>
```

This lets you define how the WAR's classloader is constructed. In this case, the WAR's classloader has been placed in the **DefaultDomain**, which is shared with all other applications that do not define their own domain. **import-all** is enabled, which means the classloader will look at all other classes exported by other applications. **export-all** is set to expose all classes in our application to other classes.

2.4.2. EAR Scoping

You can control how class isolation between deployments behave with the **isolated** property in **deployers/ear-deployer-jboss-beans.xml**, as follows:

```
<!-- A flag indicating if ear deployments should have their own scoped  
class loader to isolate theirclasses from other deployments. -->  
<property name="isolated">false</property>
```

Chapter 3. New Installation

3.1. Pre-Requisites

The JBoss Enterprise Web Platform 5 binaries require around 500MB of disk space. The main requirement of the Platform is RAM. At least 4GB is necessary to comfortably run a 64-bit developer workstation running the production server profile with JBoss Developer Studio. A 32-bit JVM uses less resources than a 64-bit JVM, but does not provide large heaps. A server with 2GB and swap space can be used for testing and development.

JBoss Enterprise Web Platform requires Java JDK 1.6.

3.1.1. Hardware, Operating System, and JVM Requirements

Hardware Requirements

The following table details the minimum hardware requirements for a JBoss Enterprise Web Platform installation that allows for all examples to be run correctly.

Table 3.1. Minimum Hardware Requirements

Component	Requirement
CPU	Intel Pentium 1 GHz or faster for simple applications
Hard disk space	1.5 GB
System RAM	1.5 GB

Supported Operating Systems

JBoss Enterprise Web Platform 5 is supported on any Operating System with a certified JVM. The Native components are supported only on supported Operating Systems. See the JBoss Support Policy for certified JVMs and Supported Operating Systems:

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

3.1.2. Configuring Your Java Environment

Enterprise Application Platform 5 requires a Java 6 JDK or JRE. Refer to [Appendix C, Installing a Java Development Kit on Red Hat Enterprise Linux](#) for instructions on JDK 1.6 installation.

Chapter 4. Installation Options

4.1. Web Services Stack

This release provides two options for the Web Services stack:

JBoss Web Services Native

JBoss Web Services Native is the Java EE 5-compliant JBoss implementation of web services standards. It is the only web services stack for versions of JBoss Enterprise Web Platform prior to 5.1.0, and is the default web services stack in JBoss Enterprise Web Platform 5.1.0.

JBoss Web Services CXF

JBoss Web Services CXF provides most of the features available in Apache CXF (including WS-Security, WS-Policy, WS-Addressing, WS-ReliableMessaging, basic WS-Trust, MTOM), plus common JBoss Web Services stack features like endpoint metrics, record management and endpoint address rewrite. JBoss Enterprise Web Platform 5.1.0 introduces JBoss Web Services CXF stack as an optional Web Services stack.

Select which Web Services stack to use during installation. To change the Web Services stack at a later date, reinstall the Platform.

4.2. PicketLink Federation

This release includes PicketLink Federation as a Technology Preview.

Technology Preview features are not fully supported under Red Hat subscription level agreements (SLAs), may not be functionally complete, and are not intended for production use. These features provide early access to upcoming product innovations, enabling customers to test functionality and provide feedback during the development process. As Red Hat considers making future iterations of Technology Preview features generally available, we provide commercially reasonable efforts to resolve any reported issues that customers experience when using these features.

PicketLink Federation brings Identity Federation and Single Sign-on to the Platform, with support for SAML 2.0, WS-Trust 1.3, XACML 2.0 (via JBossXACML), and OpenID 1.1 and 2.0.

To install PicketLink, use either the ZIP install method or the Graphical install method. The PicketLink Technology Preview is not available in the RPM install method.

4.3. Installation Methods

There are three installation methods:

ZIP download

The ZIP installation method is the easiest and quickest if you are familiar with JBoss technologies, or if you are looking for a light-weight method for testing or development. This method requires some post-installation configuration. For ZIP installation instructions refer to [Chapter 5, ZIP Installation from the Red Hat Customer Portal](#).

RPM installation

RPM installation is suitable for production deployment on Red Hat Enterprise Linux

systems. RPM installation leverages the benefits of RPM for updating, system management, and integration with administration tools. This method requires some post-installation configuration. For RPM installation instructions refer to [Chapter 6, *RPM Installation via Red Hat Network*](#).

Graphical installer

The graphical installer simplifies the installation and configuration process. In addition to installing the base files, the installer offers automation of optional component installation, and basic out-of-the-box security configuration. For graphical installer instructions refer to [Chapter 7, *Installation using the Graphical Installer*](#).

Chapter 5. ZIP Installation from the Red Hat Customer Portal

Procedure 5.1. Installation via ZIP file

Follow this procedure to install JBoss Enterprise Web Platform via ZIP file.

1. Download software

Refer to [Appendix B, The Red Hat Customer Portal](#) for file download instructions.

Choose the **Application Platform <release> Binary** download. If you want to use WS CXF as the Web Services Stack for the Platform, download the **jboss-ep-ws-cxf-5.1.0-installer.zip** file.

2. Unzip **jboss-ewp-<release>.zip** to extract the archive contents into the location of your choice.

Result:

This creates the **jboss-ewp-<release>** directory, with an installation of JBoss Enterprise Web Platform using JBoss WS Native as the Web Services Stack.

3. Optional: Use JBoss WS CXF as the Web Service stack

You need Apache Ant installed and configured on your machine to perform this task.

- a. Extract **jboss-ep-ws-cxf-5.1.0.GA-installer.zip** and move the **jboss-ws-cxf-installer** into the **jboss-as-web** directory of the Enterprise Platform.
- b. At the command line go to the directory **jboss-as-web/jboss-ws-cxf-installer** and run the command **ant**.

Result:

An installer script replaces WS Native with WS CXF.

4. Optional: Install Native Components

Refer to [Chapter 8, Install Native Components](#) for Native Component installation instructions.

5. Perform post-installation configuration

Refer to [Chapter 9, Post Installation Configuration](#) for post-installation configuration instructions.

5.1. HornetQ

HornetQ is included as an alternative JMS provider to JBoss Messaging. See the *HornetQ User Guide* for this release for further information about HornetQ functionality.

Procedure 5.2. Install HornetQ

Apache Ant must be installed and configured on your machine to perform this task. Refer to [Appendix E, Installing Apache Ant](#) for installation instructions.

You must have the correct access.redhat.com entitlements to download and install HornetQ.

1. Download the HornetQ ZIP (**jboss-eap-hornetq-5.1.0.GA-installer.zip**) from the Customer Support Portal.
2. Extract the files from **jboss-eap-hornetq-5.1.0.GA-installer.zip** into your Enterprise Application Platform installation.
3. From the command line, run the HornetQ switching script (**sh jboss-as/extras/hornetq/switch.sh**)

