JBoss Enterprise Application Platform Common Criteria Certification 6.2.2

Common Criteria Configuration Guide

for use with JBoss Enterprise Application Platform 6 Common Criteria Certification

Last Updated: 2017-11-17

JBoss Enterprise Application Platform Common Criteria Certification 6.2.2 Common Criteria Configuration Guide

for use with JBoss Enterprise Application Platform 6 Common Criteria Certification

Nidhi Chaudhary

Lucas Costi

Russell Dickenson

Sande Gilda

Vikram Goyal

Eamon Logue

Darrin Mison

Scott Mumford

David Ryan

Nidhi Srinivas

Misty Stanley-Jones

Keerat Verma

Tom Wells

Legal Notice

Abstract

Describes configuring and securing JBoss Enterprise Application Platform $PROD_VER; to meet Common Criteria EAL4 certification.

Table of Contents

CHAPTER 1. INTRODUCTION

Report a bug

1.1. PURPOSE OF THIS DOCUMENT

This document provides guidance to administrators and application developers who wish to use JBoss EAP 6.2.2 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 EAP 6.2.2 is the subject of this document as the Target of Evaluation (TOE) for Common Criteria certification. JBoss EAP 6.2.2 has been evaluated under Common Criteria version 3.1 at level of assurance EAL4. This provides assurance that this product has been structurally tested.

This chapter contains a brief introduction to the CC certification and 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 EAP 6.2.2.

Chapter 4, *Start and Stop JBoss EAP 6* provides instructions on how to start and stop the server, and the different modes of operation.

Chapter 6, *Development Guide for the Common Criteria Certified System* contains guidelines for developers creating applications for JBoss EAP 6.2.2.

Chapter 7, *Overview of the Security Functions* contains the details of the security implementation and usage limitations of JBoss EAP 6.2.2.

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 EAP 6.2.2.

Report a bug

1.2. JBOSS EAP-SPECIFIC CONVENTIONS

All instances of *EAP_HOME* in this guide refer to the JBoss EAP root installation directory. For example, if you used the ZIP installation package and extracted the JBoss EAP binary to your Linux **/home** directory, *EAP_HOME* refers to the **home/*USER*/jboss-eap-*[version]*/** directory.

Report a bug

1.3. 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 more than 20 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, the United Kingdom, and the United States amongst others. 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.

Report a bug

1.4. CERTIFIED DOCUMENTATION

When installing, configuring, and operating JBoss EAP 6.2.2 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 from the Red Hat Customer Portal:

- PDF documentation bundle
- online

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

WARNING

When operating an evaluated configuration, you must refer *only* to the Common Criteria version of the documentation for JBoss EAP 6.2.2. The standard product documentation version may contain information that could result in an evaluated configuration certification breach.

Report a bug

CHAPTER 2. REQUIREMENTS FOR THE EVALUATED CONFIGURATION

Report a bug

2.1. SOFTWARE REQUIREMENTS

Report a bug

2.1.1. Java Virtual Machine

JBoss EAP 6.2.2 is evaluated on the following Java Virtual Machines (JVMs). Only these JVMs are acceptable for the deployment of JBoss EAP 6.2.2.

Table 2.1. Java Virtual Machine Version

JVM	Version(s)
OpenJDK	1.6.x
OpenJDK	1.7.x
Oracle JDK	1.6.x
Oracle JDK	1.7.x
IBM JDK	1.6.x
IBM JDK	1.7.x

Report a bug

2.1.2. Operating System

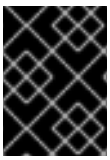
The supported operating systems for this evaluation are limited to the following products:

- Red Hat Enterprise Linux 5
- Red Hat Enterprise Linux 6
- Solaris 10
- Solaris 11
- Microsoft Windows Server 2008
- Microsoft Windows Server 2008 R2
- Microsoft Windows Server 2012

Report a bug

2.1.3. Database JDBC Drivers

JBoss EAP 6.2.2 is evaluated with the following relational database systems. Only these database systems with the specific driver versions are acceptable for use with JBoss EAP 6.2.2.



IMPORTANT

Not every JDK/OS combination is supported. For a list of supported configurations, refer here: https://access.redhat.com/site/articles/111663

Table 2.2. Allowed Database and JDBC Driver Versions

Database	JDBC Driver
IBM DB2 v9.7	IBM DB2 JDBC Universal Driver Architecture version 4.14.122 Driver download: http://www-947.ibm.com/support/entry/portal/Overview/Software/Information_Management/IBM_Data_Server_Client_Packages File name: **`db2jcc4.jar`** SHA-256: d435d21446382019f30afb90c474d1de4152814cf1995606b52a7a5fbf4af07e
IBM DB2 v10.1	IBM DB2 JDBC Universal Driver Architecture version 4.13.127 Driver download: http://www-947.ibm.com/support/entry/portal/Overview/Software/Information_Management/IBM_Data_Server_Client_Packages File name: **`db2jcc4.jar`** SHA-256: 16858ed1b11c30a9cf14424bad173bad0b16bf61ca89614ac86f9c3d4407450e
Oracle 11g R1 (v11.1.0.7.0)	Oracle 11g R1 version 11.1.0.7 Driver download: http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html. File name: **`ojdbc6.jar`** SHA-256: 2468c758f9b1af81201bb5b9665a6216cdbb7fb95e63b9015495d9d0edb47add
Oracle 11g R1 RAC v(11.1.0.7.0)	Oracle 11g RAC version 11.1.0.7 Driver download: http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html. File name: **`ojdbc6.jar`** SHA-256: 2468c758f9b1af81201bb5b9665a6216cdbb7fb95e63b9015495d9d0edb47add

