JBoss Enterprise Application Platform 5

Installation Guide

for Use with JBoss Enterprise Application Platform 5

Edition 5.2.0

Last Updated: 2017-10-13

JBoss Enterprise Application Platform 5 Installation Guide

for Use with JBoss Enterprise Application Platform 5
Edition 5.2.0

Eva Kopalova

Jared Morgan

Petr Penicka

Russell Dickenson

Scott Mumford

Legal Notice

Abstract

This Installation Guide documents relevant information regarding the installation of JBoss Enterprise Application Platform 5 and its patch releases.

Table of Contents

CHAPTER 1. INTRODUCTION

JBoss Enterprise Application Platform is the open source implementation of the Java EE suite of services. It comprises a set of offerings for enterprise customers who are looking for preconfigured profiles of JBoss Enterprise Middleware components that have been tested and certified together to provide an integrated experience. 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 Application Platform is cross-platform, easy to install and use on any operating system that supports Java. The readily available source code is a powerful learning tool to debug the server and understand it. It also gives you the flexibility to create customized versions for your personal or business use.

1.1. OTHER MANUALS

If you are looking for detailed product information refer to the manuals available online at http://docs.redhat.com/docs/en-US/JBoss_Enterprise_Application_Platform/index.html.

PART I. INSTALLING JBOSS ENTERPRISE APPLICATION PLATFORM 5

CHAPTER 2. PRE-REQUISITES

The JBoss Enterprise Application 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.

2.1. HARDWARE, OPERATING SYSTEM, AND JVM REQUIREMENTS

Hardware Requirements

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

Table 2.1. 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

Operating System, and JVM Requirements

JBoss Enterprise Application 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/.

Refer to Appendix B, *Installing a Java Development Kit* for JDK installation instructions.

CHAPTER 3. INSTALLATION METHODS

There are three installation methods:

ZIP Installation

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 4, *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 5, *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 6, *Installation using the Graphical Installer*.

CHAPTER 4. ZIP INSTALLATION FROM THE RED HAT CUSTOMER PORTAL

Procedure 4.1. Installation via ZIP file

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

1. **Download software**
   Refer to Appendix A, *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-<release>-installer.zip`**. file.

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

   In a Red Hat Enterprise Linux environment, use the **`unzip`** utility to extract the Zip archive.

   In a Microsoft Windows environment, right-click the file and select **`Extract All`**.

   In a Hewlett-Packard HP-UX environment, use the **`unzip`** utility to extract the Zip archive.

   Result:

   This creates the **`jboss-eap-<release>`** directory, with an installation of JBoss Enterprise Application Platform using JBoss WS Native as the Web Services Stack, and JBoss Messaging as the messaging provider.

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-<release>.GA-installer.zip`** and move the **`jbossws-cxf-installer`** into the **`jboss-as`** directory of the Enterprise Platform.

   b. At the command line go to the directory **`jboss-as/jbossws-cxf-installer`** and run the command **`ant`**.

      Result:

      An installer script replaces WS Native with WS CXF.

4. **Optional: Install PicketLink Federation**

   a. To install PicketLink Federation, copy the **`$JBOSS_HOME/picketlink/picketlink-federation/picketlink-core-<VERSION>.jar`** file to **`$JBOSS_HOME/common/lib`** and copy **`$JBOSS_HOME/picketlink/picketlink-federation/picketlink-jbas5-VERSION.jar`** file to **`$JBOSS_HOME/common/lib`**;

   b. Optionally, deploy the PicketLink web applications of your choice to the server by copying their directories to **`$JBOSS_HOME/jboss-as/server/PROFILE/deploy/`** directory. To do so, run the following command with the WEBAPP substituted with the application directory (**`idp.war`**, **`pdp.war`**, or **`picketlink-sts.war`**):

```
cp -r $JBOSS_HOME/picketlink/picketlink-federation-webapps/WEBAPP
$JBOSS_HOME/jboss-as/server/PROFILE/deploy/
```

5. **Optional: Install Native Components**
   Refer to Section 8.3, “Native Components” for Native Component installation instructions.