```
./switch.sh
```

Chapter 6. RPM Installation via Red Hat Network

6.1. Red Hat Network

Red Hat Network (<http://rhn.redhat.com>) is a complete systems management platform for Red Hat Enterprise Linux, providing 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 Web 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 Web Platform 5 on a Red Hat Enterprise Linux 4 machine.

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

`jb-ewp-5-i386-es-4-rpm`

32-bit AS

`jb-ewp-5-i386-as-4-rpm`

64-bit ES

`jb-ewp-5-x86_64-es-4-rpm`

64-bit AS

`jb-ewp-5-x86_64-as-4-rpm`

2. **Install JBoss Enterprise Web Platform**

Run the following commands, replacing *WS_CHOICE* with one of **jbossas-web-ws-native** or **jbossas-ws-cxf-ewp**:

```
up2date WS_CHOICE jbossas-web
up2date jboss-seam2 resteasy rh-ewp-docs jboss-eap5-native jbossas-
web-tp-licenses
```

3. **Optional: Install Native Components**

Refer to [Chapter 8, *Install Native Components*](#) for Native Component installation instructions.

4. Perform post-installation configuration

Refer to [Chapter 9, *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 Web Platform 5 on a Red Hat Enterprise Linux 5 machine.

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

`jb-ewp-5-i386-server-5-rpm`

64-bit

`jb-ewp-5-x86_64-server-5-rpm`

2. Install JBoss Enterprise Web Platform

Available options are:

✧ **`CURRENT_REPO`**: for **32-bit**, use **`rhel-i386-server-6`**; for **64-bit**, use **`rhel-x86_64-server-6`**

✧ **`WS_CHOICE`**: **`jbossas-web-ws-native`** or **`jbossas-web-ws-cxf`**

Run these commands with the chosen values for **`CURRENT_REPO`**, and **`WS_CHOICE`**.

