



JBoss Enterprise Application Platform Common Criteria Certification 5 Common Criteria Configuration Guide

for use with JBoss Enterprise Application Platform 5 Common Criteria Certification
Edition 5.1.1

JBoss Enterprise Application Platform Common Criteria Certification 5 Common Criteria Configuration Guide

for use with JBoss Enterprise Application Platform 5 Common Criteria Certification
Edition 5.1.1

Jared Morgan
Red Hat Engineering Content Services
jmorgan [at] redhat [dot] com

Legal Notice

Copyright © 2011 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Abstract

Describes configuring and securing JBoss Enterprise Application Platform 5.1.0.GA and JBoss Enterprise Application Platform 5.1.1.GA to meet Common Criteria EAL4 certification.

Table of Contents

CHAPTER 1. INTRODUCTION	4
1.1. PURPOSE OF THIS DOCUMENT	4
1.2. WHAT IS A COMMON CRITERIA COMPLIANT SYSTEM?	4
1.3. CERTIFIED DOCUMENTATION	5
CHAPTER 2. REQUIREMENTS FOR THE EVALUATED CONFIGURATION	6
2.1. SOFTWARE REQUIREMENTS	6
2.1.1. Java Virtual Machine	6
2.1.2. Operating System	6
2.1.3. Database Servers	7
2.1.4. Database Servers	10
2.2. PHYSICAL REQUIREMENTS	13
2.3. PERSONNEL REQUIREMENTS	13
2.4. CONNECTIVITY REQUIREMENTS	13
2.4.1. Cluster Connectivity Requirements	14
2.5. CONFIGURATION REQUIREMENTS	14
2.5.1. General Restrictions	14
2.5.2. Setup Configuration	15
2.5.3. Configuring Audit Logging	16
2.5.4. Security Configuration	18
2.5.4.1. JBoss SX	18
2.5.4.2. JAAS Seam Configuration	18
2.5.4.3. Securing MBean Invokers	18
2.5.4.4. JBoss Web	19
2.5.4.5. EJB Authorization Policy	19
2.5.5. Java Security Manager Policy File	20
2.5.6. Database Configuration	20
2.5.7. Guidance on Configuring Java Security Permissions	23
2.5.8. Technology Preview Components	24
CHAPTER 3. DOWNLOADING AND VERIFYING THE PACKAGES	25
3.1. VERIFY THE AUTHENTICITY OF THE DOWNLOAD SITE	25
3.2. JBOSS NATIVE COMPONENTS SUPPORT	26
3.3. ZIP INSTALLATION	27
3.3.1. Download Zip	27
3.3.2. Verifying the Downloaded Files	28
3.3.3. Install Zip	31
3.3.4. Download Patches	32
3.4. ISO INSTALLATION	33
3.4.1. Download ISO	33
3.4.2. Verify ISO	34
3.4.3. Install ISO	34
3.5. CONFIRMING THE VERSION OF YOUR JBOSS ENTERPRISE APPLICATION PLATFORM INSTALLATION	37
CHAPTER 4. LAUNCHING THE JBOSS ENTERPRISE APPLICATION PLATFORM SERVER	40
4.1. STARTING THE JBOSS ENTERPRISE APPLICATION PLATFORM SERVER	40
4.2. ENABLING THE JAVA SECURITY MANAGER	40
4.2.1. Keystore Setup	41
4.2.1.1. Creating New Keystore with the JBoss Public Key	42
4.2.1.2. Using the Java System Keystore	42
4.2.1.3. IBM JRE 1.6 and the Java Security Manager	43
4.3. ADDITIONAL POLICY FILE CONFIGURATION	43

CHAPTER 5. DEVELOPMENT GUIDE FOR THE COMMON CRITERIA CERTIFIED SYSTEM	44
5.1. ENTERPRISE APPLICATION	44
5.2. GENERAL RESTRICTIONS	44
5.3. DEVELOPER ADVICE FOR USER CREDENTIALS IN REMOTE METHOD INVOCATION	46
CHAPTER 6. OVERVIEW OF THE SECURITY FUNCTIONS	47
6.1. ACCESS CONTROL	47
6.2. AUDIT	47
6.2.1. Enabling Additional Logging	50
6.3. CLUSTERING	50
6.4. IDENTIFICATION AND AUTHENTICATION	51
6.5. TRANSACTION ROLLBACK	52
APPENDIX A. PORT CONFIGURATION IN JBOSS ENTERPRISE APPLICATION PLATFORM	54
A.1. TCP SETTINGS	54
A.2. UDP SETTINGS	56
APPENDIX B. REVISION HISTORY	58

CHAPTER 1. INTRODUCTION

1.1. PURPOSE OF THIS DOCUMENT

This document is a guidance document for administrators and application developers who wish to use JBoss Enterprise Application Platform 5.1.0 GA and JBoss Enterprise Application Platform 5.1.1.GA in a certified, Common Criteria compliant, secure configuration.

This document is intended to be self-contained in addressing the most important issues at a high level, and refers to existing documentation where more details are needed. Knowledge of the Common Criteria is not required for readers of this document.

JBoss Enterprise Application Platform v5.1.0 GA and JBoss Enterprise Application Platform 5.1.1.GA are the subjects of this document as the Target of Evaluation (TOE) for Common Criteria certification. Both JBoss Enterprise Application Platform v5.1.0 GA and JBoss Enterprise Application Platform 5.1.1.GA have been evaluated under Common Criteria version 3.1 at level of assurance EAL4. This provides assurance that these products have been structurally tested.

All usages of the term "JBoss Enterprise Application Platform" from this point forward refer to the Common Criteria certified configurations of JBoss Enterprise Application Platform v5.1.0 GA or JBoss Enterprise Application Platform 5.1.1.GA. The exception to this is where specific configuration demands the version to be mentioned.

All usages of `JBOSS_HOME` in this user guide refer to the `/jboss-as` directory in the JBoss Enterprise Application Platform root installation directory. For example, if you used the ZIP installation package and extracted the JBoss Enterprise Application Platform binary to your Linux `/home` directory, `JBOSS_HOME` refers to the `home/[user]/jboss-eap-[version]/jboss-as/` directory. If you installed JBoss Enterprise Application Platform with the ISO distribution, `JBOSS_HOME` refers to `/var/lib/jbossas`.

This chapter contains a brief introduction to the CC certification & the structure of this book.

[Chapter 2, *Requirements for the Evaluated Configuration*](#) contains the requirements for deploying the certified product.

[Chapter 3, *Downloading and Verifying the Packages*](#) contains the steps that are required to ensure you are using the certified version of JBoss Enterprise Application Platform.

[Chapter 4, *Launching the JBoss Enterprise Application Platform Server*](#) provides instructions on how to start the server and the different modes of operation.

[Chapter 5, *Development Guide for the Common Criteria Certified System*](#) contains guidelines for developers creating applications for JBoss Enterprise Application Platform.

[Chapter 6, *Overview of the Security Functions*](#) contains the details of the security implementation and usage limitations of JBoss Enterprise Application Platform.

Should there be any discrepancy between information contained in this guide and any other product documentation, this guide takes precedence; it addresses the requirements for the evaluated configuration of JBoss Enterprise Application Platform.

1.2. WHAT IS A COMMON CRITERIA COMPLIANT SYSTEM?

The *Common Criteria for Information Technology Security Evaluation*, usually known as *Common Criteria* or *CC*, is an internationally-recognized standard (ISO/IEC 15408) used as the basis for independent evaluation of the security properties of an IT product.

Common Criteria provides consumers with an impartial security assurance of a product to predefined levels. These levels range from EAL1 to EAL7, each placing increased demands on the developer for evidence of testing, in turn providing increased assurance within the product for consumers.

Under the Common Criteria Recognition Arrangement (CCRA), members agree to recognize Common Criteria certificates that have been produced by any certificate authorizing participant, in accordance with the terms laid out in the CCRA. Currently, the CCRA is comprised of 22 member nations: Australia, Austria, Canada, the Czech Republic, Finland, France, Germany, Greece, Hungary, India, Israel, Italy, Japan, the Netherlands, New Zealand, Norway, the Republic of Singapore, Spain, Sweden, Turkey, the United Kingdom, and the United States. New members are expected to join in the near future.

A system can be considered to be *CC compliant* if it matches an evaluated and certified configuration. This implies various requirements concerning hardware and software, as well as requirements concerning the operating environment, users, and the ongoing operating procedures.

You can find further information on Common Criteria at [the Common Criteria Portal](#).

1.3. CERTIFIED DOCUMENTATION

When installing, configuring, and operating the JBoss Enterprise Application Platform in a Common Criteria evaluated configuration, you must only refer to the product documentation authorized for use with this Common Criteria certification.

The product documentation bundle is available in two certified formats:

- PDF documentation bundle
- on-line

You can download the PDF documentation bundle from the Customer Support Portal in addition to the installation binary. For more information about downloading the installation files, refer to [Chapter 3, Downloading and Verifying the Packages](#)

You can view the Common Criteria certified documentation on-line by visiting <https://docs.redhat.com>, and viewing the Common Criteria documentation section.

All references to JBoss enterprise user documentation in this guide refer to the guides contained in the certified formats.



WARNING

You *must not* refer to the standard on-line product documentation for JBoss Enterprise Application 5 maintenance versions when operating an evaluated configuration. The progressively updated on-line documentation versions may contain information that if followed could result in an evaluated configuration certification breach.

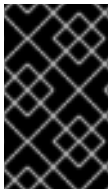
CHAPTER 2. REQUIREMENTS FOR THE EVALUATED CONFIGURATION

2.1. SOFTWARE REQUIREMENTS

2.1.1. Java Virtual Machine

JBoss Enterprise Application Platform is evaluated on the following Java Virtual Machines (JVMs). Only these JVMs are acceptable for the deployment of JBoss Enterprise Application Platform.

- Sun JRE 1.6.x
- IBM JRE 1.6.x
- OpenJDK JRE 1.6.x



IMPORTANT

We recommend to use JRE 1.6 Update 24 by Sun or OpenJDK to avoid possible DoS attacks as reported in CVE-2010-4476 (refer to <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-4476>).

2.1.2. Operating System

The supported operating systems for this evaluation are limited to the following products:

- Red Hat Enterprise Linux 6 x86
- Red Hat Enterprise Linux 6 x86-64
- Red Hat Enterprise Linux 5 x86
- Red Hat Enterprise Linux 5 x86-64
- Red Hat Enterprise Linux 4 x86
- Red Hat Enterprise Linux 4 x86-64
- Solaris 10 SPARC 64
- Solaris 10 x86-64
- Solaris 10 x86
- Solaris 9 SPARC (64-bit)
- Solaris 9 SPARC (32-bit)
- Solaris 9 x86
- Microsoft Windows Server 2008 x86-64
- Microsoft Windows Server 2008 x86

- Microsoft Windows Server 2003 x86-64
- Microsoft Windows Server 2003 x86


2.1.3. Database Servers

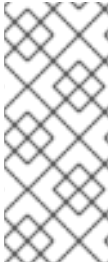
JBoss Enterprise Application Platform 5.1.0 is evaluated with the following relational database systems. Only these database systems with the specific driver versions are acceptable for use with JBoss Enterprise Application Platform 5.1.0.

Table 2.1. Allowed 5.1.0 Database and JDBC Driver Versions

Database	JDBC Driver
IBM DB2 9.7	IBM DB2 JDBC Universal Driver Architecture version 4.9.78 Driver download: http://www-947.ibm.com/support/entry/portal/Overview/Software/Information_Management/IBM_Data_Server_Client_Packages <pre>\$ sha256sum db2jcc4.jar 23e82e3e0474a8d914c7106293c47c47b2701270b76a78256a34da672f11f07b ./db2-97/jdbc4/db2jcc4.jar</pre>
Oracle 10g R2 (v10.2.0.4)	Oracle 10g R2 version 10.2.0.4 Driver download: http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html <pre>\$ sha256sum ojdbc14.jar 7ba80b6ee4f3433f88c8d878fb0dbc7d04fea736c2a6df8d34af1a4f970670a6 ./oracle10g/jdbc4/ojdbc14.jar</pre>
Oracle 11g R1 (v11.1.0.7.0)	Oracle 11g R1 version 11.2.0.1.0 Driver download: http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html <pre>\$ sha256sum ojdbc6.jar 0414637bc6876df9f611463e846ffe2753aaf2a29a32a822bb889be0d5494a1f ./oracle11gR1/jdbc4/ojdbc6.jar</pre>
Oracle 11g R1 RAC (v11.1.0.7.0)	Oracle 11g RAC version 11.2.0.1.0 Driver download: http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html <pre>\$ sha256sum ojdbc6.jar 0414637bc6876df9f611463e846ffe2753aaf2a29a32a822bb889be0d5494a1f ./oracle11gR1RAC/jdbc4/ojdbc6.jar</pre>