6. **Perform post-installation configuration**
   At this point, you have JBoss Enterprise Application Platform installed. It is now recommended to perform post-installation configuration of the platform according to instructions in the *Administration and Configuration Guide*. For instructions on how to configure the platform's security, refer to the *Security Guide*.

4.1. 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 4.2. Install HornetQ

Apache Ant must be installed and configured on your machine to perform this task. Refer to Appendix C, *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-release-installer.zip`) from the Customer Support Portal.

2. Extract the files from `jboss-eap-hornetq-release-installer.zip` into your JBoss Enterprise Application Platform installation (the archive contains the entire `jboss-eap-5.2` directory structure; therefore, merge the extracted directory with your `jboss-eap-version` directory).

3. Change to `$JBOSS_HOME/jboss-as/extras/hornetq`.

4. Verify the `switch.sh` script is configured to be executable.

5. From the command line, run the HornetQ switching script.

   ```
   [hornetq]$ ./switch.sh
   ```

CHAPTER 5. RPM INSTALLATION VIA RED HAT NETWORK

5.1. 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 Application Platform.

5.2. INSTALL ON RED HAT ENTERPRISE LINUX 4

Procedure 5.1. Install on Red Hat Enterprise Linux 4

This procedure installs the latest version of JBoss Enterprise Application 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

   jbappplatform-5-i386-es-4-rpm

   rhel-i386-es-4-extras

   32-bit AS

   jbappplatform-5-i386-as-4-rpm

   rhel-i386-as-4-extras

   64-bit ES

   jbappplatform-5-x86_64-es-4-rpm

   rhel-x86_64-es-4-extras

   64-bit AS

   jbappplatform-5-x86_64-as-4-rpm

   rhel-x86_64-as-4-extras

2. **Install JBoss Enterprise Application Platform**
   Run the following commands, replacing *MESSAGING_CHOICE* with one of `jbossas-messaging` or `jbossas-hornetq`; and replacing *WS_CHOICE* with one of `jbossas-ws-native` or `jbossas-ws-cxf`:

```
up2date MESSAGING_CHOICE WS_CHOICE jbossas
up2date jboss-seam2 resteasy rh-eap-docs
```

3. **Optional: Install PicketLink**
   Run the following command to install PicketLink:

   ```
   up2date picketlink-federation
   ```

   Optionally, install any of the additional picketlink packages: picketlink-federation-webapp-idp, picketlink-federation-webapp-pdp, picketlink-federation-webapp-sts

4. **Optional: Install Native Components**
   Refer to Section 8.3, “Native Components” for Native Component installation instructions.

5. **Apply security patches**
   Refer to Chapter 7, *Application of the Latest Security Patches*for security-patches application instructions.

6. **Perform post-installation configuration**
   At this point, you have JBoss Enterprise Application Platform installed. It is now recommended to perform post-installation configuration of the platform according to instructions in the *Administration and Configuration Guide*. For instructions on how to configure the platform's security, refer to the *Security Guide*.

5.3. INSTALL ON RED HAT ENTERPRISE LINUX 5

Procedure 5.2. Install on Red Hat Enterprise Linux 5

This procedure installs the latest version of JBoss Enterprise Application 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

   jbappplatform-5-i386-server-5-rpm

   rhel-i386-server-supplementary-5

   64-bit

   jbappplatform-5-x86_64-server-5-rpm

   rhel-x86_64-server-supplementary-5

2. **Install JBoss Enterprise Application Platform**
   Available options are:

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

   - *MESSAGING_CHOICE*: **`jbossas-messaging`** or **`jbossas-hornetq`**
   - *WS_CHOICE*: **`jbossas-ws-native`** or **`jbossas-ws-cxf`**