Database	JDBC Driver
Oracle 11g R2	Oracle JDBC Driver version 11.2.0.3.0 Driver download: http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html. File name: **`ojdbc6.jar`** SHA-256: b7a8656754c891f2d9605afc6d2f4d98a5a8d6fbafe0065b24590061794b1460
Oracle 11g R2 RAC	Oracle JDBC Driver version 11.2.0.3.0 Driver download: http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html. File name: **`ojdbc6.jar`** SHA-256: b7a8656754c891f2d9605afc6d2f4d98a5a8d6fbafe0065b24590061794b1460
Oracle 12c	• JDK6: Oracle JDBC Driver version 12.1.0.1.0 JDBC 4.0 compiled with JDK6 Driver download: http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html. File name: **`ojdbc6.jar`** SHA-256: 36faf20b866f830d0cdaad7bc3fdc9366390ec61a67e9064e27e51102e549623 • JDK7: Oracle JDBC Driver version 12.1.0.1.0 JDBC 4.1 compiled with JDK7 Driver download: http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html. File name: **`ojdbc7.jar`** SHA-256: 72449d61e4d2685db8337e54dca2c33876c320aa7940054052da9244a3e934d7
MySQL v5.5	MySQL Connector/J 5.1.23 Driver download: http://downloads.mysql.com/archives/c-j/. File name: **`mysql-connector-java-5.1.23-bin.jar`** SHA-256: 9302461695b75d0d99064ab1fa237b89532deb5e39e10b0a6dd1e70b0c33bb4b
Microsoft SQL Server 2008 R2 SP2	Microsoft SQL Server JDBC Driver 4.0.2206.100 Driver download: http://www.microsoft.com/en-au/download/details.aspx?id=11774. File name: **`sqljdbc4.jar`** SHA-256: 6b423a4e8f11dc5357d1cfa38ee2a42bae87a83126fe3363b4464e5120d33f8c

Database	JDBC Driver
Microsoft SQL Server 2012	Microsoft SQL Server JDBC Driver 4.0.2206.100 Driver download: http://www.microsoft.com/en-au/download/details.aspx?id=11774. File name: **`sqljdbc4.jar`** SHA-256: 6b423a4e8f11dc5357d1cfa38ee2a42bae87a83126fe3363b4464e5120d33f8c
PostgreSQL v9.2	JDBC4 Postgresql Driver, version 9.2-1002 Driver download: http://jdbc.postgresql.org/download/postgresql-9.2-1002.jdbc4.jar. File name: **`postgresql-9.2-1002.jdbc4.jar`** SHA-256: cd1824fa8c059e6376c92020f1e7fe6c6f34772e9f91711327e414f1b979fbe1
Enterprise DB Postgres Plus Advanced Server 9.2	Postgres Plus Advanced Server Driver 9.2 (9.2.1.3) Driver download: http://www.enterprisedb.com/downloads/postgres-postgresql-downloads. File name: **`edb-jdbc14.jar`** SHA-256: a9c7b4519e13524c930172e8a1da4d70f585e8a96275fe002d97b38aa3619649
Sybase ASE 15.7	Sybase jConnect JDBC driver v7.07 (Build 26792/EBF20686) Driver download: http://www.sybase.com/products/allproductsa-z/softwaredeveloperkit/jconnect File name: **`jconn4.jar`** SHA-256: 1349ded4fd4a7ba6fc39b2aa353924c82d4b6812587737350af44ed949d97aee

NOTE

To generate a checksum for a downloaded file, the **`sha256sum`** command can be used on 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 checksum tool.

For information on how to configure each database with JBoss EAP 6.2.2, refer to Section 5.5, “Database Configuration”.

Report a bug

2.2. PHYSICAL REQUIREMENTS

The hardware and software executing JBoss EAP 6.2.2, as well as the software critical to security policy enforcement must be protected from modification by unauthorized parties, whether internal or external. Reasonable physical security measures to ensure that unauthorized persons do not have physical access to the hardware running the JBoss EAP 6.2.2 software must be implemented.

Report a bug

2.3. PERSONNEL REQUIREMENTS

There must be one or more competent individuals who are assigned to manage JBoss EAP 6.2.2, its environment and the security of the information it contains. The system administrative personnel must not be carelessly or willfully negligent, or hostile, and follow and abide by the instructions provided by the administrator documentation.

The developer of user applications executed by JBoss EAP 6.2.2, 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 JBoss EAP 6.2.2.

Report a bug

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 EAP 6.2.2 communicates is assumed to be under the same management control and operate under the same security policy constraints as JBoss EAP 6.2.2.

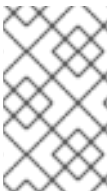
Report a bug

2.4.1. Cluster Connectivity Requirements

JBoss EAP 6.2.2 instances must operate in a network segment that is logically separated from any other network segment by use of a packet filtering mechanism. This packet filter must only allow incoming communication that meets both the following criteria:

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

All outgoing communication from one of the JBoss EAP 6.2.2 instances must be allowed.



NOTE

There are three defined interfaces to separate trusted and untrusted network traffic: public, cluster, and internal. Refer to Section 5.3.1, “Network Interfaces” for more information.

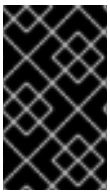
Each cluster node communicates with the other nodes by means of standard network sockets. Whenever this occurs the client side of each connection has a port number assigned to it by the host operating system from a range of ports that are reserved for client sockets. These ports are referred to as *dynamic* 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.

Report a bug

CHAPTER 3. DOWNLOADING AND VERIFYING THE PACKAGES

JBoss EAP 6.2.2 is available in Zip and RPM formats. Zip files are available from the Red Hat Customer Portal (RHCP) at https://access.redhat.com. RPM installation files 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.

Report a bug

3.1. ABOUT THE RED HAT CUSTOMER PORTAL

The *Red Hat Customer Portal* is the centralized platform for Red Hat knowledge and subscription resources. Use the *Red Hat Customer Portal* to:

- Manage and maintain Red Hat entitlements and support contracts;
- Download officially-supported software;
- Access product documentation and the Red Hat Knowledgebase;
- Contact Global Support Services; and
- File bugs against Red Hat products.

The Customer Portal is available here: https://access.redhat.com.

Report a bug

3.2. 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. Some web browsers also display the padlock icon in the address bar.

If the 'security padlock' is 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 padlock icon.
2. From the pop-up box, click **`More Information`**.
3. From the Page Info window, click **`Security`**.
4. The certificate displays details such as who owns the site, who issued the certificate, when it was issued, when it expires and fingerprint verification strings.

Figure 3.1. Example of the Red Hat Network 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 site, contact Red Hat Support and report this problem.

