JBoss Enterprise Application Platform Common Criteria Certification 5

Installation Guide

for use with JBoss Enterprise Application Platform 5 Common Criteria Certification Edition 5.1.0

Last Updated: 2017-11-20

JBoss Enterprise Application Platform Common Criteria Certification 5 Installation Guide

for use with JBoss Enterprise Application Platform 5 Common Criteria Certification
Edition 5.1.0

Isaac Rooskov

Laura Bailey

Joshua Wulf

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.

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 typo 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: *EAP 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 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.

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 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.

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**
  - **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.

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 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>
```

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>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.

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 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 *Admin Console Quickstart Guide*for more information on the Admin Console.

2.4. 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.

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 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.

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. UPGRADING FROM JBOSS ENTERPRISE APPLICATION PLATFORM 4.3 TO VERSION 5.1 VIA RPM

JBoss Enterprise Application 5 was a major release and includes major changes from JBoss Enterprise Application Platform 4.x, including a completely new Microcontainer and full compliance with the Java EE 5 standard. While Red Hat tries to maintain compatibility across releases, major releases provide us with the opportunity to advance our products in significant ways. Be aware that upgrading to a new version family (ie. 4.x to 5.x) will likely require changes to application configuration and application code.

An in-place upgrade from version 4.3 to 5.1 is available for customers who have installed the platform using RPM. This is a platform upgrade, not an assisted migration. The platform software will be updated to version 5.1, however, you will have to update configuration files and verify the compatibility of your applications. You should test and verify the entire process before applying it to a production system.

WARNING

Users on any release of JBoss Enterprise Application Platform 4.2, and users on any JBoss Enterprise Application Platform Feature Pack or Technology Preview, should *not* attempt to upgrade with this procedure.

The procedure for performing an RPM upgrade is as follows:

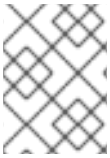
Procedure 3.1. Upgrading JBoss Enterprise Application Platform 4.3 to version 5.1 using RPM

1. Back up your environment
2. Upgrade the 4.3 install to the latest update level

NOTE

Upgrading from a 4.2 or 4.3 Tech Preview (TP) or Feature Pack (FP) is not supported.

3. Unsubscribe the system from the JBoss Enterprise Application Platform 4.3 channel in Red Hat Network
4. Subscribe the system to the JBoss Enterprise Application Platform 5.1 channel in Red Hat Network
5. Decide which of the optional components to install
6. Issue the upgrade command
7. Remove remaining 4.3 packages
8. Select Java 6 via alternatives
9. Update configuration files



NOTE

The final step, updating configuration files, is out of scope for this guide. Some guidance is given in Procedure 3.10, “Identify changed configuration files after the RPM upgrade” .

Procedure 3.2. Back up your environment

- Regular backups and the ability to restore a system to a state of known configuration are best practice at all times. You should back up your data and configuration and ensure that you are able to restore the system to a known state, before performing this procedure.

Procedure 3.3. Unsubscribe the system from the JBoss Enterprise Application Platform 4.3 channel in RHN

- Refer to the following Red Hat Knowledgebase article for instructions to modify a system's channel subscriptions: "How do I subscribe a system to a sub-channel or a child channel using Red Hat Network (RHN)?".

Procedure 3.4. Subscribe the system to the JBoss Enterprise Application Platform 5.1 channel in Red Hat Network (RHN)

- Refer to the following Red Hat Knowledgebase article for instructions to subscribe a system to a channel: "How do I subscribe a system to a sub-channel or a child channel using Red Hat Network (RHN)?".

  The channel names are as follows:

  Red Hat Network channel names for JBoss Enterprise Application Platform 5

  Red Hat Enterprise Linux 5 32-bit

  jbappplatform-5-i386-server-5-rpm

  Red Hat Enterprise Linux 5 64-bit

  jbappplatform-5-x86_64-server-5-rpm

  Red Hat Enterprise Linux 4 32-bit ES

  jbappplatform-5-i386-es-4-rpm

  Red Hat Enterprise Linux 4 32-bit AS

  jbappplatform-5-i386-as-4-rpm

  Red Hat Enterprise Linux 4 64-bit ES

  jbappplatform-5-x86_64-es-4-rpm

  Red Hat Enterprise Linux 4 64-bit AS

  jbappplatform-5-x86_64-as-4-rpm

Choose which optional components to install

JBoss Enterprise Application Platform 4.3 included JBoss WS Native as a web services provider. JBoss Enterprise Application Platform 5.1 includes both JBoss WS Native and JBoss WS CXF as web

services providers. When installing or upgrading to JBoss Enterprise Application Platform 5.1, you must select *one* of the two web services stacks provided. To switch web services stacks, you must reinstall the product.

In the following procedures *WS_CHOICE* denotes the web services provider you wish to install. Replace *WS_CHOICE* with either **jbossas-ws-native** for JBoss Native web services, or **jbossas-ws-cxf** for JBoss CXF web services.

Procedure 3.5. Issue the upgrade command on Red Hat Enterprise Linux 4

- Issue the following commands as root, substituting your choice for the optional *WS_CHOICE*:

```
up2date WS_CHOICE jbossas-messaging resteasy jboss-eap5-native
jboss-seam2
up2date -u
```

Procedure 3.6. Issue the upgrade command on Red Hat Enterprise Linux 5

- Issue the following commands as root, substituting your choice for the optional *WS_CHOICE*:

```
yum remove classpathx-jaf
yum install WS_CHOICE jbossas-messaging resteasy jboss-eap5-native
jboss-seam2
yum upgrade --disablerepo=rhel-i386-server-5
```

WARNING

It is important that you now perform the following procedure to remove remaining 4.3 packages. Since the system is no longer subscribed to the JBoss Enterprise Application Platform 4.3 channel, these packages will not receive security updates if they are left on the system.

Procedure 3.7. Remove remaining 4.3 packages on Red Hat Enterprise Linux 4

- Issue the following command as root:

  (Note that the command is split across two lines in this document for presentation purposes, but should be entered on a single line)

```
rpm -e berkeleydb jboss-profiler servletapi6 tomcat5-servlet-2.4-api
asm odmg jboss-seam-docs geronimo-j2ee-1.4-apis qdox jacorb ws-
commons-policy tanukiwrapper jboss-seam xml-commons-resolver
```

Procedure 3.8. Remove remaining 4.3 packages on Red Hat Enterprise Linux 5

- Issue the following command as root:

  (Note that the command is split across two lines in this document for presentation purposes, but should be entered on a single line)

```
yum remove bea-stax-api berkeleydb jboss-seam-docs tanukiwrapper asm
odmg jacorb bea-stax servletapi6 ws-commons-policy qdox jboss-
profiler jboss-seam geronimo-j2ee-1.4-apis
```

Procedure 3.9. Select Java 6 via alternatives

JBoss Enterprise Application Platform 5 requires a Java 6 run-time environment. Install and correctly configure a 1.6 JDK (Java Development Kit) or JRE (Java Runtime Environment) to use the Platform.

- Refer to Appendix C, *Installing a Java Development Kit on Red Hat Enterprise Linux* for information on installing and configuring a JDK for Red Hat Enterprise Linux 4 and 5.

Procedure 3.10. Identify changed configuration files after the RPM upgrade

During an RPM upgrade from version 4.3, 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.

Between versions 4.3 and 5.1 of the Enterprise Application Platform a significant number of changes have taken place. Identify the impact of these changes on your infrastructure and your applications before performing this upgrade on your production systems.

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

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

CHAPTER 4. NEW INSTALLATION

4.1. 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.

JBoss Enterprise Application Platform requires Java JDK1.6.

4.1.1. 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 4.1. 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 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/.

4.1.2. 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 5. INSTALLATION OPTIONS

5.1. 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 5.1.0.

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 Application 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.

5.2. 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.

5.3. 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 6, *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 7, *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 8, *Installation using the Graphical Installer*.

CHAPTER 6. ZIP INSTALLATION FROM THE RED HAT CUSTOMER PORTAL

Procedure 6.1. Installation via ZIP file

Follow this procedure to install JBoss Enterprise Application 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-eap-<release>.zip`** to extract the archive contents into the location of your choice.

   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.

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 **`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 Native Components**
   Refer to Chapter 9, *Install Native Components* for Native Component installation instructions.

5. **Perform post-installation configuration**
   Refer to Chapter 10, *Post Installation Configuration* for post-installation configuration instructions.

CHAPTER 7. RPM INSTALLATION VIA RED HAT NETWORK

7.1. 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.

7.2. INSTALLATION ON RED HAT ENTERPRISE LINUX 4

Procedure 7.1. 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

   32-bit AS

   jbappplatform-5-i386-as-4-rpm

   64-bit ES

   jbappplatform-5-x86_64-es-4-rpm

   64-bit AS

   jbappplatform-5-x86_64-as-4-rpm

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

   ```
   up2date jbossas-messaging WS_CHOICE jbossas
   up2date jboss-seam2 resteasy rh-eap-docs jboss-eap5-native
   ```

3. **Optional: Install Native Components**
   Refer to Chapter 9, *Install Native Components* for Native Component installation instructions.

4. **Perform post-installation configuration**

Refer to Chapter 10, *Post Installation Configuration* for post-installation configuration instructions.

7.3. INSTALL ON RED HAT ENTERPRISE LINUX 5

Procedure 7.2. 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

   64-bit

   jbappplatform-5-x86_64-server-5-rpm

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

   ```
   yum remove classpathx-jaf
   yum upgrade --disablerepo=rhel-i386-server-5
   yum install jbossas-messaging WS_CHOICE jbossas
   yum install jboss-seam2 resteasy rh-eap-docs jboss-eap5-native
   ```

3. **Optional: Install Native Components**
   Refer to Chapter 9, *Install Native Components* for Native Component installation instructions.

4. **Perform post-installation configuration**
   Refer to Chapter 10, *Post Installation Configuration* for post-installation configuration instructions.

CHAPTER 8. INSTALLATION USING THE GRAPHICAL INSTALLER

Procedure 8.1. 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 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
   ```