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

   ```
   yum remove classpathx-jaf
   yum install MESSAGING_CHOICE WS_CHOICE jbossas
   yum install jboss-seam2 resteasy rh-eap-docs
   ```

3. **Optional: Install PicketLink**
   Run the following command to install PicketLink:

   ```
   yum install picketlink-federation
   ```

   Optionally, install any of the additional picketlink packages: picketlink-federation-webapp-idp, picketlink-federation-webapp-pdp, picketlink-federation-webapp-sts

4. **Optional: Install Native Components**
   Refer to Section 8.3, “Native Components” for Native Component installation instructions.

5. **Apply security patches**
   Refer to Chapter 7, *Application of the Latest Security Patches*for security-patches application instructions.

6. **Perform post-installation configuration**
   At this point, you have JBoss Enterprise Application Platform installed. It is now recommended to perform post-installation configuration of the platform according to instructions in the *Administration and Configuration Guide* For instructions on how to configure the platform's security, refer to the *Security Guide*.

5.4. INSTALL ON RED HAT ENTERPRISE LINUX 6

This procedure installs the latest version of JBoss Enterprise Application Platform 5 on a Red Hat Enterprise Linux 6 machine, or upgrades a previous version of JBoss Enterprise Application Platform to the latest version.

Procedure 5.3. Install on Red Hat Enterprise Linux 6

This procedure installs the latest version of JBoss Enterprise Application 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

   jbappplatform-5-i386-server-6-rpm

   rhel-i386-server-supplementary-6

64-bit

jbappplatform-5-x86_64-server-6-rpm

rhel-x86_64-server-supplementary-6

2. **Install JBoss Enterprise Application Platform**
   Available options are:

   - *CURRENT_REPO*: for **`32-bit`**, use **`rhel-i386-server-6`**; for **`64-bt`**, use **`rhel-x86_64-server-6`**
   - *MESSAGING_CHOICE*: **`jbossas-messaging`** or **`jbossas-hornetq`**
   - *WS_CHOICE*: **`jbossas-ws-native`** or **`jbossas-ws-cxf`**

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

   ```
   yum remove classpathx-jaf
   yum install MESSAGING_CHOICE WS_CHOICE jbossas
   yum install jboss-seam2 resteasy rh-eap-docs
   ```

3. **Optional: Install PicketLink**
   Run the following command to install PicketLink:

   ```
   yum install picketlink-federation
   ```

   Optionally, install any of the additional picketlink packages: picketlink-federation-webapp-idp, picketlink-federation-webapp-pdp, picketlink-federation-webapp-sts

4. **Optional: Install Native Components**
   Refer to Section 8.3, “Native Components” for Native Component installation instructions.

5. **Apply security patches**
   Refer to Chapter 7, *Application of the Latest Security Patches*for security-patches application instructions.

6. **Perform post-installation configuration**
   At this point, you have JBoss Enterprise Application Platform installed. It is now recommended to perform post-installation configuration of the platform according to instructions in the *Administration and Configuration Guide*. For instructions on how to configure the platform's security, refer to the *Security Guide*.

CHAPTER 6. INSTALLATION USING THE GRAPHICAL INSTALLER

Task: Install the Platform using the Graphical Installer on Red Hat Enterprise Linux or Microsoft Windows

Complete this task to install the platform using the Graphical Installer.

Prerequisites

- JAVA_HOME is set on the installation target. Refer to Appendix B, *Installing a Java Development Kit*.

1. **Download software**
   Refer to Appendix A, *The Red Hat Customer Portal* for file download instructions.

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

2. **Run the installer**
   Execute the following command in the directory that contains the downloaded installer JAR:

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

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 Application 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.