Report a bug

3.3. VERIFY THE DOWNLOADED FILES

Each downloaded file needs to be checked to verify that it is for the certified version of JBoss EAP 6.2.2. The Red Hat Customer Portal lists the SHA-256 hash sum for each file. If the SHA-256 hash sum of a downloaded file matches that quoted on the Red Hat Customer Portal, you can assume it is verified.

For JBoss downloads, you can view the SHA-256 hash sum on the **`Software Details`** page for each file on the by clicking on the file name in the **`Download File`** list. For Red Hat Enterprise Linux downloads, the SHA-256 has sum is listed next to the ISO download link.

On Apple OSX, the **sha256** command must be replaced with **shasum -a 256**. On Microsoft Windows, a third-party SHA256 hash sum utility is required as there is no native utility.

Procedure 3.2. Using the sha256sum tool on Linux or Unix

1. Open a terminal, and navigate to the directory where the file was downloaded.
2. Execute the **sha256sum** command (or equivalent) on the file. For example:

```
$ sha256sum jboss-eap-6.2.2.zip
9414619e186708b34381ddbef1901a51335b5b45464838effb153b610b27918f
jboss-eap-6.2.2.zip
```

Result

The value generated by the **sha256sum** utility must match the value displayed on the Red Hat Customer Portal for the file. If they are not the same, your download is either incomplete or corrupt, and you will need to download the file again. If the checksum will still not successfully validate after several attempted downloads, contact Red Hat Support for assistance.

Report a bug

3.4. ZIP INSTALLATION

3.4.1. Download JBoss EAP 6 (Zip Installation)

Summary

The JBoss EAP 6 Zip file is available from https://access.redhat.com. The Zip file installation is platform-independent and is the preferred way to install JBoss EAP 6 on all supported platforms. This topic covers the steps to download the required archive.

Procedure 3.3. Download the Zip file

1. Log into the Red Hat Customer Portal at https://access.redhat.com.
2. Click **Downloads**.
3. Click **Red Hat JBoss Enterprise Application Platform** in the **Product Downloads** list.
4. Select **6.2 CC** from the **Version** drop-down menu.
5. Select **Red Hat JBoss Enterprise Application Platform 6.2.2 Common Criteria Zip** from the list of **Software Downloads**.
6. When the download has finished, verify that the checksum of the downloaded file matches the checksum listed on the Customer Portal. See Section 3.3, “Verify the Downloaded Files”.

Result

JBoss EAP 6 has been downloaded successfully to the target machine, and is ready for installation.

Report a bug

3.4.2. Install JBoss EAP 6 (Zip Installation)

Summary

This topic covers the steps to install JBoss EAP 6.2.2 using the downloaded Zip file.

Procedure 3.4. Zip File Installation

1. **Use an appropriate application to extract the `jboss-eap-6.2.2-cc.zip` bundle.**
   - In Red Hat Enterprise Linux and Solaris environments, use the **`unzip`** utility to extract the contents of the ZIP archive.
   - In a Microsoft Windows environment, right-click the file and select **`Extract All`**.

   After extracting the bundle, you will have the **`jboss-eap-6.2.2.zip`** archive and several sub-directories.
2. **Move `jboss-eap-6.2.2.zip` to the desired location.**
   Move **`jboss-eap-6.2.2.zip`** to the directory where you plan to install JBoss EAP 6. The user who will start and stop the server must have read and write access to this directory.
3. **Extract `jboss-eap-6.2.2.zip` to the desired location.**
   The directory created by extracting the Zip archive is the top-level directory for the server. This is referred to as ***`EAP_HOME`***.

Result

JBoss EAP 6.2.2 has been installed successfully.

Report a bug

3.5. RPM INSTALLATION FROM ISO

The ISO installation file for JBoss EAP 6.2.2 is to be downloaded from the Red Hat Customer Portal and contains all security and patch errata.

IMPORTANT

Installing JBoss EAP 6.2.2 using the RPM method *must* be done from the ISO file for a certified configuration. Installing JBoss EAP 6.2.2 using RPMs directly from the Red Hat Network is not valid for a certified configuration.

Prerequisites

- Register the server on the Red Hat Network.
- Subscribe to the **`Red Hat Enterprise Linux Server`** base software channel appropriate to your Red Hat Enterprise Linux version.
- Do not subscribe to the **`JBoss Application Platform for Server`** sub-channel, in the **`JBoss Enterprise Platform`** group.

Report a bug

3.5.1. Download JBoss EAP 6 ISO (RPM Installation)

Procedure 3.5. Download the JBoss EAP 6.2.2 ISO

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

1. Log into the Red Hat Customer Portal at https://access.redhat.com.
2. Click **Downloads**.
3. Click **`Red Hat JBoss Enterprise Application Platform`** in the **`Product Downloads`** list.
4. Select **`6.2 CC`** from the **`Version`** drop-down menu.
5. Click the link to the ISO to begin the download.

   Ensure you download the correct ISO for your version of Red Hat Enterprise Linux. For Red Hat Enterprise Linux 5, the file name is **`jboss-eap-6.2.2-cc-rhel-5-noarch.iso`** and for Red Hat Enterprise Linux 6, the file name is **`jboss-eap-6.2.2-cc-rhel-6-noarch.iso`**.
6. When the download has finished, verify that the checksum of the downloaded ISO matches the checksum listed on the Customer Portal. See Section 3.3, “Verify the Downloaded Files”.

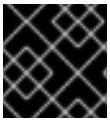
Report a bug

3.5.2. Install JBoss EAP 6 from ISO (RPM Installation)

This procedure is applicable only to Red Hat Enterprise Linux.

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 JBoss EAP 6 from ISO on Red Hat Enterprise Linux



IMPORTANT

You must activate superuser privileges to install the ISO image.

1. **Mount ISO Image**
   Mount the ISO image downloaded in Section 3.5.1, “Download JBoss EAP 6 ISO (RPM Installation)” to **`/mnt/jboss`**.

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

2. **Create Repository**
   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 EAP 6.2.2**
   Run the following command:

```
[root ~]# yum groupinstall jboss-eap6
```

Result

The installation is complete. The default ***EAP_HOME*** path for the RPM installation is **/usr/share/jbossas**.