   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 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.

   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-5.0.1`**

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 5, *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 9, *Install Native Components* for Native Component installation instructions.

13. **Perform post-installation configuration**
    Refer to Chapter 10, *Post Installation Configuration* for post-installation configuration instructions.

CHAPTER 9. INSTALL 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 OpenSSL, JBoss Native, mod_jk, mod_cluster, NSAPI for Solaris, and ISAPI for Windows.

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 *Administration and Configuration Guide* for mod_jk, and the *Mod_cluster Guide* for mod_cluster.

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. It also features
- **`ISAPI`** is a connector for the Microsoft IIS web server.

9.1. 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-utils`**.

If the server is started without the **`apr`** and **`apr-utils`** 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-5.1/native/lib.
```

9.2. 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.

9.3. NATIVE COMPONENTS INSTALLATION

Procedure 9.1. Install Native Components

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 Section 5.3, "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-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 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/eapuser/jboss-eap-5.1/native/lib64
    -Djava.endorsed.dirs=/home/eapuser/jboss-eap-5.1/jboss-
   as/lib/endorsed
   ```

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

CHAPTER 10. POST INSTALLATION CONFIGURATION

10.1. 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 10.1, “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 10.1, “Create jmx-console, admin-console, and http invoker user account” to edit your user account, or create a new one.

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

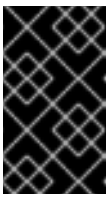
Procedure 10.1. Create jmx-console, admin-console, and http invoker user account

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

1. **Create a user in the default JAAS security domain**

   a. Edit the file **`$JBOSS_HOME/server/$PROFILE/conf/props/jmx-console-users.properties`**.

   b. Create a *username* = *password* pair.



IMPORTANT

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

2. **Grant permissions to user**

   a. Edit the file **`$JBOSS_HOME/server/$PROFILE/conf/props/jmx-console-roles.properties`**.

   b. Create an entry for the user of the form:

```
username=JBossAdmin,HttpInvoker
```

JBossAdmin

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

HttpInvoker

Grant the user permission to access the httpinvoker

10.1.2. Security Configuration: Web Console

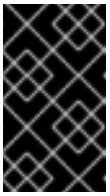
Procedure 10.2. Create web console user account

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/server/$PROFILE/deploy/management/console-mgr.sar/web-console.war/WEB-INF/classes/**.

   b. Create a *username* = *password* pair.



IMPORTANT

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

10.1.3. Security Configuration: JBoss Messaging

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

Procedure 10.3. Set suckerPassword for JBoss Messaging:

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

1. Edit the file **jboss-as/server/$PROFILE/deploy/messaging/messaging-jboss-beans.xml**.

2. Change the **suckerPassword** value.

10.2. DEFAULT DATABASE

WARNING

By default, persistence is configured to use Hypersonic (HSQLDB). This allows the JBoss Enterprise Application 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.