   Linux

   In Linux-based installations, the default installation path (which is used if another location is not chosen by the user) depends on the filesystem privileges of the user account performing the intstallation:

   - If the user has write access to **`/usr/local/`**, then this is the installation path used.
   - If the user does not have write access to **`/usr/local/`**, then the default installation path is: **`/home/`*`[username]`*`/EnterprisePlatform-`*`[version]`***

   Windows

   The default installation path in Windows Server is: **`C:\Program Files\EnterprisePlatform-`*`[version]`***

6. **Java Messaging Service and Web Services**
   In the top part of this dialog, select the Java Messaging Service that you wish to install. The two available options are **`JBoss Messaging`** and **`HornetQ`**. Only one service can be selected.

   Below, 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 8, *Optional Components* for a description of the alternatives.

7. **Select Packs**
   There is one optional component to choose in this step: PicketLink.

   To install PicketLink:

   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 Section 8.3, “Native Components” for Native Component installation instructions.

13. **Perform post-installation configuration**

At this point, you have JBoss Enterprise Application Platform installed. It is now recommended to perform post-installation configuration of the platform according to instructions in the *Administration and Configuration Guide*. For instructions on how to configure the platform's security, refer to the *Security Guide*.

CHAPTER 7. APPLICATION OF THE LATEST SECURITY PATCHES

After you have installed JBoss Enterprise Application Platform from Red Hat Network, apply the latest security patches available from Red Hat Network for your JBoss Enterprise Application Server:

1. Log in to Red Hat Customer Portal and go to https://access.redhat.com/jbossnetwork/restricted/listSoftware.html.

2. Select the `Application Platform` entry in the `Product` field.

3. After page refresh, select the appropriate JBoss Enterprise Application Platform version in the `Version` field.

4. Click the `Security Advisories` tab.

5. Download the security patches and follow the documentation in the patches to have them installed.

> **WARNING**
>
> By default, the Legacy Invoker Servlet is exposed on all network interfaces and deserializes objects sent to it via HTTP. In addition to applying the latest security patches, we recommended that you only bind this servlet to the loopback network interface so that remote users cannot use it perform deserialization attacks. Failure to restrict access to this servlet could lead to remote code execution being allowed to remote attackers.
>
> For instructions on binding this servlet to the loopback network interface, see https://access.redhat.com/solutions/45530.

CHAPTER 8. OPTIONAL COMPONENTS

8.1. 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 Application Platform prior to 5.1, and is the default web services stack in JBoss Enterprise Application Platform 5.

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 Application Platform 5 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.

8.2. PICKETLINK FEDERATION

This release includes PicketLink Federation as a supported product.

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

Refer to the installation sections for PicketLink installation instructions.

8.3. NATIVE COMPONENTS

The Native Components Package

The Native Components package is an optional component for the JBoss Enterprise Application Platform that incorporates native operating system components and connectors for web servers, including JBoss Native, mod_jk, mod_cluster, NSAPI for Solaris, NSAPI for Oracle iPlanet Web Server, ISAPI for Windows and 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.

- **`HornetQ LibAIO`** is used as a bridge between HornetQ and Linux LibAIO. It is used in HornetQ's high performance journal, when configured.

8.3.1. 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/eapuser/jboss-eap-5.2/native/lib.
```

8.3.2. Hewlett Packard HP-UX-specific notes

OpenSSL libraries are not provided in the Natives package. Instead the library provided with the operating system is used.

8.3.3. 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.

8.3.4. Native Components Installation

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

Procedure 8.1. Install Native Components from RPM

1. **Subscribe to the JBOSS EAP5 RHN channel**

    1. Using a web browser, navigate to http://access.redhat.com and log in with your credentials.

    2. View the list of all systems, and find the system on which you have installed the Enterprise Platform. Click to view its subscriptions.

    3. Add the JBoss Application Platform or JBoss EWP channel appropriate to your version of Red Hat Enterprise Linux.

2. **Install the jboss-eap5-native package**
    Log into the application server's host system as the root user.

    Execute the command `yum install jboss-eap5-native`

3. **Install the mod_cluster-jbossas package**
    Log into the application server's host system as the root user.

    Execute the command `yum install mod_cluster-jbossas`

4. **Optional: Install the mod_jk-ap20 package**
    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.

    Execute 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 Application Platform.

Prerequisite:

Install JBoss Enterprise Application Platform via ZIP, RPM, or the Graphical installer before carrying out this procedure. See Chapter 3, *Installation Methods* for more details.

1. **Download software**
    Refer to Appendix A, *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-eap-5.x` directory, so that the native directory is at the same directory level as the `jboss-as` directory.

    Result:

    The Native Components are installed.

3. **Verify installation**

During server start up 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/eapuser/jboss-eap-5.2/native/lib64
    -Djava.endorsed.dirs=/home/eapuser/jboss-eap-5.2/jboss-
as/lib/endorsed
```