Report a bug

3.6. CONFIRMING THE VERSION OF YOUR JBOSS EAP 6 INSTALLATION

There are three ways to verify the version number of your JBoss EAP 6 installation.

Using the -V with the startup script

Retrieve information about the version of your JBoss EAP 6 installation by running the same script used to start the server with only the **-V** switch. If your installation is a standalone or managed domain, for Red Hat Enterprise Linux, HP-UX and Solaris, this script is either **standalone.sh** or **domain.sh**, and on Microsoft Windows Server it is the equivalent **.bat** scripts. The startup scripts are located in ***EAP_HOME*/bin**.

Running the startup script with only the **-V** switch will not start the server, and does not require the server to be running. It displays information about the JBoss EAP version and its configured Java environment. Below is an example of using it on an installation of JBoss EAP 6 on Red Hat Enterprise Linux. Note the version number (**JBoss EAP 6.2.2.GA**) displayed as the last line of the output.

```
$ ./standalone.sh -V
=========================================================================

  JBoss Bootstrap Environment

  JBOSS_HOME: /home/user/jboss-eap-6.2

  JAVA: /usr/lib/jvm/java-1.7.0-openjdk.x86_64/bin/java

  JAVA_OPTS:  -server -XX:+UseCompressedOops -Xms1303m -Xmx1303m -
XX:MaxPermSize=256m -Djava.net.preferIPv4Stack=true -
Djboss.modules.system.pkgs=org.jboss.byteman -Djava.awt.headless=true

=========================================================================

14:38:11,609 INFO  [org.jboss.modules] (main) JBoss Modules version
1.3.0.Final-redhat-2
14:38:11,780 INFO  [stdout] (main) JBoss EAP 6.2.2.GA (AS 7.3.0.Final-
redhat-14)
```

Using the Web console

When the JBoss EAP 6 server is running, the version information is displayed at top of the home page of the Web Console, located at http://localhost:9990/console/.

View the console output, or server.log file

When a server is started, the version is echoed to the console, and written to the server log. For standalone configurations the server log is located at ***EAP_HOME*/standalone/log/server.log**, and for managed domain servers it is ***EAP_HOME*/domain/servers/*SERVER_NAME*/log/server.log**:

```
14:56:16,256 INFO  [org.jboss.as] (Controller Boot Thread) JBAS015874:
JBoss EAP 6.2.2.GA (AS 7.3.0.Final-redhat-14) started in 4872ms - Started
147 of 209 services (61 services are passive or on-demand)
```

Report a bug

3.7. UPDATING A COMMON CRITERIA COMPLIANT JBOSS EAP 6.2.2 INSTALLATION

Updates are regularly released for JBoss EAP, and these updates may include fixes for important security issues. However, applying any updates to a Common Criteria compliant JBoss EAP installation will invalidate that installation's Common Criteria certification.

WARNING

Not applying security updates is potentially a serious risk. If security updates are released, it is at your discretion whether to apply them to a Common Criteria Compliant JBoss EAP installation. You may wish to forgo Common Criteria certification in favor of applying updates that resolve security issues.

Report a bug

3.7.1. Applying Patches to a Zip Installation

IMPORTANT

Applying patches to a Common Criteria compliant installation will invalidate that installation's Common Criteria certification.

- Use the **patch** command to apply patches to your zip installation. See the *Installation Guide* for instructions.

Report a bug

3.7.2. Applying Patches to a RPM Installation from ISO

IMPORTANT

Applying patches to a Common Criteria compliant installation will invalidate that installation's Common Criteria certification.

1. **Remove the local repository**
   Remove the file named **`jbosslocal.repo`** from **`/etc/yum.repos.d/`** :

   ```
   [root ~]# rm /etc/yum.repos.d/jbosslocal.repo
   ```

2. Update the installation as a normal RPM installation. See the *Installation Guide*.

Report a bug

CHAPTER 4. START AND STOP JBOSS EAP 6

Report a bug

4.1. START JBOSS EAP 6

Start JBoss EAP 6 in one of the following ways:

- Section 4.2, “Start JBoss EAP 6 as a Standalone Server”
- Section 4.3, “Start JBoss EAP 6 as a Managed Domain”

Report a bug

4.2. START JBOSS EAP 6 AS A STANDALONE SERVER

Summary

This topic covers the steps to start JBoss EAP 6 as a Standalone Server.

Procedure 4.1. Start the Platform Service as a Standalone Server

1. **For Red Hat Enterprise Linux.**
   Run the command: ***EAP_HOME*`/bin/standalone.sh`**
2. **For Microsoft Windows Server.**
   Run the command: ***EAP_HOME*`\bin\standalone.bat`**
3. **Optional: Specify additional parameters.**
   To print a list of additional parameters to pass to the start-up scripts, use the ***-h*** parameter.

Result

The JBoss EAP 6 Standalone Server instance starts.

Report a bug

4.3. START JBOSS EAP 6 AS A MANAGED DOMAIN

Order of Operations

The domain controller must be started before any slave servers in any server groups in the domain. Use this procedure first on the domain controller, and then on each associated host controller and each other host associated with the domain.

Procedure 4.2. Start the Platform Service as a Managed Domain

1. **For Red Hat Enterprise Linux.**
   Run the command: ***EAP_HOME*`/bin/domain.sh`**
2. **For Microsoft Windows Server.**
   Run the command: ***EAP_HOME*`\bin\domain.bat`**
3. **Optional: Pass additional parameters to the start-up script.**
   For a list of parameters you can pass to the start-up script, use the ***-h*** parameter.

Result

The JBoss EAP 6 Managed Domain instance starts.

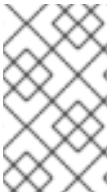
Report a bug

4.4. START JBOSS EAP 6 WITH AN ALTERNATIVE CONFIGURATION

If you do not specify a configuration file, the server starts with the default file. However, when you start the server, you can specify a configuration manually. The process varies slightly, depending on whether you are using a Managed Domain or Standalone Server, and depending on which operating system you are using.