Database	JDBC Driver
Oracle 11g R2	<p>Oracle JDBC Driver version 11.2.0.1.0</p> <p>Driver download: http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html.</p> <pre>\$ sha256sum ojdbc6.jar 0414637bc6876df9f611463e846ffe2753aaf2a29a32a822bb889be0d5 494a1f ./oracle11gR2/jdbc4/ojdbc6.jar</pre>
Oracle 11g R2 RAC	<p>Oracle JDBC Driver version 11.2.0.1.0</p> <p>Driver download: http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html.</p> <pre>\$ sha256sum ojdbc6.jar 0414637bc6876df9f611463e846ffe2753aaf2a29a32a822bb889be0d5 494a1f ./oracle11gR2RAC/jdbc4/ojdbc6.jar</pre>
MySQL v5.0 (v5.0.79)	<p>MySQL Connector/J version 5.0.8</p> <p>Driver download: http://dev.mysql.com/downloads/connector/j/.</p> <pre>\$ sha256sum mysql-connector-java-5.0.8.zip 94c4eb6185ec2e8e9d6609903ad66c7b015ffb349cfe984c0538a1a02e 675538 ./mysql50/jdbc4/mysql-connector-java-5.0.8-bin.jar</pre>
MySQL v5.1 (v5.1.36)	<p>MySQL Connector/J version 5.1.13</p> <p>Driver download: http://dev.mysql.com/downloads/connector/j/.</p> <pre>\$ sha256sum mysql-connector-java-5.1.13.zip 90492c85fad7301740e2aa2c0c86d9473b04baab3cd2870f99f8955c56 9f671d ./mysql51/jdbc4/mysql-connector-java-5.1.13-bin.jar</pre>
Microsoft SQL Server 2005	<p>Microsoft SQL Server 2005 JDBC driver v3.0</p> <p>Driver download: http://www.microsoft.com/downloads/details.aspx?familyid=99B21B65-E98F-4A61-B811-19912601FDC9&displaylang=en.</p> <pre>\$ sha256sum sqljdbc4.jar 306170cb246935349121f854ddab0d70e30c66d511bda8fe173cd2f2b8 d1718b ./mssql2005/jdbc4/sqljdbc4.jar</pre>

Database	JDBC Driver
Microsoft SQL Server 2008	<p>Microsoft SQL Server JDBC Driver 3.0</p> <p>Driver download: http://www.microsoft.com/downloads/details.aspx?FamilyId=6D483869-816A-44CB-9787-A866235EFC7C&displaylang=en.</p> <pre>\$ sha256sum sqljdbc4.jar af3c54d8857ebbfdc34a9c5a51a7ee91e7dfa583205c81b6e0ee6208efe96e04 ./mssql2008/jdbc4/sqljdbc4.jar</pre> <div style="display: flex; align-items: flex-start;">  <div style="flex-grow: 1;"> <p>NOTE</p> <p>The JDBC 3.0 driver for MS SQL Server 2008 has changed elements of the date/time dialect which causes the <code><methodname>org.hibernate.test.hql.ASTParserLoadingTest</methodname></code> test and the <code><methodname>org.hibernate.test.stateless.StatelessSessionTest</methodname></code> tests to fail.</p> </div> </div>
PostgreSQL v8.2.17	<p>JDBC4 Postgresql Driver, version 8.2-510</p> <p>Driver download: http://jdbc.postgresql.org/download.</p> <pre>\$ sha256sum postgresql-8.2-510.jdbc4.jar 5c9e0c334b2d1dcda17c34a36309a331ebd62ae3104fddabcca69695fdc48c23 ./postgresql82/jdbc4/postgresql-8.2-510.jdbc4.jar</pre>
PostgreSQL v8.3.11	<p>JDBC4 Postgresql Driver, version 8.3-605</p> <p>Driver download: http://jdbc.postgresql.org/download.</p> <pre>\$ sha256sum postgresql-8.3-605.jdbc4.jar b007b5f90258ccf98346d51fcd4475bd7d0dc089492442dceb321baddb2bb777 ./postgresql83/jdbc4/postgresql-8.3-605.jdbc4.jar</pre>
Sybase ASE 15.0.3	<p>Sybase jConnect JDBC driver v7 (Build 26502)</p> <p>Driver download: http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.uprun.doc/doc/r0011932.htm</p> <pre>\$ sha256sum jconn4-26502.jar 44cec7a2dc3dfe9b968c1c12c8f06e0c2ad412da0c34fb6b2137805608a4442b ./sybase15/jdbc4/jconn4-26502.jar</pre>

**NOTE**

The `sha256sum` command line examples given are accurate for most Linux and Unix operating systems. Mac OS X includes the equivalent command `shasum -a 256`.

If you are using Microsoft Windows you must download a third party utility to perform these steps: Microsoft Windows does not include a SHA-256 SUM tool.

For information on how to configure each database with the JBoss Enterprise Application Platform 5.1.0, refer to [Section 2.5.6, “Database Configuration”](#).


2.1.4. Database Servers

JBoss Enterprise Application Platform 5.1.1 is evaluated with the following relational database systems. Only these database systems with the specific driver versions are acceptable for use with JBoss Enterprise Application Platform 5.1.1.

Table 2.2. Allowed 5.1.1 Database and JDBC Driver Versions

Database	JDBC Driver
IBM DB2 9.7	IBM DB2 JDBC Universal Driver Architecture version 4.12.55 Driver download: http://www-947.ibm.com/support/entry/portal/Overview/Software/Information_Management/IBM_Data_Server_Client_Packages <pre>\$ sha256sum db2jcc4.jar 23e82e3e0474a8d914c7106293c47c47b2701270b76a78256a34da672f 11f07b ./db2-97/jdbc4/db2jcc4.jar</pre>
Oracle 10g R2 (v10.2.0.4)	Oracle 10g R2 version 10.2.0.5 Driver download: http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html <pre>\$ sha256sum ojdbc14.jar 7ba80b6ee4f3433f88c8d878fb0dbc7d04fea736c2a6df8d34af1a4f97 0670a6 ./oracle10g/jdbc4/ojdbc14.jar</pre>
Oracle 11g R1 (v11.1.0.7.0)	Oracle 11g R1 version 11.1.0.7 Driver download: http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html <pre>\$ sha256sum ojdbc6.jar 0414637bc6876df9f611463e846ffe2753aaf2a29a32a822bb889be0d5 494a1f ./oracle11gR1/jdbc4/ojdbc6.jar</pre>

Database	JDBC Driver
Oracle 11g R1 RAC v(11.1.0.7.0)	<p>Oracle 11g RAC version 11.1.0.7</p> <p>Driver download: http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html.</p> <pre>\$ sha256sum ojdbc6.jar 0414637bc6876df9f611463e846ffe2753aaf2a29a32a822bb889be0d5 494a1f ./oracle11gR1RAC/jdbc4/ojdbc6.jar</pre>
Oracle 11g R2	<p>Oracle JDBC Driver version 11.2.0.2.0</p> <p>Driver download: http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html.</p> <pre>\$ sha256sum ojdbc6.jar 0414637bc6876df9f611463e846ffe2753aaf2a29a32a822bb889be0d5 494a1f ./oracle11gR2/jdbc4/ojdbc6.jar</pre>
Oracle 11g R2 RAC	<p>Oracle JDBC Driver version 11.2.0.1.0</p> <p>Driver download: http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html.</p> <pre>\$ sha256sum ojdbc6.jar 0414637bc6876df9f611463e846ffe2753aaf2a29a32a822bb889be0d5 494a1f ./oracle11gR2RAC/jdbc4/ojdbc6.jar</pre>
MySQL v5.0 (v5.0.79)	<p>MySQL Connector/J version 5.0.8</p> <p>Driver download: http://dev.mysql.com/downloads/connector/j/.</p> <pre>\$ sha256sum mysql-connector-java-5.0.8.zip 94c4eb6185ec2e8e9d6609903ad66c7b015ffb349cfe984c0538a1a02e 675538 ./mysql150/jdbc4/mysql-connector-java-5.0.8-bin.jar</pre>
MySQL v5.1 (v5.1.36)	<p>MySQL Connector/J 5.1.17</p> <p>Driver download: http://dev.mysql.com/downloads/connector/j/.</p> <pre>\$ sha256sum mysql-connector-java-5.1.17.zip 90492c85fad7301740e2aa2c0c86d9473b04baab3cd2870f99f8955c56 9f671d ./mysql151/jdbc4/mysql-connector-java-5.1.13-bin.jar</pre>

Database	JDBC Driver
Microsoft SQL Server 2005	<p>Microsoft SQL Server 2005 JDBC driver v3.0.1301.101</p> <p>Driver download: http://www.microsoft.com/downloads/details.aspx?familyid=99B21B65-E98F-4A61-B811-19912601FDC9&displaylang=en.</p> <pre>\$ sha256sum sqljdbc4.jar 306170cb246935349121f854ddab0d70e30c66d511bda8fe173cd2f2b8d1718b ./mssql2005/jdbc4/sqljdbc4.jar</pre>
Microsoft SQL Server 2008 R2	<p>Microsoft SQL Server JDBC Driver 3.0.1301.101</p> <p>Driver download: http://www.microsoft.com/downloads/details.aspx?FamilyId=6D483869-816A-44CB-9787-A866235EFC7C&displaylang=en.</p> <pre>\$ sha256sum sqljdbc4.jar af3c54d8857ebbfdc34a9c5a51a7ee91e7dfa583205c81b6e0ee6208efe96e04 ./mssql2008/jdbc4/sqljdbc4.jar</pre> <p> NOTE</p> <p>The JDBC 3.0 driver for MS SQL Server 2008 has changed elements of the date/time dialect which causes the <code><methodname>org.hibernate.test.hql.ASTParserLoadingTest</methodname></code> test and the <code><methodname>org.hibernate.test.stateless.StatelessSessionTest</methodname></code> tests to fail.</p>
PostgreSQL v8.2.17	<p>JDBC4 Postgresql Driver, version 8.2-511</p> <p>Driver download: http://jdbc.postgresql.org/download.</p> <pre>\$ sha256sum postgresql-8.2-510.jdbc4.jar 5c9e0c334b2d1dcda17c34a36309a331ebd62ae3104fddabcca69695fdc48c23 ./postgresql82/jdbc4/postgresql-8.2-510.jdbc4.jar</pre>
PostgreSQL v8.3.11	<p>JDBC4 Postgresql Driver, version 8.3-606</p> <p>Driver download: http://jdbc.postgresql.org/download.</p> <pre>\$ sha256sum postgresql-8.3-605.jdbc4.jar b007b5f90258ccf98346d51fcd4475bd7d0dc089492442dceb321baddb2bb777 ./postgresql83/jdbc4/postgresql-8.3-605.jdbc4.jar</pre>

Database	JDBC Driver
Sybase ASE 15.0.3	Sybase jConnect JDBC driver v7 (Build 26502/EBF17993) Driver download: http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.uprun.doc/doc/r0011932.htm <pre>\$ sha256sum jconn4-26502.jar 44cec7a2dc3dfe9b968c1c12c8f06e0c2ad412da0c34fb6b2137805608 a4442b ./sybase15/jdbc4/jconn4-26502.jar</pre>



NOTE

The `sha256sum` command line examples given are accurate for most Linux and Unix operating systems. Mac OS X includes the equivalent command `shasum -a 256`.

If you are using Microsoft Windows you must download a third party utility to perform these steps: Microsoft Windows does not include a SHA-256 SUM tool.

For information on how to configure each database with the JBoss Enterprise Application Platform 5.1.1, refer to [Section 2.5.6, “Database Configuration”](#).

2.2. PHYSICAL REQUIREMENTS

The hardware and software executing JBoss Enterprise Application Platform, as well as the software critical to security policy enforcement must be protected from unauthorized modification including unauthorized modifications by potentially hostile outsiders. Reasonable physical security measures to ensure that unauthorized personnel do not have physical access to the hardware running the JBoss Enterprise Application Platform software must be implemented.

2.3. PERSONNEL REQUIREMENTS

There must be one or more competent individuals who are assigned to manage JBoss Enterprise Application Platform, its environment and the security of the information it contains. The system administrative personnel must not be carelessly or willfully negligent, or hostile, and will follow and abide by the instructions provided by the administrator documentation.

The developer of user applications executed by JBoss Enterprise Application Platform, including web server applications and enterprise beans, shall be trustworthy and comply with all instructions set forth by the user guidance and evaluated configuration guidance of the JBoss Enterprise Application Platform.

2.4. CONNECTIVITY REQUIREMENTS

The operating system and the Java virtual machine operate according to their specification. These external systems shall be configured in accordance with this guidance.

Any other system with which JBoss Enterprise Application Platform communicates is assumed to be under the same management control and operate under the same security policy constraints as JBoss Enterprise Application Platform.

2.4.1. Cluster Connectivity Requirements

Your JBoss Enterprise Application Platform instances must operate in a network segment that is logically separated from any other network segment using a packet filtering mechanism. This packet filter must only allow incoming communication that meets the following criteria:

- the network protocol is TCP
- the destination port is 8080 or 8443

All outgoing communication from one of the JBoss Enterprise Application Platform instances must be allowed.

Each cluster node communicates with the other nodes by means of standard network sockets. Whenever this occurs the client side of each connection has a port number assigned to it by the host operating system from a range of ports that are reserved for client sockets. These ports are referred to as *dynamic* or *ephemeral* ports. They are only used by a connection until it is closed. Once the connection is closed the port is made available for use by other new client connections. Refer to your operating system documentation if you need to configure this port range.

2.5. CONFIGURATION REQUIREMENTS

The following sections describe modifications to be made to the **production** server configuration to comply with CC requirements. It is recommended, however, to back up the production configuration prior to making the changes shown in the following subsections.

Backing up the production configuration involves making a copy of the ***JBOSS_HOME/server/production*** directory. If you are using Microsoft Windows you can use Windows Explorer to make a copy of this folder using copy-paste and rename the copy to ***production.backup***.

Under UNIX or Linux you can execute the command:

```
cp -pr JBOSS_HOME/server/production JBOSS_HOME/server/production.backup
```

In an emergency you can always retrieve the original files from the installation files.

2.5.1. General Restrictions

The following general restrictions apply when setting up a certified configuration.

JBossWS

The WS CFX and WS Native stack are allowed. The WS Metro stack is not allowed in the evaluated configuration.

Management Consoles

The following deployed applications must be secured so they are accessible by trusted administrators only. The applications must be removed from the certified configuration if this condition is not met.