```
yum remove classpathx-jaf
yum upgrade --disablerepo=CURRENT_REPO
yum install WS_CHOICE jbossas-web
yum install jboss-seam2 resteasy rh-ewp-docs jboss-eap5-native
jbossas-web-tp-licenses
```

3. Optional: Install Native Components

Refer to [Chapter 8, *Install Native Components*](#) for Native Component installation instructions.

4. Perform post-installation configuration

Refer to [Chapter 9, *Post Installation Configuration*](#) for post-installation configuration instructions.

6.4. Install on Red Hat Enterprise Linux 6

Procedure 6.3. Install on Red Hat Enterprise Linux 6

This procedure installs the latest version of JBoss Enterprise Web Platform 5 on a Red Hat Enterprise Linux 6 machine.

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 6 channel names

32-bit

`jb-ewp-5-i386-server-6-rpm`

64-bit

`jb-ewp-5-x86_64-server-6-rpm`

2. Install JBoss Enterprise Web Platform

Available options are:

✧ *CURRENT_REPO*: for **32-bit**, use **rhel-i386-server-6**; for **64-bit**, use **rhel-x86_64-server-6**

✧ *WS_CHOICE*: **jbossas-web-ws-native** or **jbossas-web-ws-cxf**

Run these commands with the chosen values for *CURRENT_REPO*, and *WS_CHOICE*.

```
yum remove classpathx-jaf
yum upgrade --disablerepo=CURRENT_REPO
yum install WS_CHOICE jbossas-web
yum install jboss-seam2 resteasy rh-ewp-docs jboss-eap5-native
jbossas-web-tp-licenses
```

3. Optional: Install Native Components

Refer to [Chapter 8, Install Native Components](#) for Native Component installation instructions.

4. Perform post-installation configuration

Refer to [Chapter 9, Post Installation Configuration](#) for post-installation configuration instructions.

Chapter 7. Installation using the Graphical Installer

Procedure 7.1. Installation via the Graphical Installer

This procedure installs the Platform via the Graphical Installer.

1. Download software

Refer to [Appendix B, The Red Hat Customer Portal](#) for file download instructions.

To install JBoss Enterprise Web Platform via the Graphical Installer, choose the **Enterprise Web Platform <release> Installer** download.

2. Run the installer

Execute the following command in the directory that contains the downloaded installer JAR:

```
java -jar jboss-ewp-installer-<release>.jar
```

On a Linux system, this must be executed as **root**. Under Windows, execute it from a command prompt with elevated privileges.

3. Language

Choose the language for the installation instructions.

4. License Agreement

Read the License Agreement carefully. You must accept the terms of the agreement to proceed with the installation. If you agree to the terms of the agreement, select the "**I accept the terms of this license agreement**" option.

5. Installation Path

Select the destination directory for JBoss Enterprise Web Platform. Type a complete path or browse for a destination directory. If the directory you enter does not exist, the installer creates the target directory in the specified path. If the directory exists already, the installer will overwrite the contents of the directory. In either case the installer prompts you to confirm the action.

The default installation path in Linux is: **/usr/local/EnterprisePlatform-5.1.0**

The default installation path in Windows Server is: **C : \Program Files\EnterprisePlatform-5.1.0**

6. Web Services

Select the Web Services stack you wish to install. The two choices are **WSNative** and **WSCXF**. Only one stack can be selected. Changing the Web Services stack after installation requires reinstalling.

Refer to [Chapter 4, Installation Options](#) for a description of the alternatives.

7. Select Packs

There is one optional component for this release: the PicketLink Federation Tech Preview.

To install the PicketLink Federation Tech Preview:

- a. Click on **eap-core**
- b. Click the arrow to the left of **eap-core** to expand the options.
- c. Click the **picketlink-federation** checkbox.

8. JMX Security

The installer creates a new JAAS security domain with an active user.

Optional: secure consoles and invokers using this security domain.

- a. Supply a password for the admin user in the new JAAS security domain.
- b. Optional: change the username for the JAAS security domain admin user.
- c. Optional: change the name of the JAAS security domain.
- d. Optional: secure the JMX and Web consoles, and http and jmx invokers using the new JAAS security domain. The default is to secure all consoles and invokers.

Result

The JAAS security domain is created and used to secure the Admin console and Tomcat console. The JAAS security domain is also used to secure any consoles and invokers specified in this step.

9. Release Notes

Updated release notes are available at <http://docs.redhat.com>.

10. Confirm Selections

Review the installation selections, then click **Next** to begin writing files to disk.

11. Set up Shortcuts

Create desktop and start menu shortcuts on this screen. If you are running the installer as the administrator (Windows) or root user (Linux), you have the option to create desktop and start menu shortcuts for all users; otherwise you are able to create shortcuts for the currently logged in user only.

12. Optional: Install Native Components

Refer to [Chapter 8, Install Native Components](#) for Native Component installation instructions.

13. Perform post-installation configuration

Refer to [Chapter 9, Post Installation Configuration](#) for post-installation configuration instructions.

Chapter 8. Install Native Components

The Native Components Package

The Native Components package is an optional component for the JBoss Enterprise Web Platform that incorporates native operating system components and connectors for web servers, including OpenSSL, JBoss Native, `mod_jk`, `mod_cluster`, NSAPI for Solaris, ISAPI for Windows, HornetQ LibAIO Native for Red Hat Enterprise Linux.

Installing JBoss Native results in higher server performance, as native operating system code becomes available for the server to optimize tasks.

For more information on configuring the web server connectors, refer to the *HTTP Connectors Load Balancing Guide*

Native Components Manifest

- ✧ JBoss Native consists of the Apache Portable Runtime (APR), OpenSSL and Tomcat Native (TC-native);
 - **Apache Portable Runtime (APR)** provides superior scalability, performance, and improved integration with native server technologies. APR is a highly portable library that is at the heart of Apache HTTP Server 2.x. It enables access to advanced IO functionality (for example: `sendfile`, `epoll` and OpenSSL), Operating System level functionality (for example: random number generation and system status), and native process handling (shared memory, NT pipes and Unix sockets).
 - **OpenSSL** implements the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols and includes a basic cryptographic library.
 - **Tomcat Native (TC-Native)** is a Java Native Interface (JNI) that provides much of Tomcat's core functionality in native code rather than Java. This allows for an overall increase in the speed of a server.
- ✧ **mod_jk** connects the Tomcat JSP container to the Apache webserver, providing load-balancing.
- ✧ **mod_cluster** is an `httpd`-based load balancer. In contrast to `mod_jk`, `mod_cluster` creates a feedback loop between the proxy server and the worker nodes, enabling intelligent load distribution and routing within a load-balancing cluster.
- ✧ **ISAPI** is a connector for the Microsoft IIS web server.
- ✧ **NSAPI** is a connector for Sun Java System Web Server, also known as Oracle iPlanet Web Server.
- ✧ HornetQ LibAIO is used as a bridge between HornetQ and Linux LibAIO. It is used in HornetQ's high performance journal, when configured.

8.1. Red Hat Enterprise Linux-specific notes

Red Hat Enterprise Linux includes some of the Native Components in the base operating system. These include OpenSSL and the Apache Portable Runtime (APR). The Apache Portable Runtime is provided by the packages **apr** and **apr-util**.

If the server is started without the **apr** and **apr-util** packages installed, a message similar to the following will appear in logs:

```
WARN [AprLifecycleListener] The Apache Tomcat Native library which allows
optimal performance
in production environments was not found on the java.library.path:
/home/ewpuser/jboss-ewp-5.1/native/lib.
```

8.2. Solaris-specific notes

Both the 32-bit and 64-bit versions of **jboss-ep-native** can be installed on the same machine. The libraries for each are separated by the directories **lib** and **lib64** respectively and each is automatically loaded depending on the JVM version that is used.

To install both 32-bit and 64-bit versions of **jboss-ep-native**, use **unzip -qo**. The **-o** option ensures that one version of the package does not replace another during the installation.

8.3. Native Components Installation

The following procedure describes installing either the `mod_cluster` or `mod_jk` load-balancing modules into the Enterprise Application Platform.

Procedure 8.1. Install Native Components from RPM

1. Subscribe to the JBOSS EAP5 RHN channel

- a. Using a web browser, navigate to <http://access.redhat.com> and log in with your credentials.
- b. View the list of all systems, and find the system on which you have installed the Enterprise Platform. Click to view its subscriptions.
- c. Add the JBoss Application Platform or JBoss EWP channel appropriate to your version of Red Hat Enterprise Linux.

2. Install the `mod_cluster-jbossas` package

Log into the application server's host system as the root user. issue the command **yum install mod_cluster-jbossas**

3. Alternative: Install the `mod_jk-ap20` package

Only follow this step if you need to use `mod_jk` instead of `mod_cluster`. Log into the application server's host system as the root user. s the root user, issue the command **yum install mod_jk-ap20**.

Procedure 8.2. Install Native Components from ZIP Archives

This procedure installs the Native Components for JBoss Enterprise Web Platform.

Prerequisite:

Install JBoss Enterprise Web Platform via ZIP, RPM, or the Graphical installer before carrying out this procedure. See [Section 4.3, "Installation Methods"](#) for more details.

1. Download software

Refer to [Appendix B, The Red Hat Customer Portal](#) for file download instructions.

To install Native Components, choose the Native Components download that corresponds to your operating system and the architecture of your Java Virtual Machine.

2. Unzip components

Extract the **native** directory from the zip file into the **jboss-ewp-5.x** directory, so that the native directory is at the same directory level as the **jboss-as-web** directory.

Result:

The Native Components are installed.

3. Verify installation

During server startup the server will report the presence of the Native libraries:

```
12:12:29,826 INFO [ServerInfo] VM arguments: -Dprogram.name=run.sh
-Xms1303m -Xmx1303m
        -XX:MaxPermSize=256m -Dorg.jboss.resolver.warning=true -
Dsun.rmi.dgc.client.gcInterval=3600000
        -Dsun.rmi.dgc.server.gcInterval=3600000 -
Dsun.lang.ClassLoader.allowArraySyntax=true
        -Djava.protocol.handler.pkgs=org.jboss.handlers.stub -
Djava.net.preferIPv4Stack=true
        -Djava.library.path=/home/ewpuser/jboss-ewp-
5.1/native/lib64
        -Djava.endorsed.dirs=/home/ewpuser/jboss-ewp-5.1/jboss-
as/lib/endorsed
```

The option **-Djava.library.path=/home/ewpuser/jboss-ewp-5.1/native/lib64** shows that the server is detecting and loading the Native libraries.

Chapter 9. Post Installation Configuration

9.1. 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 9.1, "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 [Appendix A, 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 9.1, "Create jmx-console, admin-console, and http invoker user account"](#) to edit your user account, or create a new one.

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

Procedure 9.1. 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-as-web/server/$PROFILE/conf/props/jmx-console-users.properties`.
- b. Create a *username = password* pair.



Default admin user configuration

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-as-web/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.

9.1.2. 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 9.2. 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</
depends>
```

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>
```

9.1.3. Security Configuration: Web Console

Procedure 9.3. Create web console user account

This procedure creates a user with access permissions to the web console

1. **Create a user in the web-console JAAS security domain**

- a. Edit the file `web-console-users.properties` in `jboss-as-web/server/$PROFILE/deploy/management/console-mgr.sar/web-console.war/WEB-INF/classes/`.
- b. Create a `username = password` pair.



Default admin user configuration

The commented `admin=admin` username and password 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 `web-console-roles.properties` in `jboss-as/server/$PROFILE/deploy/management/console-mgr.sar/web-console.war/WEB-INF/classes/`.
- b. Create an entry for the user of the form:

```
username=JBossAdmin,HttpInvoker
```

JBossAdmin

Grant the user permission to access the Web-Console

HttpInvoker

Grant the user permission to access the HTTP Invoker



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.

9.1.4. 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 9.4. Set `suckerPassword` for JBoss Messaging:

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

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

9.2. Default Database



Do not use the Hypersonic database in production

By default, persistence is configured to use Hypersonic (HSQLDB). This allows the JBoss Enterprise Web Platform to function immediately after installation as a development platform. However, *Hypersonic is not supported in production and should not be used in a production environment.*

The Hypersonic database, while useful as a light-weight database for development, is not suitable for production use. Some of its limitations include:

- no transaction isolation
- thread and socket leaks (`connection.close()` does not tidy up resources)
- low persistence quality (logs commonly become corrupted after a failure, preventing automatic recovery)
- database corruption
- instability under load (database processes cease when dealing with too much data)
- not viable in clustered environments

Refer to the *Getting Started Guide* for database configuration instructions.

9.3. Memory Settings for the Enterprise Web Platform

The optimal memory settings for an application server are highly dependent on the exact applications used, the number of users, the virtual or physical host upon which the installation resides, and other services running on that host.

The Enterprise Web Platform ships with default values for initial and maximum heap allocations by the JVM. These values are:

- **-Xms1303m**: Initial heap size, set in megabytes
- **-Xmx1303m**: Maximum heap size, set in megabytes

Guidelines for memory settings for the Enterprise Web Platform:

- Allocate the same values for initial and maximum heap sizes
- Use values smaller than the host's allocatable memory
- Be aware of other services and applications running on the host, and allow for their usage of memory

Fine tuning the memory settings beyond these guidelines requires production-like load testing and analysis of memory usage logs, and is highly variable between installations and applications used with the Enterprise Web Platform.

Procedure 9.5. Changing Memory Settings for the Enterprise Web Platform on Linux

1. Navigate to **JBOSS_DIST/jboss-as/bin**.
2. Using a text editor, open **run.conf**.
3. The memory options are set on this line:

```
JAVA_OPTS="-Xms1303m -Xmx1303m -XX:MaxPermSize=256m -
Dorg.jboss.resolver.warning=true -
Dsun.rmi.dgc.client.gcInterval=3600000 -
Dsun.rmi.dgc.server.gcInterval=3600000 -
Dsun.lang.ClassLoader.allowArraySyntax=true"
```

Edit the line to include the new initial and maxium heap sizes for the JVM:

```
JAVA_OPTS="-XmsINITIAL_HEAP_SIZE -XmxMAX_HEAP_SIZE -
XX:MaxPermSize=256m -Dorg.jboss.resolver.warning=true -
Dsun.rmi.dgc.client.gcInterval=3600000 -
Dsun.rmi.dgc.server.gcInterval=3600000 -
Dsun.lang.ClassLoader.allowArraySyntax=true"
```

4. The new settings will take effect when the Enterprise Web Platform is shut down and restarted.

Procedure 9.6. Changing Memory Settings for the Enterprise Web Platform on Windows

1. Navigate to **JBOSS_DIST\jboss-as\bin**.
2. Using a text editor, open **run.conf.bat**.
3. The memory options are set on this line:

```
set "JAVA_OPTS=-Xms1303m -Xmx1303m -XX:MaxPermSize=256m -
Dorg.jboss.resolver.warning=true
-Dsun.rmi.dgc.client.gcInterval=3600000 -
Dsun.rmi.dgc.server.gcInterval=3600000
-Dsun.lang.ClassLoader.allowArraySyntax=true"
```

Edit the line to include the new initial and maxium heap sizes for the JVM:

```
set "JAVA_OPTS=-XmsINITIAL_HEAP_SIZE -XmxMAX_HEAP_SIZE -
XX:MaxPermSize=256m -Dorg.jboss.resolver.warning=true -
Dsun.rmi.dgc.client.gcInterval=3600000 -
Dsun.rmi.dgc.server.gcInterval=3600000 -
Dsun.lang.ClassLoader.allowArraySyntax=true"
```

4. The new settings will take effect when the Enterprise Web Platform is shut down and restarted.

9.4. Run the Application Server as a Service

Procedure 9.7. Running as a Service on Microsoft Windows Server

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 JBoss Enterprise Web 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.



Uninstalling a Service

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

Chapter 10. Test your Installation

Procedure 10.1. Test the Platform installation

This procedure performs a basic check of the Platform installation

1. Start the Server

There are several options to start the server:

a. Option 1 - Shortcut

Start the server using a desktop or start menu shortcut created by the Graphical Installer.

b. Option 2 - `run.sh` / `run.bat`

Start the server using the `run.sh` (Linux) or `run.bat` (Windows) script.

Execute the following command in a terminal in the `jboss-as/bin` directory:

Linux

```
./run.sh
```

Windows

```
run.bat
```

Result:

The server starts using the **default** profile.

2. Test the Server homepage

Open `http://127.0.0.1:8080` in a web browser on the server machine.

Result:

The JBoss Enterprise Web Platform server homepage is displayed.

Chapter 11. Uninstall JBoss Enterprise Web Platform

The graphical installer creates an uninstall utility, and optionally a shortcut icon. The uninstall utility is **Uninstaller/uninstaller.jar** in the JBoss Enterprise Web Platform top-level directory.

JBoss Enterprise Web Platform can be uninstalled with the uninstall utility, or by deleting the top-level directory of the installation.

Disabling Authentication

This appendix enables a user to disable authentication for specific services.

All specified paths in the sections below are relative to the **jboss-as-web** directory.

Disabling Authentication for JMX Console:

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

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

Comment out the following <security-constraint> block:

```
<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>
  </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/$PROFILE/deploy/management/console-mgr.sar/web-console.war/WEB-INF/web.xml
```

Comment out the following <security-constraint> block:

```
<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>
  </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/$PROFILE/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>
  </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/$PROFILE/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:

Comment out the `<interceptor>` block that specifies the **AuthenticationInterceptor** module:

```
<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>
```

Disabling Authentication for the ProfileService:

To disable authentication for the **ProfileService**, edit the following file and comment out the contents of the **serverProxyInterceptors** list:

```
deploy/profileservice-jboss-beans.xml
```

Comment out the following <bean> block:

```
<bean class="org.jboss.aspects.security.AuthenticationInterceptor">
  <constructor>
    <parameter>
      <value-factory bean="JNDIBasedSecurityManagement"
method="getAuthenticationManager" parameter="jmx-console"/>
    </parameter>
  </constructor>
</bean>
<bean
class="org.jboss.aspects.security.RoleBasedAuthorizationInterceptor">
  <constructor>
    <parameter>
      <value-factory bean="JNDIBasedSecurityManagement"
method="getAuthenticationManager" parameter="jmx-console"/>
    </parameter>
    <parameter>
      <value-factory bean="JNDIBasedSecurityManagement"
method="getAuthenticationManager" parameter="jmx-console"/>
    </parameter>
  </constructor>
</bean>
```

Disabling Authentication for JBossWS:

To disable authentication for JBossWS, edit the following file and comment out the <security-constraint>:

```
deploy/jbossws.sar/jbossws-management.war/WEB-INF/web.xml
```

Comment out the following <security-constraint> block:

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>ContextServlet</web-resource-name>
    <description>An example security config that only allows users with
the role 'friend' to access the JBossWS console web application
    </description>
    <url-pattern>/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>friend</role-name>
  </auth-constraint>
</security-constraint>
```

The Red Hat Customer Portal

The Red Hat Customer Portal at <http://access.redhat.com> provides access to the value of the Red Hat Subscription, including knowledge base articles, support case management, and file downloads.



Prerequisites

To download JBoss Enterprise Web Platform you need a login to the Red Hat Customer Portal (<http://access.redhat.com>) with a valid JBoss Enterprise Web Platform subscription.

Procedure B.1. Downloading Files

This procedure downloads files needed to install JBoss Enterprise Web Platform.

1. Open <http://access.redhat.com> in a web browser.
2. Click the **Downloads** option in the menu across the top of the page.
3. Click on **Download your software** in the list under JBoss Enterprise Middleware.
4. Enter your login information.

Result:

You are taken to the Software Downloads page.

5. Select **Enterprise Web Platform** from either the drop-down box or the menu on the left.

Result:

You are presented with a list of file downloads.

- ✦ See [Chapter 7, Installation using the Graphical Installer](#) for Graphical Installer instructions.
- ✦ See [Chapter 5, ZIP Installation from the Red Hat Customer Portal](#) for ZIP installation instructions.
- ✦ See [Chapter 8, Install Native Components](#) for Native Component installation instructions.

Installing a Java Development Kit on Red Hat Enterprise Linux

Red Hat supports the JBoss Enterprise Web 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. The JBoss Enterprise Web Platform Platform is also supported on Red Hat Enterprise Linux 5 when it is run using **OpenJDK** 1.6. These JDKs can be installed by using the Red Hat Network (RHN).



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.

C.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 C.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 C.4, "Setting the default JDK with the /usr/sbin/alternatives Utility"](#)

C.2. Sun Java Development Kit on Red Hat Enterprise Linux 5

Use this procedure to install the **Sun Microsystems Java Development Kit** on **Red Hat Enterprise Linux 5**.

**Important**

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

Procedure C.2. Installing the Sun Microsystems JDK on Red Hat Enterprise Linux 5**1. Subscribe to Supplementary Server channel.**

The **Sun Microsystems Java Development Kit** is available in the **Supplementary Server** channel.

2. Install the package.

To install the **Sun Microsystems Java Development Kit** package, input this command:

```
yum install java-1.6.0-sun-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 C.4, “Setting the default JDK with the /usr/sbin/alternatives Utility”](#)

C.3. Sun JDK on Red Hat Enterprise Linux AS/ES 4

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

**Important**

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

Procedure C.3. Installing the Sun Microsystems JDK on Red Hat Enterprise Linux AS/ES 4**1. Subscribe to the Extras channel.**

The **Sun Microsystems 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-1.6.0-sun-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 C.4, “Setting the default JDK with the `/usr/sbin/alternatives` Utility”](#)

C.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 C.4, “Using `/usr/sbin/alternatives` to Set the Default JDK”](#) for syntax of the `/usr/sbin/alternatives` command.