Prerequisites

- Before using an alternate configuration file, prepare it using the default configuration as a template. For a Managed Domain, the configuration file needs to be placed in the ***EAP_HOME*/domain/configuration/** directory. For a Standalone Server, the configuration file should be placed in the ***EAP_HOME*/standalone/configuration/** directory.



NOTE

Several example configurations are included in the ***EAP_HOME*/docs/examples/configs/** directory. Use these examples to enable extra features such as clustering or the Transactions XTS API.

Procedure 4.3. Start the Instance with an Alternative Configuration

1. **Standalone server**
   For a Standalone Server, provide the filename of the configuration file as an option to the ***--server-config*** parameter. The configuration file must be located in the ***EAP_HOME*/standalone/configuration/** directory, and you need to specify the file path relative to that directory.

   Example 4.1. Using an alternate configuration file for a Standalone Server in Red Hat Enterprise Linux

   ```
   [user@host bin]$ ./standalone.sh --server-config=standalone-alternate.xml
   ```

   This example uses the ***EAP_HOME*/standalone/configuration/standalone-alternate.xml** configuration file.

   Example 4.2. Using an alternate configuration file for a Standalone Server in Microsoft Windows Server

   ```
   C:\EAP_HOME\bin> standalone.bat --server-config=standalone-alternate.xml
   ```

   This example uses the ***EAP_HOME*\standalone\configuration\standalone-alternative.xml** configuration file.

2. **Managed Domain**
   For a Managed Domain, provide the file name of the configuration file as an option to the ***--domain-config*** parameter. The file must be present in the ***EAP_HOME*/domain/configuration/** directory, and you need to specify the path relative to that directory.

   Example 4.3. Using an alternate configuration file for a Managed Domain in Red Hat Enterprise Linux

   ```
   [user@host bin]$ ./domain.sh --domain-config=domain-alternate.xml
   ```

   This example uses the ***EAP_HOME*/domain/configuration/domain-alternate.xml** configuration file.

   Example 4.4. Using an alternate configuration file for a Managed Domain in Microsoft Windows Server

   ```
   C:\EAP_HOME\bin> domain.bat --domain-config=domain-alternate.xml
   ```

   This example uses the ***EAP_HOME*\domain\configuration\domain-alternate.xml** configuration file.

Result

JBoss Enterprise Application Platform is now running, using your alternate configuration file.

Report a bug

4.5. REFERENCE OF SWITCHES AND ARGUMENTS TO PASS AT SERVER RUNTIME

The application server startup script accepts the addition of arguments and switches at runtime. The use of these parameters allows for the server to be started under alternative configurations to those defined in the **standalone.xml**, **domain.xml** and **host.xml** configuration files. This might include starting the server with an alternative set of socket bindings or a secondary configuration. A list of these available parameters can be accessed by passing the help switch at startup.

Example 4.5.

The following example is similar to the server startup explained in Section 4.2, “Start JBoss EAP 6 as a Standalone Server” and Section 4.3, “Start JBoss EAP 6 as a Managed Domain”, with the addition of the ***-h*** or ***--help*** switches. The results of the help switch are explained in the table below.

Standalone mode:

```
[localhost bin]$ standalone.sh -h
```

Domain mode:

```
[localhost bin]$ domain.sh -h
```

Table 4.1. Table of runtime switches and arguments

Argument or Switch	Mode	Description
`--admin-only`	Standalone	Set the server's running type to **`ADMIN_ONLY`**. This will cause it to open administrative interfaces and accept management requests, but not start other runtime services or accept end user requests.
`--admin-only`	Domain	Set the host controller's running type to **`ADMIN_ONLY`** causing it to open administrative interfaces and accept management requests but not start servers or, if this host controller is the master for the domain, accept incoming connections from slave host controllers.
`-b <value>`, **`-b=<value>`**	Standalone, Domain	Set system property **`jboss.bind.address`** to the given value.
`-b<interface>=<value>`	Standalone, Domain	Set system property **`jboss.bind.address.<interface>`** to the given value.
`--backup`	Domain	Keep a copy of the persistent domain configuration even if this host is not the Domain Controller.
`-c <config>`, **`-c=<config>`**	Standalone	Name of the server configuration file to use. The default is **`standalone.xml`**.
`-c <config>`, **`-c=<config>`**	Domain	Name of the server configuration file to use. The default is **`domain.xml`**.
`--cached-dc`	Domain	If the host is not the Domain Controller and cannot contact the Domain Controller at boot, boot using a locally cached copy of the domain configuration.
`--debug [<port>]`	Standalone	Activate debug mode with an optional argument to specify the port. Only works if the launch script supports it.
`-D<name>[=<value>]`	Standalone, Domain	Set a system property.
`--domain-config=<config>`	Domain	Name of the server configuration file to use. The default is **`domain.xml`**.
`-h`, **`--help`**	Standalone, Domain	Display the help message and exit.

Argument or Switch	Mode	Description
--host-config=<config>	Domain	Name of the host configuration file to use. The default is **host.xml**.
--interprocess-hc-address=<address>	Domain	Address on which the host controller should listen for communication from the process controller.
--interprocess-hc-port=<port>	Domain	Port on which the host controller should listen for communication from the process controller.
--master-address=<address>	Domain	Set system property **jboss.domain.master.address** to the given value. In a default slave Host Controller config, this is used to configure the address of the master Host Controller.
--master-port=<port>	Domain	Set system property **jboss.domain.master.port** to the given value. In a default slave Host Controller config, this is used to configure the port used for native management communication by the master Host Controller.
--read-only-server-config=<config>	Standalone	Name of the server configuration file to use. This differs from ***--server-config*** and ***-c*** in that the original file is never overwritten.
--read-only-domain-config=<config>	Domain	Name of the domain configuration file to use. This differs from ***--domain-config*** and ***-c*** in that the initial file is never overwritten.
--read-only-host-config=<config>	Domain	Name of the host configuration file to use. This differs from ***--host-config*** in that the initial file is never overwritten.
-P <url>, **-P=<url>**, **--properties=<url>**	Standalone, Domain	Load system properties from the given URL.
--pc-address=<address>	Domain	Address on which the process controller listens for communication from processes it controls.
--pc-port=<port>	Domain	Port on which the process controller listens for communication from processes it controls.
-S<name>[=<value>]	Standalone	Set a security property.
--server-config=<config>	Standalone	Name of the server configuration file to use. The default is **standalone.xml**.