The option **`-Djava.library.path=/home/eapuser/jboss-eap-5.2/native/lib64`** shows that the server is detecting and loading the Native libraries.

CHAPTER 9. TESTING YOUR INSTALLATION

Procedure 9.1. 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** (Red Hat Enterprise Linux/Hewlett-Packard HP-UX) or **run.bat** (Microsoft Windows Server) script.

      NOTE

      For a full list of parameters for **run.bat** see the *Using run.sh* section of the *Getting Started Guide*. The commands for **run.sh** and **run.bat** are identical.

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

      Red Hat Enterprise Linux/Hewlett-Packard HP-UX

      ```
      ./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 Application Platform server homepage is displayed.

CHAPTER 10. UNINSTALLING JBOSS ENTERPRISE APPLICATION PLATFORM

Refer to the tasks in this chapter to correctly remove the platform based on the installation method you originally chose.

10.1. UNINSTALLING JBOSS ENTERPRISE APPLICATION PLATFORM FROM A GRAPHICAL INSTALLATION

Perform the following steps to remove JBoss Enterprise Application Platform from the system in case that you installed it using the Chapter 6, *Installation using the Graphical Installer*method. This procedure is compatible with all operating systems.

1. Navigate to the **JBoss Platform** menu item in your system.
2. Select the **Uninstall Platform** menu item. The `IzPack - Uninstaller` window opens.
3. Check the Force Deletion option in the window to remove all files and folders associated with the platform. The directory structure indicated in the window is the target for removal.
4. > **WARNING**
   >
   > This step completely removes the platform, and all related configuration files stored in the location indicated in Step 3. Ensure you have made copies of configuration files you may want to reuse if you decide to reinstall the platform later.

   Click **`Uninstall`**.
5. The Platform uninstalls, and a file removal status is displayed in the window's status bar.
6. Once the removal process completes, the status bar displays `[Finished]`.
7. Click **`Quit`**.
8. You have completely removed the platform, and all related configuration files and folders from the original installation location. The JBoss Platform menu item is no longer present in the Applications menu.

10.2. UNINSTALLING JBOSS ENTERPRISE APPLICATION PLATFORM FROM A ZIP INSTALLATION

If you installed the platform using the Chapter 4, *ZIP Installation from the Red Hat Customer Portal* method, you only need to delete the root installation directory. The root directory is the one that contains the jboss-as subdirectory.

1. Open a terminal.

2. Navigate to the location where you installed the platform.

3. **WARNING**

   This step completely removes the platform, and all related configuration files stored in the location indicated in Step 2. Ensure you have made copies of configuration files you may want to reuse if you decide to reinstall the platform later.

   Execute the following command, substituting *[root_folder_name]* with the full path, and name of the platform's root installation folder.

   If you installed the platform to a protected directory on your system, ensure you run this command with the correct access privileges.

   ```
   [home]$ rm -r [root_folder_name]
   ```

4. You have completely removed the platform, and all related configuration files and folders from the original installation location.