- JMX Console (***jmx-console.war***)

Location: ***jboss-as/server/production/deploy/jmx-console.war/***

- Web Console (`web-console.war`)

Location: `jboss-as/server/production/deploy/management/`

- Administration Console (`admin-console.war`)

Location: `jboss-as/server/production/deploy/admin-console.war/`

If you do not intend to use one or more of the management consoles, delete the entire directory related to each. Doing this guarantees that unauthorized users will not be able to access your system through an unused, and potentially unsecured management console.

2.5.2. Setup Configuration

Procedure 2.1. Evaluated Configuration Setup Configuration

The following configuration steps must be performed to ensure compliance with Common Criteria requirements.

1. Disable Simple Network Management Protocol (SNMP)

Delete the directory `JBOSS_HOME/server/production/deploy/snmp-adaptor.sar`

```
$ rm -rf JBOSS_HOME/server/production/deploy/snmp-adaptor.sar
```

2. Disable Remote Method Invocation (RMI) under the Internet Inter-ORB Protocol (IIOP)

To disable RMI/IIOP delete following files:

- `JBOSS_HOME/server/production/conf/jacorb.properties`
- `JBOSS_HOME/server/production/deploy/iiop-service.xml`
- `JBOSS_HOME/server/production/lib/jacorb.jar`

```
$ rm JBOSS_HOME/server/production/conf/jacorb.properties
$ rm JBOSS_HOME/server/production/deploy/iiop-service.xml
$ rm JBOSS_HOME/server/production/lib/jacorb.jar
```

3. Disable AJP from JBoss Web.

Comment out the following section from

`JBOSS_HOME/server/production/deploy/jbossweb.sar/server.xml`:

```
<Connector protocol="AJP/1.3" port="8009"
address="{jboss.bind.address}" redirectPort="8443" />
```

4. Disable Clustering High-Availability JNDI service (port 1102)

To disable clustering HA, do the following:

- Delete the file `JBOSS_HOME/server/production/deploy/cluster/hajndi-jboss-beans.xml`

```
rm JBOSS_HOME/server/production/deploy/cluster/hajndi-jboss-beans.xml
```

- b. Disable the HA Naming service interface via HTTP by commenting out following <mbean> definition in `JBOSS_HOME/server/production/deploy/httpa-invoker.sar/META-INF/jboss-service.xml`:

```
<mbean code="org.jboss.invocation.http.server.HttpProxyFactory"
name="jboss:service=invoker,type=http,target=HAJNDI">
```

5. Enable Password Hashing

Use password hashing and do not store plain text passwords on the server.



NOTE

For more information regarding configuring password hashing, refer to the *Password Hashing* section in the *JBoss Security Guide*

6. Disable Technology Preview Components

Ensure Technology Preview components are disabled.

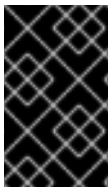


NOTE

[Section 2.5.8, “Technology Preview Components”](#) details the components shipped with JBoss Enterprise Application Platform that must be disabled, and how to disable them.

2.5.3. Configuring Audit Logging

Audit logging can be configured to print authentication and authorization information for each thread and EJB call.



IMPORTANT

Logging individual requests is a resource intensive activity. Test the impact this will have on your server and application performance before enabling this level of logging on a production server.

Procedure 2.2. Monitor Server Startup and Shutdown Events

Enable server startup and shutdown events by making the recommended changes to `JBOSS_HOME/server/production/conf/jboss-log4j.xml`

1. Uncomment Security Audit Appender

Uncomment the following block.

```
<!-- Security AUDIT Appender -->
<appender name="AUDIT"
class="org.jboss.logging.appender.DailyRollingFileAppender">
  <errorHandler
class="org.jboss.logging.util.OnlyOnceErrorHandler"/>
  <param name="File" value="${jboss.server.log.dir}/audit.log"/>
  <param name="Append" value="true"/>
  <param name="DatePattern" value="'.'yyyy-MM-dd"/>
  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%d %-5p [%c] (%t:%x)
```

```
%m%n" />
</layout>
</appender>
```

2. Uncomment Security Audit Provider

Uncomment the following block:

```
<!-- Category specifically for Security Audit Provider -->
<category name="org.jboss.security.audit.providers.LogAuditProvider"
additivity="false">
  <priority value="TRACE"/>
  <appender-ref ref="AUDIT"/>
</category>
```

3. Configure SecurityInterceptor logging level

Set the logging level of the `SecurityInterceptor` class to `TRACE` by adding the `<priority>` element to the root `<category>` element.

```
<category name="org.jboss.ejb.plugins.SecurityInterceptor">
  <priority value="TRACE" />
  <appender-ref ref="AUDIT" />
</category>
```

4. Enable logging for ServerImpl log messages

Set the priority and appender-ref levels for the Microcontainer bootstrap by adding the `<category>` block as specified.

```
<category name="org.jboss.bootstrap.microcontainer">
  <priority value="INFO"/>
  <appender-ref ref="AUDIT"/>
</category>
```

5. Enable logging for web-based requests

If you need additional logging for web-based requests, uncomment the `AccessLogValve` in `JBOSS_HOME/server/production/deploy/jbossweb.sar/server.xml`.

```
<Valve className="org.apache.catalina.valves.AccessLogValve"
  prefix="localhost_access_log." suffix=".log"
  pattern="common" directory="{jboss.server.home.dir}/log"
  resolveHosts="false" />
```

The access log is saved in the `log` directory of the server configuration.

6. Update ConversionPattern

Update the `ConversionPattern` parameter in the appender/layout element to show thread information by replacing the Default Pattern with the Full Pattern:

```
<!--The full pattern: Date MS Priority [Category] (Thread:NDC)
Message -->
<param name="ConversionPattern" value="%d %-5r %-5p [%c] (%t:%x)
%m%n" />
```

2.5.4. Security Configuration

The following configuration steps must be performed to ensure security compliance with Common Criteria requirements.

2.5.4.1. JBoss SX

Custom Login Modules are not permitted; the only login modules allowed are the following:

- `org.jboss.security.auth.spi.UsersRolesLoginModule`
- `org.jboss.security.auth.spi.LdapLoginModule`
- `org.jboss.security.auth.spi.DatabaseServerLoginModule`
- `org.jboss.security.auth.spi.BaseCertLoginModule`

This restriction on login modules is also applicable to the `DynamicLoginConfig` service.

Only the following security managers are allowed to be configured and used for authentication purposes:

- `org.jboss.security.plugins.JaasSecurityManager`
- `org.jboss.security.plugins.JaasSecurityDomain`

Additional security-related modules that are permitted are the following:

- `org.jboss.security.authorization.modules.DelegatingAuthorizationModule`
- `org.jboss.security.integration.JNDIBasedSecurityRegistration`
- `org.jboss.security.auth.certs.SubjectDNMapping`

Other modules, such as SRP module are not allowed.

2.5.4.2. JAAS Seam Configuration

For JBoss Seam, the simplified JAAS configuration provided by the Seam Security API must not be used. The default system JAAS configuration must be used, and configured according to the *Advanced Authentication Features* section of the *Seam Reference Guide*

Using the default system JAAS configuration ensures user identification and authentication are performed by the JBoss Enterprise Application Server.

JBoss Seam provides additional interfaces for implementing other security functions such as authorization (for example, entity bean permissions). This functionality is controlled by JBoss Seam, and is therefore outside the scope of the evaluated product.

2.5.4.3. Securing MBean Invokers

The `httpa-invoker.sar` found in the deploy directory is a service that provides RMI/HTTP access for EJBs and the JNDI Naming service. This includes a servlet that processes posts of `marshaled org.jboss.invocation.Invocation` objects that represent invocations that should be dispatched onto the `MBeanServer`. Effectively this allows access to MBeans that support the detached invoker operation via HTTP when sending appropriately formatted HTTP posts. This servlet has to be

protected against the use by unprivileged users. To secure this access point you need to secure the `JMXInvokerServlet` servlet found in the `httpa-invoker.sar/invoker.war/WEB-INF/web.xml` descriptor.

To prevent attacks with other HTTP methods (for example, with the HEAD method), locate `JBOSS_HOME/server/production/deploy/httpa-invoker.sar/invoker.war/WEB-INF/web.xml` (deployment descriptor of the `JMXInvokerServlet` servlet) and remove the following lines from the `web-app/security-constraint/web-resource-collection` node:

```
<http-method>GET</http-method>
<http-method>POST</http-method>
```

The `jmx-invoker-service.xml` is a service that exposes the JMX MBeanServer interface via an RMI compatible interface using the RMI/JRMP detached invoker service. This interface has to be made unavailable to unprivileged users which can be done by using the `org.jboss.jmx.connector.invoker.AuthenticationInterceptor` interceptor for performing identification and authentication using JAAS. Additionally, access control has to be configured using the interceptors of either `org.jboss.jmx.connector.invoker.RolesAuthorization` or `org.jboss.jmx.connector.invoker.ExternalizableRolesAuthorization`.



NOTE

For more information regarding securing the MBean invokers, refer to the *How to Secure the JBoss Server* chapter in the *Security Guide*.

2.5.4.4. JBoss Web

The JAAS based authentication and authorization realm implementation (`org.jboss.web.tomcat.security.JBossWebRealm`) cannot be replaced. The same is true for the authenticator classes defined for each authentication method (BASIC, CLIENT-CERT, DIGEST, FORM, NONE) in `JBOSS_HOME/server/production/deployers/jbossweb.deployer/META-INF/war-deployers-jboss-beans.xml`.

Additionally, the `allRolesMode` within `JBOSS_HOME/server/production/deploy/jbossweb.sar/server.xml` must be set to `strict`. This requires the authenticated user to be assigned to one of the `web-app/security-role/role-name` roles in order to be authorized.

```
<Realm className="org.jboss.web.tomcat.security.JBossWebRealm"
  certificatePrincipal="org.jboss.security.auth.certs.SubjectDNMapping"
  allRolesMode="strict" />
```

2.5.4.5. EJB Authorization Policy

Applications can implement custom authentication and authorization verification using a JACC Authorization Module. In Enterprise Application Platform 5, the JACC authorization module forms part of a JAAS security domain.

When configuring your application specific security policy, you must declare one (or more) of the following authorization modules in the security domain `<policy-module>` element.

- `org.jboss.security.authorization.modules.DelegatingAuthorizationModule`

- `org.jboss.security.authorization.modules.JACCAuthorizationModule`

By declaring these authorization modules, your application will reference the configuration in the `security-policies-jboss-beans.xml` and extend the configuration using the settings in the application-specific security domain contained in the EJB.

If you are using the XACML Authorization Module, you must also use either the JACC Authorization Module or the Delegating Authorization Module to ensure the application policy you configure uses the settings configured in the `security-policies-jboss-beans.xml` file.

For specific information relating to configuring the `JACCAuthorizationModule` or `DelegatingAuthorizationModule`, refer to the *Authentication* chapter in the *Security Guide*.

2.5.5. Java Security Manager Policy File

To operate JBoss Enterprise Application Platform according to the requirements of the certification, you must do the following to ensure applications running on the system have the correct access privileges:

- Install the `jbossas-security-policy-cc` package.
- Configure the Java Security Manager to use the policy file.

Correctly installing the `jbossas-security-policy-cc` package is covered as part of the installation procedures in [Chapter 3, Downloading and Verifying the Packages](#). The `jbossas-security-policy-cc` package provides the `security_cc.policy` file, which is installed in the `JBOSS_HOME/bin/` directory.

The security manager policy file for the common criteria evaluated configuration can require additions of permissions that are needed for database drivers to function for user applications. The system administrator can assign permissions to the database drivers that are needed by user applications. It is recommended that the most restrictive permissions are added

You must define security access permissions for the database

For security reasons, you must manually specify the policy file in the `run.conf` (Linux) or `run.conf.bat` (Windows) file. For complete instructions on configuring the JSM to use the `security_cc.policy`, refer to the *Using the Security Manager* section in the *Security Guide*.

2.5.6. Database Configuration

The default database HSQLDB that the Enterprise Application Platform ships with must be disabled as it is not supported. Additional configuration is also required for JDBC drivers and supporting. This section will outline how this can be done and then refer you to information on how to configure supported databases. This must be done in the `production` server profile.

Procedure 2.3. Configure Database

1. **Create DefaultDS file**

Create a default DS file for the desired database. Examples of this file are located in `JBOSS_HOME/docs/examples/jca`.



IMPORTANT

A `DefaultDS` file must be supplied in the `JBOSS_HOME/server/production/deploy` directory.

2. Delete HSQLDB files

Delete the following files as they refer to the HSQLDB database:

- o `JBOSS_HOME/server/production/deploy/hsqldb-ds.xml`
- o `JBOSS_HOME/common/lib/hsqldb.jar`
- o `JBOSS_HOME/common/lib/hsqldb-plugin.jar`
- o `JBOSS_HOME/server/production/deploy/messaging/hsqldb-persistence-service.xml`

3. Remove HSQLDB Security Domain

Comment out the security domain for `HsqlDbRealm` in the `JBOSS_HOME/server/production/conf/login-config.xml` file as shown.

```

<!-- Security domains for testing new jca framework
<application-policy name = "HsqlDbRealm">
  <authentication>
    <login-module>
      <code =
"org.jboss.resource.security.ConfiguredIdentityLoginModule"
      <flag = "required">
      <module-option name = "principal">sa</module-option>
      <module-option name = "userName">cctest</module-option>
      <module-option name = "password">cc1248</module-option>
      <module-option name = "managedConnectionFactoryName">
        jboss.jca:service=LocalTxCM,name=DefaultDS
      </module-option>
    </login-module>
  </authentication>
</application-policy>
-->

```

4. Copy persistence service configuration file

The `[database]-persistence-service.xml` file contains the persistence service definition for JBoss Messaging, for the database specified by the `[database]` in the filename.