Argument or Switch	Mode	Description
`-u <value>`, **`-u=<value>`**	Standalone, Domain	Set system property **`jboss.default.multicast.address`** to the given value.
`-v`, **`-V`**, **`--version`**	Standalone, Domain	Display the application server version and exit.

Report a bug

CHAPTER 5. CONFIGURATION REQUIREMENTS

The following sections describe modifications to be made to the server configuration to comply with CC requirements. When changes are made via the management console or management CLI, the existing configuration is automatically backed up. You can use those backups for reference or to revert to an earlier configuration if required. See the *Administration and Configuration Guide* for further details of this feature.

Report a bug

5.1. WS-TRANSACTION SUPPORT

WS-Transaction support is provided through the XTS subsystem in JBoss EAP 6.2.2. This subsystem is not enabled by default and must not be enabled as WS-Transaction support is not allowed in the common certified configuration.

Report a bug

5.2. VIRTUAL FILE SYSTEM (VFS) CONFIGURATION

To ensure compliance with Common Criteria requirements, the following configuration for the Virtual File System (VFS) must be applied.

Procedure 5.1. Configure VFS for Common Criteria Requirements

1. For a zip installation, modify ***EAP_HOME*/modules/system/layers/base/.overlays/layer-base-jboss-eap-6.2.2.CP/org/jboss/as/server/main/module.xml**:

   Or, for a RPM from ISO installation, modify **/usr/share/jbossas/modules/system/layers/base/org/jboss/as/server/main/module.xml**:

   Replace the line:

   ```
   <module name="org.jboss.vfs"/>
   ```

   with:

   ```
   <module name="org.jboss.vfs" services="import"/>
   ```

2. For a zip installation, modify ***EAP_HOME*/modules/system/layers/base/org/jboss/as/standalone/main/module.xml**:

   Or, for a RPM from ISO installation, modify **/usr/share/jbossas/modules/system/layers/base/org/jboss/as/standalone/main/module.xml**:

   Add the VFS configuration within the **<dependencies>** element:

   ```
   <dependencies>
     ...
   ```

```
  <module name="org.jboss.vfs" services="import"/>
</dependencies>
```

Report a bug

5.3. NETWORK CONFIGURATION

5.3.1. Network Interfaces

The following network interfaces are defined and created so that trusted and untrusted network traffic is separated.

Network Interfaces

public

For communication to and from external, potentially untrusted parties.

cluster

For communication between cluster nodes. Cannot be accessed by untrusted parties. This must be enforced as part of network/firewall configuration.

internal

For communication between trusted servers or users (such as administrators) via the internal network. Cannot be accessible to untrusted parties or general users of the system.

Report a bug

5.3.2. Network Interface Configuration

To ensure compliance with Common Criteria requirements, JBoss EAP 6.2.2 must have the following network configurations applied.

Procedure 5.2. Define and Configure Network Interfaces

1. **Remove `unsecure` Network Interface**

   ```
   [standalone@localhost:9999 /] /interface=unsecure/:remove
   ```

2. **Create `internal` and `cluster` Network Interfaces**

   ```
   [standalone@localhost:9999 /] /interface=cluster/:add(inet-
   address=expression "${jboss.bind.address.cluster:127.0.0.1}")
   [standalone@localhost:9999 /] /interface=internal/:add(inet-
   address=expression "${jboss.bind.address.internal:127.0.0.1}")
   ```

3. **Bind each socket to the specified network interface.**
   For each line in Table 5.1, “Network Bindings”, use the following command to bind the socket to the specified network interface. Refer to the *Administration and Configuration* Guide.

```
[standalone@localhost:9999 /] /socket-binding-group=standard-
sockets/socket-binding=BINDING_NAME/:write-
attribute(name=interface,value=NETWORK_INTERFACE)
```

For example, the following command binds **management-native** to the **internal** network interface:

```
[standalone@localhost:9999 /] /socket-binding-group=standard-
sockets/socket-binding=management-native/:write-
attribute(name=interface,value=internal)
```

4. **Restart JBoss EAP**
   Restart JBoss EAP so that the network bindings take effect.

Table 5.1. Network Bindings

Binding Name	Network Interface	Port Number
management-native	internal	9999
management-http	internal	9990
management-https	internal	9443
ajp	internal	8009
http	public	8080
https	public	8443
jacorb	internal	3528
jacorb-ssl	internal	3529
jgroups-mping	cluster	0 - multicast: 45700
jgroups-tcp	cluster	7600
jgroups-tcp-fd	cluster	57600
jgroups-udp	cluster	55200 - multicast: 45688
jgroups-udp-fd	cluster	54200
messaging	internal	5445
messaging-group	internal	0 - multicast: 9876
messaging-throughput	internal	5455

Binding Name	Network Interface	Port Number
modcluster	public/internal	0 - multicast: 23364
remoting	internal	4447
txn-recovery-environment	internal	4712
txn-status-manager	internal	4713

Report a bug

5.4. SECURITY CONFIGURATION

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

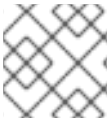
Report a bug

5.4.1. About Authorization

Authorization is a mechanism for granting or denying access to a resource based on identity. It is implemented as a set of declarative security roles which can be granted to principals.

JBoss EAP 6 uses a modular system to configure authorization. Each security domain can contain one or more authorization policies. Each policy has a basic module which defines its behavior. It is configured through specific flags and attributes. The easiest way to configure the authorization subsystem is by using the web-based management console.

Authorization is different from authentication, and usually happens after authentication. Many of the authentication modules also handle authorization.



NOTE

XACML is not permitted in the Common Criteria Certified configuration.

Report a bug

5.4.2. Java Security Manager Policy File

To operate JBoss EAP 6.2 according to the requirements of the certification, you must install the Common Criteria-evaluated Java Security Manager policy to ensure applications running on the system have the correct access privileges.