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

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

Table C.2. javac alternative commands

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

JDK	alternative command
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`.

Installing the Sun JDK on Microsoft Windows

Procedure D.1. Installing and Configuring the 32-bit Sun JDK on Microsoft Windows

1. Download the Software

Download the **Sun Java 2 Development Kit** from <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.

2. Create an environmental variable called **JAVA_HOME** that points to directory in which the JDK will be installed, such as **C:\Program Files\Java\jdk1.6.0_16**. To do this, click on the **Start Menu**, open the **Control Panel**, (if necessary, switch to **Classic View**), open the **System Control Panel** applet, select the **Advanced Tab**, and click on the **Environment Variables** button.

3. Add the JDK's **bin** directory to the path **PATH**.

To do this, open the **Control Panel** from the **Start Menu**, (if necessary, switch to **Classic View**), then edit the **PATH** environment variable found in **System -> Advanced -> Environment Variables -> System Variables**. Append a semicolon and **%JAVA_HOME%\bin** to the end of the **PATH** value.

4. So that Java can be run from the command line, add the **jre\bin** directory to the path so that it looks similar to **C:\Program Files\Java\jdk1.5.0_11\jre\bin**.

Installing Apache Ant

The Java build tool *Apache Ant* is not required for the installation or normal operation of the JBoss Enterprise Application Platform. However, it is occasionally needed for some configuration tasks and also for building and deploying some applications.



Note

If running a development workstation, Apache Ant may already be installed.



Note

To learn more about **Apache Ant**, visit the project's website at <http://ant.apache.org>.

Procedure E.1. Installing Apache Ant on Red Hat Enterprise Linux

- » Download and install **Apache Ant** on Red Hat Enterprise Linux Repository by issuing this command:

```
[localhost]$ sudo yum install ant
```

Procedure E.2. Installing Apache Ant on Other Operating Systems

1. Download and Extract

Download the **Apache Ant** binary release from <http://ant.apache.org/bindownload.cgi>.

Once it is downloaded, extract it in a preferred installation location, such as **c:\Program Files\Apache\Ant** or **/opt/apache-ant-1.8/**.

2. Add the ANT_HOME Environmental Variable

Next, create an environmental variable called **ANT_HOME**. This variable has to contain the path created in the previous step.

- » Do this on Red Hat Enterprise Linux by adding the following line to the **~/.bash_profile** file, substituting the path with that created above.

```
export ANT_HOME=/opt/apache-ant-1.7.1
export ANT_HOME=/opt/apache-ant-1.7.1
```

- » On Microsoft Windows, do this by click on the **Start Menu** and opening the **Control Panel** then selecting **System -> Advanced -> Environment Variables**. Create a new variable, calling it **ANT_HOME** and configure it to point to the **ant** directory.

3. Include bin in the PATH

Next, append the **ant** installation's **bin** directory the **PATH** environmental variable.

- ✳ On Unix/Linux systems, one does this simply by adding the following line to the `~/.bash_profile` file after the one which sets the **ANT_HOME** variable:

```
export PATH=$PATH:$ANT_HOME/bin
export PATH=$PATH:$ANT_HOME/bin
```

- ✳ On Microsoft Windows, do this task by opening the **Control Panel** then selecting **System** -> **Advanced** -> **Environment Variables**->**System Variables** -> **Path**. Create a new variable, calling it **ANT_HOME**. Next, add a semicolon and `%ANT_HOME%\bin` to the end of the path value.

To test the **Apache Ant** installation, run **ant -version** from within a command line shell. The output should look similar to this:

```
[localhost]$ ant -version
Apache Ant version 1.8 compiled on June 27 2008
```

Revision History

Revision 5.1.1-105	March 5, 2015	Ella Deon Ballard
Fixing typo in Appendix D.		
Revision 5.1.1-104.400	2013-10-31	Rüdiger Landmann
Rebuild with publican 4.0.0		
Revision 5.1.1-104	2012-07-18	Anthony Towns
Rebuild for Publican 3.0		
Revision 5.1.1-100	Mon Jul 18 2011	Jared Morgan
Incorporated changes for JBoss Enterprise Web Platform 5.1.1 GA. For information about documentation changes to this guide, refer to <i>Release Notes 5.1.1</i> .		
Revision 5.1.0-105	Wed Sep 29 2010	Laura Bailey, Joshua Wulf
Revised for JBoss Enterprise Web Platform 5.1.0.GA.		