PART II. MIGRATING TO ENTERPRISE APPLICATION PLATFORM 5

CHAPTER 11. 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.

NOTE

JBoss Enterprise Application 5 is a major release, and includes major changes from JBoss Enterprise Application Platform 4.3. With the release of JBoss Enterprise Application Platform 5.1.1 onwards, an RPM upgrade path from JBoss Enterprise Application Platform 4.3 is no longer available.

JBoss Enterprise Application Platform 4.3 customers that want to upgrade to JBoss Enterprise Application Platform 5 should back-up their production systems and proceed with installing the platform using an installation option described in Part I, “Installing JBoss Enterprise Application Platform 5”.

11.1. 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.

11.1.1. JBoss Application Server 5 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 life cycle 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 EAP 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 EAP, which will allow for fine grained selection of services for both unit testing and embedded scenarios.

11.1.1.1. 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 EAP 4.x *server configuration*. In EAP 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).

11.1.2. 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.

11.1.3. Java Enterprise Edition 5 Compliance

JBoss Enterprise Application 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.

11.1.4. 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.

11.1.5. 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).

11.1.6. 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.

11.1.7. 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 11.3, “Admin Console” for more information about this new management console.

11.1.8. 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.

11.1.9. Distribution with Red Hat Signed JARs

JAR files included with JBoss Enterprise Application 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.

11.2. WHAT'S DIFFERENT IN ENTERPRISE APPLICATION PLATFORM 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.

11.2.1. 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`**
  - **`profileservice-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.

11.2.2. 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.