Copy the `[database]-persistence-service.xml` file that corresponds to the database you are using from the `JBOSS_HOME/docs/examples/jms` directory to `JBOSS_HOME/server/production/deploy`.



NOTE

The table definitions in any `[database]-persistence-service.xml` are not optimized for performance.

5. Relocate JDBC driver libraries

Place the supported JDBC driver libraries in the directory `JBOSS_HOME/server/production/lib/`.



IMPORTANT

Ensure you follow the policy guidelines in [Section 2.5.5, “Java Security Manager Policy File”](#) and choose a supported JDBC driver from [Table 2.1, “Allowed 5.1.0 Database and JDBC Driver Versions”](#) or [Table 2.2, “Allowed 5.1.1 Database and JDBC Driver Versions”](#) to maintain an evaluated configuration.

6. Add JDBC Grant Statement

Add the following grant statement for the JDBC driver you are using to the Java Security Manager policy file. The policy file is located in `JBOSS_HOME/bin/security_cc.policy`. Substitute the directory name of the JDBC driver where `[cc.jdbc.driver]` is specified in the code sample.



IMPORTANT

Each JDBC driver can use different permissions. Check the JDBC driver documentation and replace `java.security.AllPermission;` with a secure permission scheme supported by the driver.

```
// granting permissions to JDBC driver
grant codeBase "file:${jboss.server.home.dir}/lib/[cc.jdbc.driver]"
{
    permission java.security.AllPermission;
};
```

7. Oracle Database Persistence Plugin Optimization

When using the Oracle Database, the database persistence plugin definition must be changed in `JBOSS_HOME/server/production/deploy/ejb2-timer-service.xml` from being:

```
<attribute name="DatabasePersistencePlugin">
org.jboss.ejb.txtimer.GeneralPurposeDatabasePersistencePlugin
</attribute>
```

to being:

```
<attribute name="DatabasePersistencePlugin">
org.jboss.ejb.txtimer.OracleDatabasePersistencePlugin
</attribute>
```

**NOTE**

JBoss Enterprise Application Platform requires a database to store its operational state. The JNDI name referring to the database is `java:/DefaultDS`. The database has to be separated from all application databases: user applications must not provide additional tables to the `java:/DefaultDS` database, but must use their own dedicated databases to store their objects.

This setup prevents attacks with SQL injection through user applications and information leaks from `java:/DefaultDS` database, as such injections are always limited to the connected database.

**NOTE**

The *Installation and Configuration Guide* contains specific information about the supported databases, and their configuration. Read this information in conjunction with the *Common Criteria Configuration Guide* overrides to ensure you maintain an evaluated configuration.

2.5.7. Guidance on Configuring Java Security Permissions

The system administrator for the operation of the certified system is expected to configure the security permissions for all enterprise applications that are deployed on the certified system, when the certified system runs in the security manager enabled mode.

**WARNING**

In addition to the General Restrictions listed in [Chapter 5, Development Guide for the Common Criteria Certified System](#) the following permissions *must not be granted* to any application in order to maintain a certified configuration:

- file permissions, except to files that are dedicated to the application
- network permissions
- permissions to load native code.

**IMPORTANT**

You must not assign a `java.security.AllPermission` (or equivalent for your JDBC driver) to any of the user applications interacting with the certified system.

User Applications must not be granted any other runtime, or socket permissions

Refer to the Java documentation for information on configuring permissions in the JVM:

- Java 1.6:
<http://download.oracle.com/javase/6/docs/technotes/guides/security/permissions.html>

A single entry in the Java Security Manager policy shipped with the certified system follows the standard Java Standard Edition model. More information is provided in the Java documentation:

- Java 1.6: <http://download.oracle.com/javase/6/docs/technotes/guides/security/PolicyFiles.html>

An example would be the following:

```
grant codeBase "file:${jboss.server.home.dir}/deploy/jmx-console.war/-" {  
    permission java.security.AllPermission;  
};
```

This is defined by the certified system by default to provide all permissions to the `jmx-console` web application that ships with JBoss Enterprise Application Platform in the `/deploy` directory.

So if the administrator needs to provide permissions to an enterprise application called as `TestDeployment.ear` in the `deploy` directory of the certified system, then an example entry would be the following:

```
grant codeBase "file:${jboss.server.home.dir}/deploy/jmx-console.war/-" {  
    permission java.util.PropertyPermission "*", "read";  
    permission javax.security.auth.AuthPermission  
    "createLoginContext.a_login";  
    permission javax.security.auth.AuthPermission "getLoginConfiguration";  
};
```

This entry provides the enterprise application called as `TestDeployment.ear` to read Java properties as well as the ability to create JAAS login context and obtain JAAS login configuration.

The certified system in the security manager enabled mode is a locked down system that forces the system administrator to configure the necessary security permissions for the operation of the user applications on the certified system.

Any interaction with the JBoss JMX Kernel (which is the standard Java MBeanServer) will require the appropriate `javax.management.MBeanPermission` as specified in the Java MBeanServer interface:

- Java 1.6: <http://java.sun.com/javase/6/docs/api/javax/management/MBeanServer.html>

2.5.8. Technology Preview Components

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. You must disable these components to operate in a Common Criteria certified configuration. The list of Technology Preview components include:

PicketLink

A security and identity framework that provides support for Security Assertion Markup Language (SAML) v2.0 and WS-Trust.

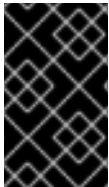
The PicketLink technology preview folder is included in the root folder, at the same level as the `jboss-as` folder. If you leave the `picketlink` folder as originally shipped in the EAP binary, PicketLink is unable to be launched, and subsequently interact with the certified configuration.

CHAPTER 3. DOWNLOADING AND VERIFYING THE PACKAGES

JBoss Enterprise Application Platform is delivered on-line through the Red Hat Customer Portal (RHCP) at <https://access.redhat.com> JBoss Enterprise Application Platform is available as ZIP files from the RHCP.

RPM installation files for JBoss Enterprise Application Platform are delivered through the Red Hat Network at <https://rhn.redhat.com>

To guarantee authenticity of the downloaded software, verify the authenticity of the files and their source.



IMPORTANT

Unless specifically stated otherwise, the screen shots and other samples shown in this section are examples only. The actual presentation of the download websites may change over time.

3.1. VERIFY THE AUTHENTICITY OF THE DOWNLOAD SITE

Red Hat Customer Portal and Red Hat Network are both secure sites. This is indicated by the 'security padlock' icon in the browser status bar. The padlock may also present itself in the address bar depending on what browser you are using.

If the 'security padlock' not visible, check the authenticity of the site by viewing the identity certificate.

Procedure 3.1. Checking Site Security with Firefox

1. In the address bar, click the security area at the beginning of the URL.
2. From the pop-up box, click **More Information**.
3. From the Page Info window, click **View Certificate**.
4. The certificate displays details such as who the owner of the page is, who issued the certificate, when it was issued and when it expires as well as fingerprint verification strings.

An example of the certificate for <https://rhn.redhat.com> is shown in [Figure 3.1, “The RHN SSL Certificate”](#).



General **Details**

This certificate has been verified for the following uses:

SSL Server Certificate

Issued To

Common Name (CN)	rhn.redhat.com
Organization (O)	Red Hat Inc
Organizational Unit (OU)	Red Hat Production Operations
Serial Number	0B:D3:6A

Issued By

Common Name (CN)	<Not Part Of Certificate>
Organization (O)	Equifax
Organizational Unit (OU)	Equifax Secure Certificate Authority

Validity

Issued On	07/06/2009
Expires On	09/06/2011

Fingerprints

SHA1 Fingerprint	1C:89:74:1F:41:B4:5F:C8:AC:BF:53:9D:88:FC:BF:80:BF:92:12:AE
MD5 Fingerprint	47:E7:2C:E7:45:63:79:72:40:A4:26:2A:99:DA:84:4E

Figure 3.1. The RHN SSL Certificate

If neither of the lock icons are present in your browser and a verified certificate cannot be found, you may not be connected to the correct site. If you are unable to reach the secure Red Hat Customer Portal, contact Red Hat Support and report this problem.

3.2. JBOSS NATIVE COMPONENTS SUPPORT

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.



IMPORTANT

Solaris and Windows Native Components are not part of the evaluated configuration. While Solaris and Windows Native components are shipped with the binaries, only Linux Native has been verified as part of the EAL4 certification level.



IMPORTANT

OpenSSL native components are supported for Red Hat Enterprise Linux 4, Red Hat Enterprise Linux 5, and Red Hat Enterprise Linux 6.

If you want to use Native Components in your evaluated configuration, refer to the installation instructions contained in this chapter.

For more information on installing Linux Native components, refer to the *Install Native Components* section in the *Installation Guide*.

3.3. ZIP INSTALLATION

You can install JBoss Enterprise Application Platform from a ZIP archive. The ZIP binaries are available through the Red Hat Customer Portal (RHCP) at <https://access.redhat.com>.

3.3.1. Download Zip

You must first download the Zip binary from <https://access.redhat.com>

Procedure 3.2. 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. In the Downloads menu, click on **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.

6. Select either 5.1.0 or 5.1.1 from the Version drop-down box.

Result:

You are presented with all versions of the selected platform.

7. **Select the correct software**

For 5.1.0, click the **Application Platform 5.1.0 Binary** link (not the download link).

For 5.1.1, click the **Application Platform 5.1.1 Binary** link (not the download link).

Result:

You are presented with the Software Details screen, with SHA-256 information.

8. On the displayed **Software Details** page, click the **Download** located next to the **File** drop-down box.

**NOTE**

Record the SHA-256 sum from the Software Details screen. You use this checksum to verify the authenticity of the download in [Section 3.3.2, “Verifying the Downloaded Files”](#)

9. Continue to [Procedure 3.3, “Installation using ZIP file”](#)

3.3.2. Verifying the Downloaded Files

Use the `sha256sum` utility to calculate the checksum values of the files to compare to the supplied values on the website. The checksum values are also documented in [Table 3.1, “5.1.0 SHA-256 checksum values”](#) and [Table 3.2, “5.1.1 SHA-256 checksum values”](#) for completeness.

**NOTE**

The command line examples given are accurate for most Linux and Unix operating systems. Mac OS X includes the equivalent command `shasum -a 256`.

If you are using Microsoft Windows you will have to download a third party utility to perform these steps as it does not include a SHA-256 SUM tool.

After you have downloaded the file, run the SHA-256 utility and specify the file you downloaded as the first argument as demonstrated here:

Example 3.1. Using the `sha256sum` tool on Linux or Unix

This example assumes you have chosen to use JBoss Enterprise Application Platform 5.1.0.GA, and have saved the file to the default `Download` directory. If you saved the file to another location, or are using JBoss Enterprise Application Platform 5.1.1.GA, navigate to the correct directory and execute the `sha256sum` command in that directory.

```
[Download]$ sha256sum jboss-eap-5.1.0.zip
74d1e2cd49739f3c66fb0139de33a667 jboss-eap-5.1.0.zip
```

The values generated by the `sha256sum` utility must match the values displayed on the Downloads page for the file, and those documented in [Table 3.1, “5.1.0 SHA-256 checksum values”](#) or [Table 3.2, “5.1.1 SHA-256 checksum values”](#). If they are not the same, your download is either incomplete or corrupted. You will need to download it again. If after several attempts you are unable to download a copy of the file that produces a valid checksum values you should open a support case to report the problem.

The complete list of SHA-256 checksums for JBoss Enterprise Application Platform 5.1.0.GA is listed in [Table 3.1, “5.1.0 SHA-256 checksum values”](#).

The complete list of SHA-256 checksums for JBoss Enterprise Application Platform 5.1.1.GA is listed in [Table 3.2, “5.1.1 SHA-256 checksum values”](#).