10.3. MEMORY SETTINGS FOR SEAM EXAMPLE APPS

If no other profile is specified at server startup, the **`default`** server profile is used. However, the **`production`** profile is recommended to run the example Seam applications included with JBoss Enterprise Application Platform.

To avoid memory issues, adjust the memory settings before deploying the applications. Do this by updating **`JAVA_OPTS`** settings in the file **`JBOSS_DIST/jboss-as/bin/run.conf`** (Linux) or **`JBOSS_DIST\jboss-as\bin\run.conf.bat`** (Windows) to match your application requirements. The default settings are:

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

10.4. 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 Enterprise Application Platform directory where the service installation script is located.**
   `cd` ***`JBOSS_DIST`*****`\native\sbin`**

3. **Optional: Edit `services.bat` to pass parameters to the Application Server at start-up.**
   Under `:cmdStart`, alter the following line:

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

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

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

4. **Run the service installation script.**
   ```
   service.bat install
   ```

5. **Check that the service is installed.**
   Under the Windows services list you will find this listed by the short name JBEAP5SVC and the long name JBoss EAP 5.

NOTE

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

CHAPTER 11. TEST YOUR INSTALLATION

Procedure 11.1. 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 Application Platform server homepage is displayed.

CHAPTER 12. UNINSTALL JBOSS ENTERPRISE APPLICATION PLATFORM

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

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

APPENDIX A. 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** 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
```

The following fragment should be commented out:

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

The following fragment should be commented out:

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

The following fragment should be commented out:

```
<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
```

The following fragment should be commented out:

```
<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**:

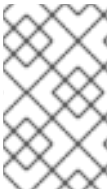
```
deploy/jbossws.sar/jbossws-management.war/WEB-INF/web.xml
```

The following fragment should be commented out:

```
<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>
```

APPENDIX B. 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 B.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 8, *Installation using the Graphical Installer* for Graphical Installer instructions.
- See Chapter 6, *ZIP Installation from the Red Hat Customer Portal* for ZIP installation instructions.
- See Chapter 9, *Install Native Components* for Native Component installation instructions.

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

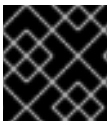
Red Hat supports the JBoss Enterprise Application Platform when it is run on **Red Hat Enterprise Linux** version 4 or 5 in conjunction with the **Sun Microsystems Java Development Kit** (JDK) version 1.6. The JBoss Enterprise Application Platform is also supported on Red Hat Enterprise Linux 5 when 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.

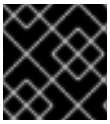
2. **Install the package.**
   To install the **Sun Microsystems Java Development Kit** package, input this command:

   ```
   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.

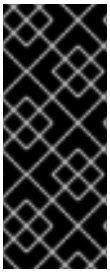
2. **Install using the up2date command.**
   Run this command to install the package:

   ```
   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	`/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 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	`/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 D. REVISION HISTORY

Revision 5.1.0-110.33.400 **2013-10-30** **Rüdiger Landmann**

Rebuild with publican 4.0.0

Revision 5.1.0-110.33 **July 24 2012** **Ruediger Landmann**

Rebuild for Publican 3.0

Revision 5.1-0 **Wed Sep 15 2010** **Laura Bailey, Joshua Wulf**

Changed version number - edition now denotes the revision.

Revised for JBoss Enterprise Application Platform 5.1.0.GA, including:

Added instructions for upgrading from 4.3 to 5.1 via Red Hat Network for systems that use the RPM installation method. Refer to Chapter 3, *Upgrading from JBoss Enterprise Application Platform 4.3 to version 5.1 via RPM*

Removed instructions for upgrading the platform using JBoss Operations Network. No distdiff patch is available for the Platform at this time.

JBPAPP-4575 - Added advice that Hypersonic is not suitable for production environments.

JBPAPP-4826 - Added instructions regarding upgrading the platform using JBoss Operations Network.

JBPAPP-3266 - No instructions on running as a service on Linux

JBPAPP-4849 - Corrected errors in service names.

JBPAPP-4875 - Various fixes

JBPAPP-4551 - Various installation path related fixes