11.2.3. Differences in Application Server Configuration Files

11.2.3.1. 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 Application 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>
```

11.2.3.2. 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 11.1. 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 11.2. 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>XMLOVERRIDE</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/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.

11.2.3.3. 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.

11.2.3.4. 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.

11.2.3.5. Transactions

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

11.2.3.6. 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.

11.2.3.7. Security

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

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

11.2.3.8. 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`**.

11.3. ADMIN CONSOLE

The first release of the JBoss Enterprise Application 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 Application 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.

11.4. APPLICATIONS

JBoss Enterprise Application 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 Application 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.

11.4.1. 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 EAP 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 Application 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.

11.4.2. 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 12. JBOSS ENTERPRISE APPLICATION PLATFORM PRE-UPGRADE TEST GUIDELINES

JBoss Enterprise Application 5.2.0 is a minor release of JBoss Enterprise Application Platform 5.1.

An in-place upgrade from version 5.1.x to the latest version is available for customers who have installed the platform using RPM.

> **IMPORTANT**
>
> This is a platform upgrade, not an assisted migration. The platform software will be updated to the latest version, however, you will have to update configuration files and verify the compatibility of your applications.

Follow Upgrade JBoss Enterprise Application Platform 5.1.x to the latest 5.1.x version using RPM to test and verify the entire process before applying it to a production system.

Upgrade JBoss Enterprise Application Platform 5.1.x to the latest 5.1.x version using RPM

Follow the overarching guidelines in this task to perform a non-production system pre-upgrade test. Complete this task before upgrading your production systems as a best practice platform upgrade task.

Once you are satisfied with the results collected as a result of the task guidelines, refer to Chapter 5, *RPM Installation via Red Hat Network*for upgrade commands and prerequisites for different operating systems.

Prerequisites

- You have backed-up your JBoss Enterprise Application Platform data and configuration, and have verified you can restore the system to a known state.

1. Stop all JBoss instances.

2. Upgrade the 5.1 install to the latest update level.

3. Locate and examine all **.rpmnew** files installed on your system by the upgrade process.

   ```
   find $JBOSS_HOME -name *.rpmnew -ls
   ```

   Identify the impact of these changes on your infrastructure and your applications.

   Compare the old versions of the new files, and make any necessary changes to the files before performing this upgrade on your production systems.

   During an RPM upgrade, RPM will install new versions of configuration files. These new versions will be saved with the extension .rpmnew, in order to preserve your existing configuration data. After the upgrade look for these files and compare them with your existing configuration files, making any necessary changes.

4. Start all JBoss instances.

5. Systematically test all applications and verify all applications work according to original specifications.

6. Once satisfied with the results of your testing, roll the upgrade out to your production systems.

APPENDIX A. 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.

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

Procedure A.1. Downloading Files

This procedure downloads files needed to install JBoss Enterprise Application 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 `Application 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 6, *Installation using the Graphical Installer*for Graphical Installer instructions.
- See Chapter 4, *ZIP Installation from the Red Hat Customer Portal*for ZIP installation instructions.
- See Section 8.3, "Native Components" for Native Component installation instructions.

APPENDIX B. INSTALLING A JAVA DEVELOPMENT KIT

If installing on Red Hat Enterprise Linux, it's recommended to obtain it from the Red Hat Network (RHN). If you have difficulties subscribing to the correct software channels in Red Hat Network, 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.

B.1. ORACLE JDK ON RED HAT ENTERPRISE LINUX 5 AND 6

Use this procedure to install the **Oracle Java Development Kit** on **Red Hat Enterprise Linux 5** or **Red Hat Enterprise Linux 6**.

IMPORTANT

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

Procedure B.1. Installing Oracle JDK on Red Hat Enterprise Linux 5 and 6

1. **Subscribe to the Supplementary Server channel.**
   The **Oracle Java Development Kit** is available in the **Supplementary Server** channel.

2. **Install the Oracle JDK package.**
   Issue one of the following commands to install the respective **Oracle Java Development Kit** package:

   ```
   # yum install java-1.6.0-sun-devel
   ```

   ```
   # yum install java-1.7.0-oracle-devel
   ```

3. **Set Oracle JDK as the system's default Java Development Kit.**
   You can use the **alternatives** tools to select the default JDK as well as to verify the setting (refer to Section B.6, "Setting the default JDK with the **alternatives** Utility").

B.2. OPENJDK ON RED HAT ENTERPRISE LINUX 5 AND 6

Use this procedure to install **OpenJDK** on **Red Hat Enterprise Linux 5** or **Red Hat Enterprise Linux 6**.

IMPORTANT

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

Procedure B.2. Installing OpenJDK on Red Hat Enterprise Linux 5 and 6

1. **Subscribe to the base channel.**
   The **OpenJDK** is available in **Red Hat Enterprise Linux**'s **base channel**.

2. **Install the OpenJDK package.**
   Issue one of the following commands to install the respective **OpenJDK** package:

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