Table 3.1. 5.1.0 SHA-256 checksum values

File	SHA-256 Checksum
jbossws-cxf-3.1.2.SP7-src-dist.zip	44f7dce413fb462fe983ab7716f27abe34 9e6570575a890334c1ce9fc477e0a5
jboss-ep-ws-cxf-5.1.0-installer.zip	8e8aab592d9d02d855819b47740a6573 7f2f9b7ab4d5ceffc324a1043423d9a8
jboss-ep-native-5.1.0-RHEL6-x86_64.zip	eee724ccd622101e57abd7151578b852 5a29ee5e5f03202caaeace664b0e8a58
jboss-ep-native-5.1.0-RHEL6-i386.zip	66a4d3a6b101e28dcb328bb7b804cc4a 7914d54646219f818d32409da0850e52
jboss-ep-native-5.1.0-RHEL5-x86_64.zip	1a3b341f592cda120e3dd7df8bd704466 e62ca082f533799fa20acc021e59ae8
jboss-ep-native-5.1.0-RHEL5-i386.zip	5e3434354b2ed7ab5211e0e1bbe4160c1 c5e1714f2cca5d7ffafd4457e5c29af
jboss-ep-native-5.1.0-RHEL4-x86_64.zip	946a7b27b30a2b5edaae3d92630743fb8 8a37fde3d14b4d7ec93be5a540065b7
jboss-ep-native-5.1.0-RHEL4-i386.zip	09c145fa1e9a2cbb504a3543e46efd66e ce3dbbbd5a0ce7a531009d6085b385
jboss-eap-installer-5.1.0.jar	cfb52030cbb4a7ed08ec1a06e13940630 3cc7a6116c5efbae3cac2ef205431ab
jboss-eap-5.1.0.zip	d1edcf783b1095ea61c08a70d5d646183 8023290359c2911972c43e4263664e4
JBPAPP-5367.zip	b19d407b079350bd0ac30c42af94a8863 378c588eebcc0dfa50296166424368f
JBPAPP-5386.zip	c42ab845202389947b17036452b6260d a845a890f68c4086295e052339647f30
RHEL5.5-JBEAP-5-20110106.0-i386-disc1-ftp.iso	e75bb1712854c815cf693cb2ad111d88d 4c97b2790d25a8330dbb21bec68d346
RHEL5.5-JBEAP-5-20110106.0-x86_64-disc1-ftp.iso	729dcdd4cf3520cef274b501bf05627f88 6fdf8f3eaedf23ccf0a97b36914161
JBEAP5-re20110105.3-i386-disc1-ftp.iso	632bfe680de1a8c41ad5c5dafcfb3e1e7a6 ac23efc30452ee97de9475c162e01
JBEAP5-re20110105.3-variant-src-disc1.iso	efa3e5d6bf561dbcc1c0fee225be5c31d23 d50a130ec2c406daa3a968746e279

File	SHA-256 Checksum
JBEAP5-re20110105.3-x86_64-disc1-ftp.iso	694b14dc0336a9ccd56cb5297f38a3e32 8 4e7c2b411b2238108709c4e13b10e3

Table 3.2. 5.1.1 SHA-256 checksum values

File	SHA-256 Checksum
jboss-eap-5.1.1.zip	a36152d7790631142579da4a3f72a5d 25a8fc509266d6bc7bb65664ad9fe7e01
jboss-eap-installer-5.1.1.jar	a48168fd52f19c40bfedb03eb7b9caca3 aaca56361d801ca4a83d832da0a859d
jboss-ep-native-5.1.1-RHEL4-i386.zip	269dbd16118886493056083cfc2d288 5bbb69f051037b9cf15883115467e99c7
jboss-ep-native-5.1.1-RHEL4-x86_64.zip	dc9b3b75abddf566caab9d60a5edb923 cf993b033a2696e3bb8a0d80e7cb4549
jboss-ep-native-5.1.1-RHEL5-i386.zip	6353d73ece9ece8a4f6400d77fb86fb63 36a8e2c9898c76d61481df3a2ea99e5
jboss-ep-native-5.1.1-RHEL5-x86_64.zip	58757206257e54238db355064cf8d9a4 6e43ce7b0ce962368d051c6ee644c723
jboss-ep-native-5.1.1-RHEL6-i386.zip	2aac5e5c6ad7b0d42b0a578f936a7fda9 5360c0d36f7fa45c1ca070b01b496e7
jboss-ep-native-5.1.1-RHEL6-x86_64.zip	b4cb563ef8bcd67189befd6447a08698 cbd920332ed736520107d41afc1a9ad0
jboss-ep-ws-cxf-5.1.1-installer.zip	60969ff6d02f9ad7a52a6dff8abc7db89 dfb99b38ecdf9329881a22392d06e37
JBPAPP-5367.zip	b19d407b079350bd0ac30c42af94a8863 378c588eebcc0dfa50296166424368f
JBEAP5-re20120206.0-i386-disc1-ftp.iso	d73b1581d9f6bdf563c5efdd9bed5eab 61ed627e80a33b30d2617cb9d48cbfca
JBEAP5-re20120206.0-x86_64-disc1-ftp.iso	caaf5221a0e406c2ec4515c864889a00 995ffd619ce10ad3c14938b803fc9848
RHEL-5-JBEAP-5-20120206.0-i386-disc1-ftp.iso	5efb04bb525cf6ca0841db3d8a72bb6e 7383d2761e5bbe8ce342a56357ccac7

File	SHA-256 Checksum
RHEL-5-JBEAP-5-20120206.0-x86_64-disc1-ftp.iso	e177d3e3071741317c9cb916b569464 6d987145ea20e7348a2dfe9df27bb86a6
JB-EAP-5-RHEL-6-20120206.0-Server-i386-disc1-ftp.iso	9f41998258030c307e7c7ab1faedb4e3 d535ec4e13493e9eac1b8eac1f9a3460
JB-EAP-5-RHEL-6-20120206.0-Server-x86_64-disc1-ftp.iso	2532121c107341668e2c131f84526b1 3add669ab6e10bbdae1e20d0704e22525

3.3.3. Install Zip

In [Section 3.3.2, “Verifying the Downloaded Files”](#) you verified the integrity of the Zip archive. You can now install the Zip binary on your system.

Procedure 3.3. Installation using ZIP file

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

1. Unzip file `jboss-eap-[version].zip` to extract the archive contents into the location of your choice.

Result:

This creates the `jboss-eap-[version]` directory, with an un-configured installation of JBoss Enterprise Application Platform.

2. Optional: Download and Install Native Components

Native components are downloaded separate to the primary application binary.

- a. Choose a Native Components binary that corresponds to one of the evaluated Operating Systems listed in [Section 2.1.2, “Operating System”](#).
- b. Click the name of the Native Component binary to display the SHA-256 checksum.
- c. Repeat the process in [Section 3.3.2, “Verifying the Downloaded Files”](#) to verify the file's authenticity.
- d. Unzip the downloaded file to JBoss installation directory (such as `jboss-eap-5.1.0`) as `run.sh` can locate the components automatically when starting the Enterprise Application Server.
- e. Run the following `rm` command to remove the `https` directory and retain the compatibility with Common Criteria:

```
# rm -rf JBOSS_INSTALL_DIR/jboss-ep-[version]/native/JBOSS_NATIVE_LIB/httpd
```

Substitute `JBOSS_INSTALL_DIR` with JBoss installation directory, `[version]` with the product version you chose, and `JBOSS_NATIVE_LIB` with the library directory (`lib` or `lib64`).

**NOTE**

Only Red Hat Enterprise Linux 4, Red Hat Enterprise Linux 5 and Red Hat Enterprise Linux 6 JBoss Native Components are certified and supported in Common Criteria certified configurations.

3. Install Security Patches

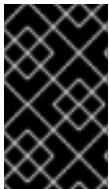
Follow the procedures in [Section 3.3.4, “Download Patches”](#) to correctly download the required security patches for the evaluated configuration and follow the patch installation instructions inside individual ZIP files (**README.txt**).

3.3.4. Download Patches

The Java Security Manager (JSM) requires a Common Criteria tested and certified policy file to control how applications interact with the application server under an evaluated configuration.

Procedure 3.4. Install required security patches

Follow this procedure to install required security patches.

**IMPORTANT**

Ensure you verify the authenticity of the files you download in the following procedure by verifying the SHA-256 checksums according to [Section 3.3.2, “Verifying the Downloaded Files”](#)

1. Open <http://access.redhat.com> in a web browser.
2. Hover over the **Downloads** option in the menu across the top of the page.
3. In the Downloads menu, click **JBoss Enterprise Middleware**
4. Enter your login information.

Result:

You are taken to the Software Downloads page.

5. Select **Application Platform** in the drop-down box or in the menu on the left.

Result:

You are presented with a list of file downloads.

6. Select 5.1.0 or 5.1.1 from the Version drop-down box, depending on the platform version you want to install.
7. For 5.1.1, download the JBoss Remoting 2.5.3SP1 security advisory:

- a. Click the **Security Advisories** tab label.

Result:

You are presented with the list of available security advisories.

- b. Click the **Apply jboss remoting 2.5.3SP1 fix** link.

- c. On the displayed **Software Details** tab, click the **Download** link next to the **File** drop-down box and confirm the download of the **JBPAPP-5386.zip** file.

**NOTE**

Record the SHA-256 checksum from the Software Details screen. You use this checksum to verify the authenticity of the download in [Section 3.3.2, “Verifying the Downloaded Files”](#)

8. Download **security_cc.policy** to EAP 5.1:

- a. Click the **Patches** tab label.

Result:

You are presented with the list of available security advisories.

- b. Click the **Add security_cc.policy to EAP 5.1x** link.
- c. On the displayed **Software Details** tab, click the **Download** link next to the **File** drop-down box and confirm the download of the **JBPAPP-5367.zip** file.

**NOTE**

Record the SHA-256 sum from the Software Details screen. You use this checksum to verify the authenticity of the download in [Section 3.3.2, “Verifying the Downloaded Files”](#)

3.4. ISO INSTALLATION

ISO installation files for JBoss Enterprise Application Platform are delivered through the Red Hat Network at <https://rhn.redhat.com>. The ISO files contain all security and patch errata.

3.4.1. Download ISO

Follow [Procedure 3.5, “Download the JBoss Enterprise Application Platform ISO”](#) to download the ISO for the evaluated configuration from Red Hat Network.

Procedure 3.5. Download the JBoss Enterprise Application Platform ISO

This procedure downloads files needed to install JBoss Enterprise Application Platform from an ISO.

You must have an entitlement to access the ISO images. Contact the Red Hat Customer Center for subscription management and customer support if you can not complete the procedure.

1. Open <http://access.redhat.com> in a web browser.
2. Log in using your customer portal credentials.
3. Hover the mouse over the **Download** menu in the top menu bar.
4. Under the Red Hat Enterprise Linux group, click **Downloads**.

Result:

The Download Software page opens, which lists all ISO images available for Red Hat products.

5. In the list of available software, expand the node for the Red Hat Enterprise Linux system supported by the certification.



NOTE

If you are not sure which node to select, refer to [Section 2.1.2, “Operating System”](#) for the supported Red Hat operating systems, and [Table 3.1, “5.1.0 SHA-256 checksum values”](#) and [Table 3.2, “5.1.1 SHA-256 checksum values”](#) for the list of supported ISO images.

6. Click the **JBoss Application Platform (v 5)** group.

Result:

You are redirected to the ISO image download page.

7. Verify the ISO is correct for your system by visually comparing the SHA-256 checksum with the checksum declared in [Table 3.1, “5.1.0 SHA-256 checksum values”](#) or [Table 3.2, “5.1.1 SHA-256 checksum values”](#)



NOTE

The name of the ISO link is not the same as the actual ISO filename. Use the tables specified to verify you are selecting the correct ISO image.

8. Record the corresponding SHA-256 sum for the ISO download. You use this checksum to verify the authenticity of the ISO in the next procedure.
9. Click the ISO link.

Result:

The download will begin. When the download has finished, you have successfully downloaded the correct ISO for the evaluated configuration.

3.4.2. Verify ISO

You must verify the authenticity of the ISO by checking the SHA-256 sum you recorded in [Procedure 3.5, “Download the JBoss Enterprise Application Platform ISO”](#) with the sum associated with the ISO.

Follow the guidelines in [Section 3.3.2, “Verifying the Downloaded Files”](#)

3.4.3. Install ISO

The version of Red Hat Enterprise Linux you run in your production environment determines what procedure to follow to install the ISO:

- Red Hat Enterprise Linux 4 users must follow [Procedure 3.6, “Install ISO on Red Hat Enterprise Linux 4”](#).
- Red Hat Enterprise Linux 5 users must follow [Procedure 3.7, “Install ISO on Red Hat Enterprise Linux 5”](#).

- Red Hat Enterprise Linux 6 users must follow [Procedure 3.8, “Install ISO on Red Hat Enterprise Linux 6”](#)

All ISO images contain the relevant security errata and patches for the evaluated configuration. You do not need to install any other errata when you choose the ISO installation method.

Procedure 3.6. Install ISO on Red Hat Enterprise Linux 4



IMPORTANT

You must activate superuser privileges to install the ISO image.

1. Mount ISO Image

Mount the ISO image downloaded in [Procedure 3.5, “Download the JBoss Enterprise Application Platform ISO”](#) to `/mnt/jboss`.

```
[root ~]# mkdir /mnt/jboss
[root ~]# mount -o loop PATH_TO_ISO_IMAGE /mnt/jboss
```

2. Create Local up2date Repository

Create a local repository for the ISO to retrieve packages from during installation.

Add the following line to `/etc/sysconfig/rhn/sources`.

```
echo "dir jbosslocal /mnt/jboss/RedHat/RPMS/" >>
/etc/sysconfig/rhn/sources
```

3. Install JBoss Enterprise Application Platform

Run the following `up2date` command, replacing `WS_CHOICE` with `jbossas-ws-native` or `jbossas-ws-cxf`

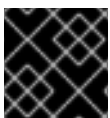
```
[root ~]# up2date -i jbossas-messaging WS_CHOICE jbossas jbossas-
security-policy-cc
```

4. Optionally install Native Components

Run the following `up2date` command to install certified and supported native components

```
[root ~]# up2date jboss-eap5-native
```

Procedure 3.7. Install ISO on Red Hat Enterprise Linux 5



IMPORTANT

You must activate superuser privileges to install the ISO image.

1. Mount ISO Image

Mount the ISO image downloaded in [Procedure 3.5, “Download the JBoss Enterprise Application Platform ISO”](#) to `/mnt/jboss`.

```
[root ~]# mkdir /mnt/jboss
[root ~]# mount -o loop PATH_TO_ISO_IMAGE /mnt/jboss
```

2. Create Repository

Create a local repository for the ISO to retrieve packages from during installation.

Create a file named `jbosslocal.repo` in `/etc/yum.repos.d/`.

```
[root ~]# cat << EOF > /etc/yum.repos.d/jbosslocal.repo
[jbosslocal]
name=jbosslocal
baseurl=file:///mnt/jboss/JBEAP
enabled=1
gpgcheck=0
EOF
```

3. Install JBoss Enterprise Application Platform

Run the following `yum` command, replacing `WS_CHOICE` with `jbossas-ws-native` or `jbossas-ws-cxf`:

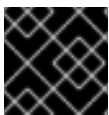
```
[root ~]# yum install jbossas-messaging WS_CHOICE jbossas jbossas-
security-policy-cc
```

4. Optionally install Native Components

Run the following `yum install` command to install certified and supported native components:

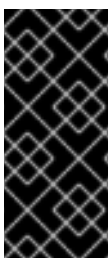
```
[root ~]# yum install jboss-eap5-native
```

Procedure 3.8. Install ISO on Red Hat Enterprise Linux 6



IMPORTANT

You must activate superuser privileges to install the ISO image.