Procedure 5.3. Install Common Criteria-evaluated Security Manager Policy

1. **Create the Common Criteria-evaluated Policy File**

Copy the following text into a text editor and save it in ***EAP_HOME*/bin/** directory. The suggested file name is **jbosseap62.policy**.

```
// Grant all to the jboss-modules.jar
grant codeBase "file:${jboss.home.dir}/jboss-modules.jar" {
  permission java.security.AllPermission;
};

// Standard extensions get all permissions by default
grant codeBase "file:${java.home}/lib/ext/*" {
  permission java.security.AllPermission;
};
```

NOTE

The CC-evaluated policy file may need additional permissions configured, such as permissions to database drivers. These permissions are site-specific.

2. **Configure the Java Parameters**
   Add the following Java command line parameters to the JBoss EAP start-up configuration scripts (standalone.conf, domain.conf, standalone.conf.bat, domain.conf.bat):

   -Djava.security.manager

   Enables the security manager

   -Djava.security.policy==/path/to/security.policy

   Specify the path to the security policy file.

   -Djboss.home.dir=/path/to/JBOSS_EAP_HOME

   Define the system property that is used in the policy file.

   -Djboss.modules.policy-permissions=true

   Enables deployment level security permissions.

   Example 5.1. standalone.conf

   ```
   JAVA_OPTS="$JAVA_OPTS -Djava.security.manager -
   Djava.security.policy==$JBOSS_HOME/bin/jbosseap62.policy -
   Djboss.home.dir=$JBOSS_HOME -Djboss.modules.policy-
   permissions=true"
   ```

NOTE

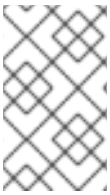
The JBOSS_HOME environment variable is not defined when domain.conf is processed. It can be used only in standalone.conf.

3. **Configure the Java Security Manager to use the policy file**
   See *Java Security Manager* in the *Security Guide*.

Report a bug

5.4.3. Enable Audit Logging

To enable audit logging to record authentication and authorization information for every thread and EJB call, start the CLI management console and follow this procedure.



NOTE

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 5.4. Enable Audit Logging

1. Create a periodic rotating file handler named **`AUDIT`**. The format of log file must be defined with this format to be common criteria compliant.

```
/subsystem=logging/periodic-rotating-file-
handler=AUDIT/:add(suffix=.yyyy-MM-dd,formatter=%d{HH:mm:ss,SSS} %-
5p [%c] (%t)
%s%E%n,level=TRACE,file={"relative-to" =>
"jboss.server.log.dir","path" => "audit.log"})
```

2. Create a logger category for the JBoss EAP logging subsystem.

```
/subsystem=logging/logger=org.jboss.security.audit/:add(level=TRACE,
category=org.jboss.security.audit,handlers=["AUDIT"])
```

3. Enable audit logging in each application by using the **`jboss-web.xml`** descriptor located in the **`WEB-INF`** directory, setting the tag ***`disable-audit`*** to **`false`**.

```
<?xml version="1.0" encoding="UTF-8"?>
  <jboss-web>
    <security-domain>security_domain_for_the_app</security-domain>
     <disable-audit>false</disable-audit>
  </jboss-web>
```

Report a bug

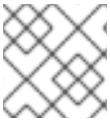
5.4.4. Management Interface Audit Logging

5.4.4.1. About Management Interface Audit Logging

Audit logging can be enabled, so that operations carried out via the management API are recorded in an audit log. Whether those operations are carried out via the management console, management CLI interface, or a custom-written interface, all are subject to audit logging. Log records can be output to a file, forwarded to a Syslog server or both. By default, audit logging is disabled.

Logging data is output in JSON format, with several configuration options available for influencing the operations included in the log and the log entries' format.

Before being stored, log entries pass through a formatter and handler. The formatter specifies the format of the log entries, while the handler outputs the records to the specified destination(s). Only one formatter is currently available, which outputs entries in JSON format.



NOTE

Audit logging can be configured only via the management CLI.

Report a bug

5.4.4.2. Enable Management Interface Audit Logging from the Management CLI

To enable audit logging from the management CLI, use the following command.

```
/core-service=management/access=audit/logger=audit-log:write-
attribute(name=enabled,value=true)
```

Audit logging is preconfigured to output to the file *EAP_HOME*/standalone/data/audit-log.log.

Report a bug

5.4.4.3. About a Management Interface Audit Logging Formatter

The formatter specifies the format of the log entries.

Table 5.2. JSON Formatter Fields

Attribute	Description
include-date	Boolean value which defines whether or not the timestamp is included in the formatted log records.
date-separator	A string containing characters to separate the date and the rest of the formatted log message. Will be ignored if ***include-date***=**false**.
date-format	The date format to use for the timestamp as understood by java.text.SimpleDateFormat. Ignored if ***include-date***=**false**.
compact	If **true** it will format the JSON on one line. There may still be values containing new lines, so if having the whole record on one line is important, set ***escape-new-line*** or ***escape-control-characters*** to **true**.
escape-control-characters	If **true** it will escape all control characters (ASCII entries with a decimal value < 32) with the ASCII code in octal; for example, a new line becomes '#012'. If this is **true**, it will override ***escape-new-line***=**false**.

Attribute	Description
escape-new-line	If true it will escape all new lines with the ASCII code in octal; for example **#012**.

Report a bug

5.4.4.4. About a Management Interface Audit Logging File Handler

A file handler specifies the parameters by which audit log records are output to a file. Specifically it defines the formatter, file name and path for the file.

Table 5.3. File Handler Audit Log Fields

Attribute	Description	Read Only
formatter	The name of a JSON formatter to use to format the log records.	False
path	The path of the audit log file.	False
relative-to	The name of another previously named path, or of one of the standard paths provided by the system. If ***relative-to*** is provided, the value of the path attribute is treated as relative to the path specified by this attribute.	False
failure-count	The number of logging failures since the handler was initialized.	True
max-failure-count	The maximum number of logging failures before disabling this handler.	False
disabled-due-to-failure	**true** if this handler was disabled due to logging failures.	True

Report a bug

5.4.4.5. About a Management Interface Audit Logging Syslog Handler

A syslog handler specifies the parameters by which audit log entries are sent to a syslog server, specifically the syslog server's hostname and the port on which the syslog server is listening.