```
# yum install java-1.7.0-openjdk-devel
```

3. **Set OpenJDK as the system's default Java Development Kit.**
   You can use the `alternatives` tools to select the default JDK or verify the setting (refer to Section B.6, "Setting the default JDK with the **alternatives** Utility").

B.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 B.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 the Sun Microsystems JDK package.**
   Issue the following command to install the package:

   ```
   # up2date java-1.6.0-sun-devel
   ```

3. **Set Sun Microsystems JDK to the system's default Java Development Kit.**
   You can use the `alternatives` tools to select the default JDK or verify the setting (refer to Section B.6, "Setting the default JDK with the **alternatives** Utility").

B.4. JDK ON HEWLETT-PACKARD HP-UX

Refer to the Hewlett-Packard HP-UX official support site for instructions on downloading and installing a supported JDK.

B.5. ORACLE JDK ON MICROSOFT WINDOWS

Use this procedure to install the **Java Development Kit** on **Windows Server 2008 or Windows 2008 R2.**

Procedure B.4. Installing and Configuring JDK on Microsoft Windows

1. **Download and install JDK.**
   Download the **Java Development Kit** from http://www.oracle.com. Run and follow the on-screen instructions to install the JDK.

   ### IMPORTANT

   Make sure that you download the correct JDK: check the Java version and architecture.

2. **Create `JAVA_HOME` environment variable.**
   The variable must point to the directory in which the JDK is installed, such as `C:\Program Files\Java\jdk<VERSION>\`:

   a. Click **Start Menu**.

   b. Click **Control Panel**.

   c. In the `Control Panel` window, write `Control Panel\System and Security\System` and press `Enter`.

   d. In the `System` window, click `Advanced system settings` on the left.

   e. In the `System Properties` dialog box, click the `Environment Variables` button.

   f. In the `System variables` part of the dialog box, click the `New` button.

   g. In the `New System Variable` dialog box, define the `JAVA_HOME` as the variable name and `<DISK>:\Program Files\Java\jdk<VERSION>` as the variable value and click `OK`.

3. **Add the JDK's `bin` directory to `Path`.**
   In the `System variables` part of the `System Properties` dialog box, select the `Path` environment variable and click `Edit`: in the displayed dialog box, append `;%JAVA_HOME%\bin` to the end of the `Path` value.

B.6. SETTING THE DEFAULT JDK WITH THEALTERNATIVES UTILITY

The **alternatives** utility serves for managing different software packages that provide the same functionality. **Red Hat Enterprise Linux** uses **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 B.5, “ Using **alternatives** to Set the Default JDK ” for syntax of the `alternatives` command.

Procedure B.5. Using alternatives to Set the Default JDK

1. **Become the root user.**
   The **alternatives** utility needs to be run with root privileges. Use the `su` command or other mechanism to gain these privileges.

2. **Set `java`.**
   Run the following command: `/usr/sbin/alternatives --config java`

   Next, follow the on-screen directions to ensure that the correct version of `java` is selected. Table B.1, “`java` alternative commands” shows the relevant command settings for each of the different JDKs.

   Table B.1. `java` alternative commands

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

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 B.2, "`javac` alternative commands" shows the appropriate command settings for the different JDKs.

   Table B.2. `javac` alternative commands

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

4. **Extra Step: Set `java_sdk_1.6.0`.**
   The **Sun Microsystems JDK 1.6** requires an additional command be run:

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

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

APPENDIX C. 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 C.1. Installing Apache Ant on Red Hat Enterprise Linux

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

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

Procedure C.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 into 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/Hewlett-Packard HP-UX systems 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
     ```

   - On Microsoft Windows Server, do this by click on the **Start Menu** and opening the **Control Panel** then selecting **System -> Advanced -> Environment Variables**. Create a new variable named **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 Red Hat Enterprise Linux/Hewlett-Packard HP-UX systems, add the following line to the **~/.bash_profile** file after the one which sets the **ANT_HOME** variable:

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

   - On Microsoft Windows Server, 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
```

APPENDIX D. REVISION HISTORY

Revision 5.2.0-109 **Wed May 18 2016** **Lucas Costi**

Added warning for Legacy Invoker Servlet.

Revision 5.2.0-107 **Thu Jul 11 2013** **Russell Dickenson**

Incorporated changes for JBoss Enterprise Application Platform 5.2.0 GA. For information about documentation changes to this guide, refer to *Release Notes 5.2.0*.

Revision 5.1.2-100 **Thu Dec 8 2011** **Jared Morgan**

Incorporated changes for JBoss Enterprise Application Platform 5.1.2 GA. For information about documentation changes to this guide, refer to *Release Notes 5.1.2*.

Revision 5.1.1-100 **Mon Jul 18 2011** **Jared Morgan**

Incorporated changes for JBoss Enterprise Application Platform 5.1.1 GA. For information about documentation changes to this guide, refer to *Release Notes 5.1.1*.

Revision 5.1.0-100 **Wed Sep 15 2010** **Laura Bailey, Joshua Wulf**

Incorporated changes for JBoss Enterprise Application Platform 5.1.0 GA. For information about documentation changes to this guide, refer to *Release Notes 5.1.0*.