IMPORTANT

This procedure is only relevant for Red Hat Enterprise Linux 6 and JBoss Enterprise Application Platform 5.1.1.

JBoss Enterprise Application Platform 5.1.0 ISOs are not compatible with Red Hat Enterprise Linux 6.

1. Mount ISO Image

Mount the ISO image downloaded in [Procedure 3.5, “Download the JBoss Enterprise Application Platform ISO”](#) to `/mnt/jboss`.

```
[root ~]# mkdir /mnt/jboss
[root ~]# mount -o loop PATH_TO_ISO_IMAGE /mnt/jboss
```

2. Create Repository

Create a local repository for the ISO to retrieve packages from during installation.

Create a file named `jbosslocal.repo` in `/etc/yum.repos.d/`.

```
[root ~]# cat << EOF > /etc/yum.repos.d/jbosslocal.repo
[jbosslocal]
name=jbosslocal
baseurl=file:///mnt/jboss
enabled=1
gpgcheck=0
EOF
```

3. Install JBoss Enterprise Application Platform

Run the following `yum` command, replacing `WS_CHOICE` with `jbossas-ws-native` or `jbossas-ws-cxf`

```
[root ~]# yum install jbossas-messaging WS_CHOICE jbossas jbossas-
security-policy-cc
```

4. Optionally install Native Components

Run the following `yum install` command to install certified and supported native components

```
[root ~]# yum install jboss-eap5-native
```

3.5. CONFIRMING THE VERSION OF YOUR JBOSS ENTERPRISE APPLICATION PLATFORM INSTALLATION

There are three ways you can verify the version number of your JBoss Enterprise Application Platform installation.

Using the `-V` with the startup script

You can retrieve information about the version of your JBoss Enterprise Application Platform installation by running the same script used to start the server with the `-V` switch. For Linux and Unix installations this script is `run.sh` and on Microsoft Windows installations it is `run.bat`. Regardless of platform the script is located in `JBOSS_HOME/bin`. Using these scripts to start your server is covered in [Chapter 4, Launching the JBoss Enterprise Application Platform Server](#)

Running this script with the `-V` switch will not start the server, nor does it require the server to be running. It displays information about the JBoss Enterprise Application Platform version and its configured Java environment. Below is an example of using this on an installation of JBoss Enterprise Application Platform on Red Hat Linux. Note the version number (**JBoss 5.1.0.GA** or **JBoss 5.1.1.GA**) displayed as the last item before the license information.

The screen below shows output for JBoss Enterprise Application Platform 5.1.0.GA. The output will be similar for JBoss Enterprise Application Platform 5.1.1.GA

```
[bin]$ ./run.sh -V
```

```
=====
JBoss Bootstrap Environment
```

```
JBOSS_HOME: /opt/JBoss/EnterprisePlatform-5.1.0/jboss-eap-5.1/jboss-as
```

```
JAVA: /usr/lib/jvm/jre-1.6.0-openjdk/bin/java
```

```
JAVA_OPTS: -Dprogram.name=run.sh -server -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.net.preferIPv4Stack=true
```

```
CLASSPATH: /opt/JBoss/EnterprisePlatform-5.1.0/jboss-eap-5.1/jboss-
as/bin/run.jar
```

```
=====
JBoss 5.1.0 (build: SVNTag=JBPAPP_5_1_0 date=201009150028)
```

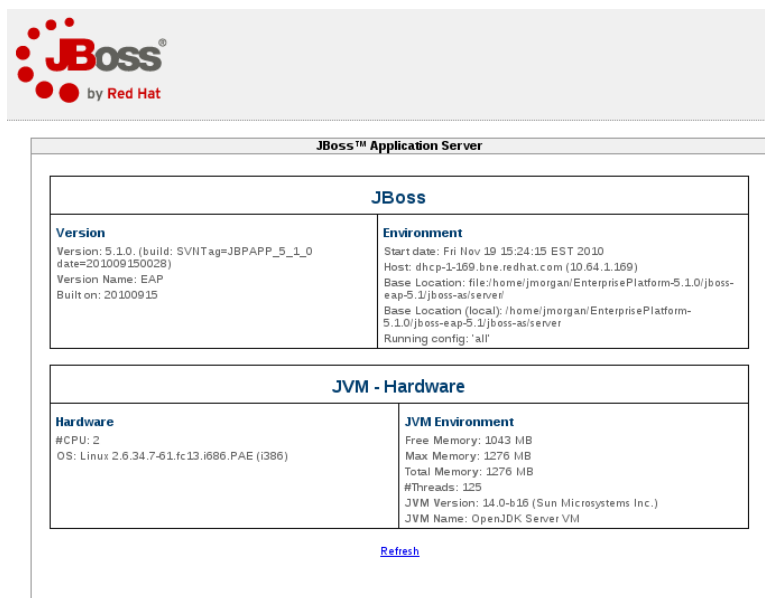
```
Distributable under LGPL license.
```

```
See terms of license at gnu.org.
```

```
[bin]$
```

Using the Web console

When the JBoss Enterprise Application Platform server is running you can retrieve its version information from the first page of the Web Console. This is located at <http://localhost:8080/web-console/>.



The screenshot shows the JBoss Web Console interface. At the top left is the JBoss logo with 'by Red Hat' underneath. The main content area is titled 'JBoss™ Application Server' and contains two sections: 'JBoss' and 'JVM - Hardware'. The 'JBoss' section is divided into 'Version' and 'Environment' sub-sections. The 'Version' sub-section lists: Version: 5.1.0 (build: SVNTag=JBPAPP_5_1_0 date=201009150028), Version Name: EAP, and Built on: 20100915. The 'Environment' sub-section lists: Start date: Fri Nov 19 15:24:15 EST 2010, Host: dhcp-1-169.bne.redhat.com (10.64.1.169), Base Location: file:/home/jmorgan/EnterprisePlatform-5.1.0/jboss-eap-5.1/jboss-as/server, Base Location (local): /home/jmorgan/EnterprisePlatform-5.1.0/jboss-eap-5.1/jboss-as/server, and Running config: 'all'. The 'JVM - Hardware' section is divided into 'Hardware' and 'JVM Environment' sub-sections. The 'Hardware' sub-section lists: #CPU: 2 and OS: Linux 2.6.34.7-61.fc13.i686.PAE (i386). The 'JVM Environment' sub-section lists: Free Memory: 1043 MB, Max Memory: 1276 MB, Total Memory: 1276 MB, #Threads: 125, JVM Version: 14.0-b16 (Sun Microsystems Inc.), and JVM Name: OpenJDK Server VM. A 'Refresh' link is located at the bottom of the 'JVM - Hardware' section.

Figure 3.2. Version details displayed in Web Console

View the console output, or boot.log file

When the server is started, the version is echoed to the console, and written to `JBOSS_HOME/server/production/log/boot.log`:

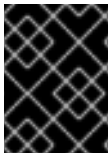
```
15:36:39,026 INFO [ServerImpl] Starting JBoss (Microcontainer)...  
15:36:39,027 INFO [ServerImpl] Release ID: JBoss [EAP] 5.1.0  
(build: SVNTag=JBPAPP_5_1_0 date=201009150028)
```

CHAPTER 4. LAUNCHING THE JBOSS ENTERPRISE APPLICATION PLATFORM SERVER

JBoss Enterprise Application Platform includes a startup script for both Linux/Unix platforms & Microsoft Windows as well a configuration file, `run.conf`, which determines the startup environment of the server. For Linux and Unix installations the startup script is `run.sh` and on Microsoft Windows installations it is `run.bat`. Regardless of platform the script is located in `JBOSS_HOME/bin`.

JBoss Enterprise Application Platform has been certified with the use of the Java Security Manager. You must use the policy settings as defined in the JBoss CC Security Policy file, which is located in `JBOSS_HOME/bin/security_cc.policy`.

Follow the guidance in [Section 4.2, “Enabling the Java Security Manager”](#) and [Section 2.5.5, “Java Security Manager Policy File”](#).



IMPORTANT

Operating JBoss Enterprise Application Platform using the Java Security Manager and different policy settings is not considered to be a certified configuration.

4.1. STARTING THE JBOSS ENTERPRISE APPLICATION PLATFORM SERVER

To start the JBoss Enterprise Application Platform server, use the start up script that is appropriate for your platform. You must use the `-c` command parameter to specify the **production** server configuration.

Example 4.1. Starting the server on Unix or Linux

```
[home]$ cd JBOSS_HOME/bin
[bin]$ ./run.sh -c production
```

Example 4.2. Starting the server on Windows

```
cd JBOSS_HOME/bin
$ run.bat -c production
```

4.2. ENABLING THE JAVA SECURITY MANAGER

Enabling the Java Security Manager (JSM) with the specified policy ensures JBoss Enterprise Application Platform remains protected from any deployed application accidentally or intentionally interfering with its operation.

The policy limits granting full permissions to those jar files included with the evaluated configuration.

**WARNING**

You must configure the policy settings as explained in [Section 2.5.5, “Java Security Manager Policy File”](#). Operating JBoss Enterprise Application Platform using the JSM with different policy settings is not considered to be a certified configuration.

To enable the JSM, you must edit the `run.conf` (Linux) or `run.conf.bat` (Windows) file, located in the `JBOSS_HOME/bin/` directory.

**NOTE**

Read the *Java Security Manager* chapter in the *JBoss Security Guide* for complete instructions regarding JSM activation and configuration. Refer back to the *Common Criteria Configuration Guide* for certification-specific overrides.

Enabling the Java Security Manager

To enable JSM for JBoss EAP add following lines to `run.conf`:

- Add this line to enable JSM and set its policy:

```
JAVA_OPTS="$JAVA_OPTS -Djava.security.manager -
Djava.security.policy==$JBOSS_HOME/bin/security_cc.policy"
```

- Add this line to set Java system properties which are referred by the added security policy:

```
JAVA_OPTS="$JAVA_OPTS -Djboss.home.dir=$JBOSS_HOME -
Djboss.server.home.dir=$JBOSS_HOME/server/production -
Djava.protocol.handler.pkgs=org.jboss.handlers.stub"
```

**IMPORTANT**

Make sure to add the lines exactly as shown including the double equal sign (this orders JSM to use only this policy without combining it with the system policy).

- Add this line to ensure the security policy persists when an RPM installation is stopped and restarted :

```
export JBOSS_HOME=/var/lib/jbossas
```

4.2.1. Keystore Setup

Because the security policy uses jar file signatures, you need to enable a keystore, which will hold JBoss public keys for signature validation and permission granting to JBoss provided code.

You can create your keystore with a JBoss public key (refer to [Section 4.2.1.1, “Creating New Keystore with the JBoss Public Key”](#)) or use the Java System keystore (refer to [Section 4.2.1.2, “Using the Java System Keystore”](#))

4.2.1.1. Creating New Keystore with the JBoss Public Key

Follow this procedure to create a keystore with the JBoss public key:

1. Run the following command to create keystore that contains the JBoss public key:

```
keytool -importcert -alias jboss -keystore
JBOSS_HOME/server/production/cc.keystore \
-storepass jbosseap -file JBOSS_HOME/bin/JBossPublicKey.RSA -
noprompt \
-trustcacerts
```



NOTE

The `jboss` alias must end up in `trustedCertEntry`. You can check the result with the following `keytool` command:

```
keytool -list -keystore
JBOSS_HOME/server/production/cc.keystore -storepass
jbosseap
```

2. Run this command to create the password file:

```
echo jbosseap > JBOSS_HOME/server/production/cc.password
```

Password file is a plain file with the password for key store opening (`cc.keystore`).

3. Uncomment lines number 6 and 7 of the `JBOSS_HOME/bin/security_cc.policy` file to enable the keystore:

```
keystore "file:${jboss.server.home.dir}/cc.keystore";
keystorePasswordURL "file:${jboss.server.home.dir}/cc.password";
```



NOTE

The password `jbosseap` used in Step 1 during key store creation must be the same as the password in the `cc.password` file. We highly recommend you use a password different than the example password in this guide.

4.2.1.2. Using the Java System Keystore

Follow this procedure to use the Java System keystore:

Run the following command to modify you Java system keystore:

```
keytool -importcert -alias jboss -keystore
JAVA_HOME/jre/lib/security/cacerts \
-storepass changeit -file JBOSS_HOME/bin/JBossPublicKey.RSA -noprompt \
```

```
-trustcacerts
```

Make sure you are running the command as a user with write permission for the `$JAVA_HOME` directory. The default password for the cacerts keystore is **changeit**.



IMPORTANT

Every change to the Java runtime JBoss public key must be added to cacerts keystore.

4.2.1.3. IBM JRE 1.6 and the Java Security Manager

IBM JRE 1.6 uses a default policy provider which does not work correctly with the JBoss Enterprise Application Platform security policy. You must change the JRE configuration to use the standard policy provider if you want to use IBM JRE 1.6 to host JBoss Enterprise Application Platform with the Java Security Manager enabled.

You do this by editing the file `JAVA_HOME/jre/lib/security/java.security` and setting the value of `policy.provider` to `sun.security.provider.PolicyFile` instead of `org.apache.harmony.security.fortress.DefaultPolicy`:

```
policy.provider=sun.security.provider.PolicyFile
```

4.3. ADDITIONAL POLICY FILE CONFIGURATION

Users and administrators may add their own permission blocks to the policy file, however the permissions specified for JBoss Enterprise Application Platform must not be changed; doing so will invalidate the certification.

Any modifications to the security policy that are not documented in this guide will invalidate the certification configuration. Refer to [Section 2.5.7, “Guidance on Configuring Java Security Permissions”](#) for additional information on this topic.

CHAPTER 5. DEVELOPMENT GUIDE FOR THE COMMON CRITERIA CERTIFIED SYSTEM

Read this section to understand the guidelines trusted developers must follow when developing programs or applications that run on the secure certified system.

5.1. ENTERPRISE APPLICATION

An enterprise application is a Java Enterprise Edition (formerly J2EE) version 1.5 compliant application software. Typically the application accepts requests from clients, does some processing and responds with results. The enterprise application that is developed by the trusted developer is hereby referred to as a *user application*.

The types of enterprise applications include the following:

1. Web Applications based on Servlets and Java Server Pages (JSP)
2. Enterprise Java Beans (EJB)
3. JavaEE 1.4 Web Service Applications which can be based on Stateless EJBs or Plain Old Java Objects (POJOs) deployed as Java Servlets.

5.2. GENERAL RESTRICTIONS

The trusted software developer must follow the following restrictions when developing secure software for the certified system.

1. Application Programming Interfaces (APIs) that are not documented in the applicable product documentation *must not be used*. For more information about providing security permissions to user applications, refer to <the guidance section for System administrators to configure the certified system>
2. The programming restrictions mandated by the *Enterprise JavaBeans Specification v2.1* must be strictly followed. For more information, refer to [JSR-000153 Enterprise JavaBeans 2.1 specification](#). (Section 25.2, pages 562-564).

Enterprise Java Beans Specification Developer Restrictions

The restrictions are:

- An enterprise bean must not use read/write static fields. Using read-only static fields is allowed. Therefore, it is recommended that all static fields in the enterprise bean class be declared as final.
- An enterprise bean must not use thread synchronization primitives to synchronize execution of multiple instances.
- An enterprise bean must not use the AWT functionality to attempt to output information to a display or to input information from a keyboard.
- An enterprise bean must not use the `java.io` package to attempt to access files and directories in the file system.
- An enterprise bean must not attempt to listen on a socket, accept connections on a socket, or use a socket for multicast.

- The enterprise bean must not attempt to query a class to obtain information about the declared members that are not otherwise accessible to the enterprise bean because of the security rules of the Java language. The enterprise bean must not attempt to use the Reflection API to access information that the security rules of the Java programming language make unavailable.
- The enterprise bean must not attempt to
 - create a class loader
 - obtain the current class loader
 - set the context class loader
 - set security manager
 - create a new security manager
 - stop the JVM
 - or change the input, output, and error streams
- The enterprise bean must not attempt to set the socket factory used by `ServerSocket`, `Socket`, or the stream handler factory used by `URL`.
- The enterprise bean must not attempt to manage threads. The enterprise bean must not attempt to start, stop, suspend, or resume a thread, or to change a thread's priority or name. The enterprise bean must not attempt to manage thread groups.
- The enterprise bean must not attempt to obtain the security policy information for a particular code source.
- The enterprise bean must not attempt to load a native library.
- The enterprise bean must not attempt to gain access to packages and classes that the usual rules of the Java programming language make unavailable to the enterprise bean.
- The enterprise bean must not attempt to define a class in a package.
- The enterprise bean must not attempt to access or modify the security configuration objects (`Policy`, `Security`, `Provider`, `Signer`, and `Identity`).
- The enterprise bean must not attempt to use the subclass and object substitution features of the Java Serialization Protocol.
- The enterprise bean must not attempt to pass this as an argument or method result. The enterprise bean must pass the result of `SessionContext.getEJBObject`, `SessionContext.getEJBLocalObject`, `EntityContext.getEJBObject`, or `EntityContext.getEJBLocalObject` instead.

These restrictions are enforced by the Java Security Manager when the certified system is run in the security manager enabled mode. The system administrators of the certified system must ensure that they do not provide the user applications security permissions that relax any of the aforementioned restrictions, thereby endangering the security and stability of the certified system.

5.3. DEVELOPER ADVICE FOR USER CREDENTIALS IN REMOTE METHOD INVOCATION

In Remote Method Invocation (RMI), credentials are transmitted from client to server. These credentials populate the security context in the method invocation object. This is implemented using the `setPrincipal` and `setCredential` methods.

Example 5.1. Setting Principal and Credential

```
MethodInvocation mi = new MethodInvocation();
mi.setPrincipal(new SimplePrincipal("myusername"));
mi.setCredential("mypassword");
```

These additional payloads can be retrieved at the server side using similar methods on the invocation object.

Example 5.2. Retrieving Principal and Credential

```
Principal p = mi.getPrincipal();
Object cred = mi.getCredential();
// Now do authentication (and then authorization)
```

CHAPTER 6. OVERVIEW OF THE SECURITY FUNCTIONS

The following sections describe the JBoss security functions included in the product evaluation.

6.1. ACCESS CONTROL

JBoss Enterprise Application Platform has access control mechanisms to restrict access for the following request types:

HTTP

URLs and paths provided with URLs can be protected from access by subjects.

EJB

EJBs and associated method names can be protected from invocation by subjects.

JMS

Message queue destinations and topic destinations can be protected from access by subjects.

Web Services

Plain Old Java Objects (POJOs) deployed as Servlets and Session Beans can be protected from access by subjects.

JMX

The JMX invokers can be protected by validating the role of the authenticated user.

For more information refer to the *Administration and Configuration Guide*

6.2. AUDIT

JBoss Enterprise Application Platform can generate audit records for access control events. Attempts to access web resources, invocation of EJB methods, unauthorized message destinations, and regular Web Service related access control can all be logged. As the administrator you can select the level of events to audit.

The JBoss Application Server (JBoss AS) generates log events at start-up time and when it is shutdown:

Example 6.1. Start up log events

```
15:36:39,026 INFO [ServerImpl] Starting JBoss (Microcontainer)...
15:36:39,027 INFO [ServerImpl] Release ID: JBoss [EAP] 5.1.0 (build:
SVNTag=JBPAPP_5_1_0 date=201009150028)
15:36:39,027 INFO [ServerImpl] Bootstrap URL: null
15:36:39,027 INFO [ServerImpl] Home Dir: /opt/JBoss/EnterprisePlatform-
5.1.0/jboss-eap-5.1/jboss-as
15:36:39,027 INFO [ServerImpl] Home URL:
file:/opt/JBoss/EnterprisePlatform-5.1.0/jboss-eap-5.1/jboss-as/
15:36:39,027 INFO [ServerImpl] Library URL:
file:/opt/JBoss/EnterprisePlatform-5.1.0/jboss-eap-5.1/jboss-as/lib/
15:36:39,028 INFO [ServerImpl] Patch URL: null
15:36:39,028 INFO [ServerImpl] Common Base URL:
```

```
file:/opt/JBoss/EnterprisePlatform-5.1.0/jboss-eap-5.1/jboss-as/common/
15:36:39,028 INFO [ServerImpl] Common Library URL:
file:/opt/JBoss/EnterprisePlatform-5.1.0/
jboss-eap-5.1/jboss-as/common/lib/
15:36:39,028 INFO [ServerImpl] Server Name: production
15:36:39,028 INFO [ServerImpl] Server Base Dir:
/opt/JBoss/EnterprisePlatform-5.1.0/
jboss-eap-5.1/jboss-as/server
15:36:39,028 INFO [ServerImpl] Server Base URL:
file:/opt/JBoss/EnterprisePlatform-5.1.0/jboss-eap-5.1/jboss-as/server/
15:36:39,028 INFO [ServerImpl] Server Config URL:
file:/opt/JBoss/EnterprisePlatform-5.1.0/jboss-eap-5.1/jboss-
as/server/production/conf/
15:36:39,028 INFO [ServerImpl] Server Home Dir:
/opt/JBoss/EnterprisePlatform-5.1.0/jboss-eap-5.1/jboss-
as/server/production
15:36:39,029 INFO [ServerImpl] Server Home URL:
file:/opt/JBoss/EnterprisePlatform-5.1.0/jboss-eap-5.1/jboss-
as/server/production/
15:36:39,029 INFO [ServerImpl] Server Data Dir:
/opt/JBoss/EnterprisePlatform-5.1.0/jboss-eap-5.1/jboss-
as/server/production/data
15:36:39,029 INFO [ServerImpl] Server Library URL:
file:/opt/JBoss/EnterprisePlatform-5.1.0/jboss-eap-5.1/jboss-
as/server/production/lib/
15:36:39,029 INFO [ServerImpl] Server Log Dir:
/opt/JBoss/EnterprisePlatform-5.1.0/jboss-eap-5.1/jboss-
as/server/production/log
15:36:39,029 INFO [ServerImpl] Server Native Dir:
/opt/JBoss/EnterprisePlatform-5.1.0/jboss-eap-5.1/jboss-
as/server/production/tmp/native
15:36:39,029 INFO [ServerImpl] Server Temp Dir:
/opt/JBoss/EnterprisePlatform-5.1.0/jboss-eap-5.1/jboss-
as/server/production/tmp
15:36:39,029 INFO [ServerImpl] Server Temp Deploy Dir:
/opt/JBoss/EnterprisePlatform-5.1.0/jboss-eap-5.1/jboss-
as/server/production/tmp/deploy
15:36:39,587 INFO [ServerImpl] Starting Microcontainer,
bootstrapURL=file:/opt/JBoss/EnterprisePlatform-5.1.0/jboss-eap-
5.1/jboss-as/server/production/conf/bootstrap.xml
15:36:40,024 INFO [VFSCacheFactory] Initializing VFSCache
[org.jboss.virtual.plugins.cache.CombinedVFSCache]
15:36:40,026 INFO [VFSCacheFactory] Using VFSCache
[CombinedVFSCache[real-cache: null]]
15:36:40,259 INFO [CopyMechanism] VFS temp dir:
/opt/JBoss/EnterprisePlatform-5.1.0/jboss-eap-5.1/jboss-
as/server/production/tmp
15:36:40,260 INFO [ZipEntryContext] VFS force nested jars copy-mode is
enabled.
15:36:41,278 INFO [ServerInfo] Java version: 1.6.0_18,Sun Microsystems
Inc.
15:36:41,278 INFO [ServerInfo] Java Runtime: OpenJDK Runtime
Environment (build 1.6.0_18-b18)
15:36:41,278 INFO [ServerInfo] Java VM: OpenJDK Server VM 14.0-b16,Sun
Microsystems Inc.
15:36:41,278 INFO [ServerInfo] OS-System: Linux 2.6.34.7-
```

```

61.fc13.i686.PAE,i386
15:36:41,279 INFO [ServerInfo] VM arguments: -Dprogram.name=run.sh -
Xms1303m -Xmx1303m -XX:MaxPermSize=256m -
Dorg.jboss.resolver.warning=true -Dsun.rmi.dgc.client.gcInterval=36000000
-Dsun.rmi.dgc.server.gcInterval=36000000 -
Dsun.lang.ClassLoader.allowArraySyntax=true -
Djava.net.preferIPv4Stack=true -
Djava.endorsed.dirs=/opt/JBoss/EnterprisePlatform-5.1.0/jboss-eap-
5.1/jboss-as/lib/endorsed
15:36:41,302 INFO [JMXKernel] Legacy JMX core initialized

```

Example 6.2. Shutdown log events

```

2010-11-19 15:59:54,304 INFO
[org.jboss.bootstrap.microcontainer.ServerImpl] (JBoss Shutdown Hook)
Runtime shutdown hook called, forceHalt: true
2010-11-19 15:59:54,305 INFO [org.apache.coyote.http11.Http11Protocol]
(JBoss Shutdown Hook) Pausing Coyote HTTP/1.1 on http-127.0.0.1-8080
2010-11-19 15:59:54,322 INFO [org.apache.coyote.http11.Http11Protocol]
(JBoss Shutdown Hook) Stopping Coyote HTTP/1.1 on http-127.0.0.1-8080
2010-11-19 15:59:54,326 INFO [org.apache.coyote.ajp.AjpProtocol] (JBoss
Shutdown Hook) Pausing Coyote AJP/1.3 on ajp-127.0.0.1-8009
2010-11-19 15:59:54,332 INFO [org.apache.coyote.ajp.AjpProtocol] (JBoss
Shutdown Hook) Stopping Coyote AJP/1.3 on ajp-127.0.0.1-8009
2010-11-19 15:59:54,396 INFO
[org.jboss.web.tomcat.service.deployers.TomcatDeployment] (JBoss
Shutdown Hook) undeploy, ctxPath=/jmx-console
2010-11-19 15:59:54,417 INFO
[org.jboss.web.tomcat.service.deployers.TomcatDeployment] (JBoss
Shutdown Hook) undeploy, ctxPath=/
2010-11-19 15:59:54,424 INFO
[org.jboss.web.tomcat.service.deployers.TomcatDeployment] (JBoss
Shutdown Hook) undeploy, ctxPath=/admin-console
2010-11-19 15:59:54,462 INFO
[org.jboss.resource.connectionmanager.ConnectionFactoryBindingService]
(JBoss Shutdown Hook) Unbound ConnectionManager
'jboss.jca:service=ConnectionFactoryBinding,name=JmsXA' from JNDI name
'java:JmsXA'
2010-11-19 15:59:54,512 INFO
[org.jboss.jms.server.connectionfactory.ConnectionFactory] (JBoss
Shutdown Hook)
org.jboss.jms.server.connectionfactory.ConnectionFactory@8301 undeployed
2010-11-19 15:59:54,513 INFO
[org.jboss.jms.server.connectionfactory.ConnectionFactory] (JBoss
Shutdown Hook)
org.jboss.jms.server.connectionfactory.ConnectionFactory@b24e3f
undeployed
2010-11-19 15:59:54,514 INFO
[org.jboss.jms.server.connectionfactory.ConnectionFactory] (JBoss
Shutdown Hook)
org.jboss.jms.server.connectionfactory.ConnectionFactory@355f75
undeployed
2010-11-19 15:59:54,514 INFO

```

```
[org.jboss.jms.server.destination.QueueService] (JBoss Shutdown Hook)
Queue[/queue/DLQ] stopped
2010-11-19 15:59:54,515 INFO
[org.jboss.jms.server.destination.QueueService] (JBoss Shutdown Hook)
Queue[/queue/ExpiryQueue] stopped

...

2010-11-19 15:59:59,358 INFO
[org.jboss.bootstrap.microcontainer.ServerImpl] (JBoss Shutdown Hook)
Shutdown complete
```

The audit facility is based on the integrated log4j mechanism. log4j has three main components: loggers, appenders, and layouts. These three types of components work together to enable developers to log messages according to message type and level, and to control at run-time how these messages are formatted and where they are reported.

The audit information is recorded in text files which can be reviewed using tools from the underlying operating system, such as pagers or editors.

User information (principal name) appears *only* in the first log that records the authentication request, and also in the ERROR log generated if the authentication is unsuccessful. Subsequent log events do not record explicitly the user executing the methods.

User information can be obtained by using the container and thread ids that are recorded in each audit log and remain during the life of the user session.

In [Example 6.3, “Log output”](#), the first log entry informs that authentication for container 753, thread id 826541 has been requested by principal name “scott”. The second log records the execution of a method, and, although the principal name does not appear, it can be inferred by looking at all logs with the same container and thread id.

Example 6.3. Log output

```
2008-12-12 16:04:33,753 826541 TRACE
[org.jboss.ejb.plugins.SecurityInterceptor]
(WorkerThread#0[127.0.0.1:33182]:) Authenticated principal=scott
2008-12-12 16:04:33,753 826541 TRACE
[org.jboss.ejb.plugins.SecurityInterceptor]
(WorkerThread#0[127.0.0.1:33182]:) method=public abstract
org.jboss.test.jca.securedejb CallerIdentity
org.jboss.test.jca.securedejb CallerIdentityHome.create() throws
javax.ejb.CreateException, java.rmi.RemoteException, interface=HOME,
requiredRoles=[CallerIdentityUser]
```

6.2.1. Enabling Additional Logging

Additional logging for EJB application requests has been configured during the setup process of this guide when audit logging was configured. For more information regarding audit logging configuration refer to [Section 2.5.2, “Setup Configuration”](#)

6.3. CLUSTERING

A cluster is a group of linked systems (nodes) working closely together to increase efficiency. Clustering enables the execution of applications on several parallel servers. In a JBoss Enterprise Application Platform cluster, each node is a JBoss server instance. Several JBoss server instances are grouped together to form a cluster, also known as a "partition".

JBoss Enterprise Application Platform implements two different cluster configurations: a failover cluster and a load-balanced cluster.

In a failover cluster scenario, a single node serves requests from clients. In the event that the node fails, another node in the cluster continues to service requests.

In a load-balanced cluster scenario, multiple nodes serve requests from clients. In this scenario, a single address is serviced with the power of multiple systems.

In both cases, the server state is distributed across different servers. If any of the servers fails, the application is still accessible through other active cluster nodes.

Communication between the different cluster nodes ensures the data consistency of the following information:

- Applications - an application deployed on one node is replicated to the other nodes of the cluster (farming deployment)
- State of HTTP sessions, EJB 3.0 session beans, EJB 3.0 entity beans, as well as Hibernate persistence objects (distributed state replication service using JBoss Cache)
- State of HTTP sessions and EJB 2.x session beans (distributed state replication service using HttpSessionState MBean)
- JMS queues.

6.4. IDENTIFICATION AND AUTHENTICATION

Each user is assigned a unique user identifier. Access control decisions and auditing use this identifier. JBoss Enterprise Application Platform authenticates the user's claimed identity before allowing the user to perform any actions. After successful authentication JBoss Enterprise Application Platform associates the identifier with the thread spawned for the user.

JBoss Enterprise Application Platform provides different identification and authentication mechanisms for various request types.

HTTP and Web Services

HTTP-basic authentication, HTTP-digest authentication, form-based authentication, client certificate based authentication.

EJB

Username and password-based authentication, and client certificate based authentication.

JMS

Username and password-based authentication.

JNDI

Username and password-based authentication.

JMX Invokers

Username and password-based authentication.

JBoss Enterprise Application Platform uses JBoss SX framework to implement identification and authentication. The JBossSX framework utilizes the Java Authentication and Authorization Service (JAAS) provided by the Java Virtual Machine. The authentication capabilities of JAAS are used to implement the declarative role-based J2EE security model.

The following authentication back-ends are configurable with the JAAS modules.

- File-based storage
- BaseCertLoginModule
- LDAP
- Databases accessible through JDBC

Password quality can be enforced with configuration options for the JAAS modules provided by JBoss Enterprise Application Platform.

For information on how to configure the JAAS modules, refer to the "Using JBoss Login Modules" section of the *JBoss Security Guide*

6.5. TRANSACTION ROLLBACK

JBoss Enterprise Application Platform supports the aggregation of operations into transactions, which can be applied and rolled back consistently.

A transaction is a unit of work containing one or more operations involving one or more shared resources having atomicity, consistency, isolation and durability (ACID) properties - the four important properties of transactions.

Atomicity

A transaction must be atomic. This means that either all the work done in the transaction must be performed, or none of it must be performed. Doing only part of a transaction is not allowed.

Consistency

When a transaction is completed, the system must be in a stable and consistent condition.

Isolation

Different transactions must be isolated from each other. This means that the partial work done in one transaction is not visible to other transactions until the transaction is committed, and that each process in a multi-user system can be programmed as if it was the only process accessing the system.

Durability

The changes made during a transaction are made persistent when it is committed. When a transaction is committed, its changes will not be lost, even if the server crashes afterward.

The default transaction manager for JBoss Enterprise Application Platform is JBoss Transactions, a fast in-VM transaction manager implementation.

Traditionally, ACID transaction systems have shared the following characteristics:

- transactions are short lived
- resources (such as databases) are locked for the duration of the transaction
- participants have a high degree of trust with each other.

The advent of the Internet and Web services has given rise to distributed transactions between participants unknown to each other. JBoss Transactions adds native support for Web services transactions by providing the components necessary to build interoperable, reliable, multi-party, Web services-based applications with minimum effort.

The programming interfaces are based on the Java API for XML Transactions (JAXTX) and include protocol support for the WS-AtomicTransaction and WS-BusinessActivity specifications. JBoss is designed to support multiple coordination protocols.

JBoss Enterprise Application Platform supports both local and distributed transactions. A transaction is considered to be distributed if it spans multiple process instances, i.e. virtual machines (VMs). Typically a distributed transaction will contain participants that are located within multiple VMs but the transaction is coordinated in a separate VM (or co-located with one of the participants). If the deployment requires distributed transactions then the Web Services transactions component can be utilized, which uses SOAP/HTTP.

APPENDIX A. PORT CONFIGURATION IN JBOSS ENTERPRISE APPLICATION PLATFORM

Certain configuration files require ports to be set so components that reference them can access the files through the correct port. The following tables describe the configuration files with specific port requirements.

A.1. TCP SETTINGS

Table A.1, “[bindings-jboss-beans.xml Port Configuration](#)” describes TCP port configuration for `bindings-jboss-beans.xml`. The file is located in `JBOSS_HOME/server/production/conf/bindingservice.beans/META-INF/bindings-jboss-beans.xml`

Table A.1. `bindings-jboss-beans.xml` Port Configuration

Port	Status	Purpose
1098	Enabled	RMI naming service
3528	Disabled	IIOp port assigned by IANA
4444	Enabled	RMI JRMP invoker
4445	Enabled	RMI pooled invoker
4446	Enabled	Remoting server connector
4447	Enabled	Remoting server connector
4457	Enabled	Remoting server connector
4712	Enabled	JBossTS recovery manager
4713	Enabled	JBossTS transaction status manager
4714	Enabled	Process ID for JBossTS
8080	Enabled	HTTP Connector
8083	Enabled	RMI Classloading mini web server
8443	Enabled	JBossWS HTTPS connector socket

Table A.2, “[cluster-service.xml Port Configuration](#)” describes TCP port configuration for `cluster-service.xml`. The file is located in `JBOSS_HOME/server/production/deploy/cluster-service.xml`

Table A.2. `cluster-service.xml` Port Configuration

Port	Status	Purpose
1100	Disabled	Clustering
1101	Disabled	Clustering
4448	Disabled	PooledInvokerHA

Table A.3, “[clustered-hsqldb-persistence-service.xml Port Configuration](#)” describes TCP port configuration for `clustered-hsql-persistence-service.xml`. The file is located in `JBOSS_HOME/server/production/deploy/messaging/clustered-hsql-persistence-service.xml`

Table A.3. clustered-hsqldb-persistence-service.xml Port Configuration

Port	Status	Purpose
7900	Disabled	

Table A.4, “[ejb3-connectors-jboss-beans.xml Port Configuration](#)” describes TCP port configuration for `ejb3-connectors-jboss-beans.xml`. The file is located in `JBOSS_HOME/server/production/deploy/ejb3-connectors-jboss-beans.xml`

Table A.4. ejb3-connectors-jboss-beans.xml Port Configuration

Port	Status	Purpose
3873	Enabled	EJB3 Remoting Connector

Table A.5, “[jboss-service.xml Port Configuration](#)” describes TCP port configuration for `jboss-service.xml`. The file is located in `JBOSS_HOME/server/production/conf/jboss-service.xml`

Table A.5. jboss-service.xml Port Configuration

Port	Status	Purpose
1099	Enabled	RMI bootstrap naming service

Table A.6, “[jgroups-channelfactory-stacks.xml Port Configuration](#)” describes TCP port configuration for `jgroups-channelfactory-stacks.xml`. The file is located in `JBOSS_HOME/server/production/deploy/cluster/jgroups-channelfactory.sar/META-INF/jgroups-channelfactory-stacks.xml`

Table A.6. jgroups-channelfactory-stacks.xml Port Configuration

Port	Status	Purpose
7500	Enabled	UDP and TCP stack diagnostics port
7600	Enabled	TCP stack <i>with</i> flow control/message bundling
7650	Enabled	TCP stack <i>without</i> flow control/message bundling
7900	Enabled	
45700	Enabled	TCP stack diagnostics port
45701	Enabled	Multicast port for multicast-based automatic discovery
45710	Enabled	Multicast port for TCP based stack optimized for the JBoss Messaging Data Channel

Table A.6, “[jgroups-channelfactory-stacks.xml Port Configuration](#)” describes TCP port configuration for `server.xml`. The file is located in `JBOSS_HOME/server/production/deploy/jbossweb.sar/server.xml`

Table A.7. `server.xml` Port Configuration

Port	Status	Purpose
8009	Disabled	AJP Connector

A.2. UDP SETTINGS

Table A.8, “[bindings-jboss-beans.xml Port Configuration](#)” describes UDP port configuration for `bindings-jboss-beans.xml`. The file is located in `JBOSS_HOME/server/production/conf/bindingservice.beans/META-INF/bindings-jboss-beans.xml`

Table A.8. `bindings-jboss-beans.xml` Port Configuration

Port	Status	Purpose
1102	Disabled	
1161	Disabled	snmp
1162	Disabled	snmp
53200	Enabled	Multicast port for IP multicast based stack, with flow control, message bundling enabled. For JBoss Messaging Control Channel clustered services.

Port	Status	Purpose
54200	Enabled	JBoss Messaging port stack with flow control, message bundling disabled, and stack optimized
54225	Enabled	Multicast port for IP multicast based stack, with flow control, message bundling enabled. For services that make high-volume asynchronous RPCs, such as JBoss Cache.
54250	Enabled	Multicast port for IP multicast based stack, with flow control and message bundling disabled. Used for synchronous calls, and low message volume rates and sizes. Not suitable for high volume; you may run out of memory.

Table A.6, “[jgroups-channelFactory-stacks.xml Port Configuration](#)” describes UDP port configuration for `jgroups-channelFactory-stacks.xml`. The file is located in `JBOSS_HOME/server/production/deploy/cluster/jgroups-channelFactory.sar/META-INF/jgroups-channelFactory-stacks.xml`

Table A.9. `jgroups-channelFactory-stacks.xml` Port Configuration

Port	Status	Purpose
7500	Enabled	Diagnostics port for UDP and TCP stacks
45688	Enabled	Multicast port for IP multicast based stack, with flow control, message bundling disabled, and stack optimized
45689	Enabled	Multicast port for IP multicast based stack, with flow control, message bundling disabled, and stack optimized
45699	Enabled	Multicast port for IP multicast based stack, with flow control, message bundling disabled, and stack optimized

APPENDIX B. REVISION HISTORY

Revision 5.1.1-121.400 Rebuild with publican 4.0.0	2013-10-31	Rüdiger Landmann
Revision 5.1.1-121 Rebuild for Publican 3.0	2012-07-18	Anthony Towns
Revision 5.1.0-120 Finalized document for JBoss Enterprise Application Platform 5 Common Criteria Certification General Availability (GA).	Tue Feb 14 2012	Jared Morgan