Sending audit logging to a syslog server provides more security options than logging to a local file or local syslog server. Multiple syslog handlers can be defined.

Syslog servers vary in their implementation, so not all settings are applicable to all syslog servers. Testing has been conducted using the *rsyslog* syslog implementation. The referenced RFCs are:

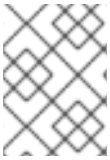
- **http://www.ietf.org/rfc/rfc3164.txt**
- **http://www.ietf.org/rfc/rfc5424.txt**
- **http://www.ietf.org/rfc/rfc6587.txt**

Table 5.4. Syslog Handler Fields

Field	Description	Read-only
formatter	The name of the formatter to use to format the log records.	False
failure-count	The number of logging failures since the handler was initialized	True
max-failure-count	The maximum number of logging failures before disabling this handler.	False
disabled-due-to-failure	**True** if this handler was disabled due to logging failures.	True
syslog-format	Syslog format: *RFC-5424* or *RFC-3164*.	False
max-length	The maximum length of a log message (in bytes), including the header. If undefined, it will default to **1024** bytes if the **syslog-format** is **RFC3164**, or **2048** bytes if the **syslog-format** is **RFC5424**.	False.
truncate	Whether or not a message, including the header, should truncate the message if the length in bytes is greater than the maximum length. If set to false messages will be split and sent with the same header values.	False

Report a bug

5.4.4.6. Enable Management Interface Audit Logging to a Syslog Server



NOTE

Add the prefix **/host=*HOST_NAME*** to the **/core-service** commands if the change is to be applied to a managed domain.

Procedure 5.5. Enable Logging to a Syslog Server

1. **Create a syslog handler named `msyslog`**

```
[standalone@localhost:9999 /]batch
[standalone@localhost:9999 /]/core-
service=management/access=audit/syslog-
handler=mysyslog:add(formatter=json-formatter)
[standalone@localhost:9999 /]/core-
service=management/access=audit/syslog-
handler=mysyslog/protocol=udp:add(host=localhost,port=514)
[standalone@localhost:9999 /]run-batch
```

2. **Add a reference to the syslog handler.**

```
[standalone@localhost:9999 /]/core-
service=management/access=audit/logger=audit-
log/handler=mysyslog:add
```

Result

Management interface audit log entries are logged on the syslog server.

Report a bug

5.4.4.7. Management Interface Audit Logging Options

In addition to enabling or disabling management interface audit logging, other configuration options are available.

Configuration Options

log-boot

If set to **`true`**, management operations when booting the server are included in the audit log, **`false`** otherwise. Default: **`false`**.

log-read-only

If set to **`true`**, all operations will be audit logged. If set to **`false`** only operations that change the model will be logged. Default: **`false`**.

Report a bug

5.4.4.8. Management Interface Audit Log Fields

Table 5.5. Management Interface Audit Log Fields

Field Name	Description

Field Name	Description
type	This can have the values **`core`**, meaning it is a management operation, or **`jmx`** meaning it comes from the JMX subsystem (see the JMX subsystem for configuration of the JMX subsystem's audit logging).
r/o	**`true`** if the operation does not change the management model, **`false`** otherwise.
booting	**`true`** if the operation was executed during the bootup process, **`false`** if it was executed once the server is up and running.
version	Version number of the JBoss EAP instance.
user	Username of the authenticated user. If the operation has been logged via the CLI on the same computer as the running server, the special **`$local`** user is used.
domainUUID	An identifier to link together all operations as they are propagated from the doman controller to its servers, slave host controllers, and slave host controller servers.
access	This can have one of the following values: NATIVE, HTTP, JMX. NATIVE - The operation came in through the native management interface, for example the CLI. HTTP - The operation came in through the domain HTTP interface, for example the admin console. JMX - The operation came in through the JMX subsystem. See JMX for how to configure audit logging for JMX.
remote-address	The address of the client executing this operation.
success	**`true`** if the operation succeeded, **`false`** if it was rolled back.
ops	The operations being executed. This is a list of the operations serialized to JSON. At boot this will be all the operations resulting from parsing the XML. Once booted the list will typically contain a single entry.

Report a bug

5.4.5. PicketBox

Only the following login modules are allowed to be configured and used for authentication purposes:

- File based authentication using UsersRolesLoginModule
- File based authentication for EJB Remoting Framework using RemotingLoginModule
- Certificate based authentication using BaseCertLoginModule
- LDAP based authentication using LdapLoginModule
- Advanced LDAP based authentication using LdapExtLoginModule
- Database based authentication using DatabaseServerLoginModule

Report a bug

5.4.6. EJB Authorization Policy

Applications can implement custom authentication and authorization verification using a JACC Authorization Module. In JBoss EAP 6.2.2, 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.

- `code=Delegating`
- `code=JACC`

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

Report a bug

5.5. DATABASE CONFIGURATION

NOTE

For better startup server behavior, the preferred installation method for JDBC drivers is to install them as a core module.

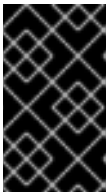
Procedure 5.6. Configure Database

1. **Remove ExampleDS and h2 database driver**
   Using the CLI, execute the following commands to remove the example DS, and h2 database driver:

   ```
   [standalone@localhost:9999 /] /subsystem=datasources/data-
   source=ExampleDS/:remove
   [standalone@localhost:9999 /] /subsystem=datasources/jdbc-
   driver=h2/:remove
   ```

2. **Add JDBC Grant Statement**
   Add the following **`grant`** statement to the Java Security Manager policy file for the JDBC driver you are using. The policy file is located at ***`EAP_HOME`*`/bin/eap62.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.home.dir}/standalone/deployments/[cc.jdbc.driver]" {
  permission java.security.AllPermission;
};
```

Report a bug

5.6. 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 6, *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

For example, 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/TestDeployment.ear/-"
{
 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

Report a bug

CHAPTER 6. 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.

Report a bug

6.1. ENTERPRISE APPLICATION

An enterprise application is a Java Enterprise Edition (formerly J2EE) version 1.6 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 1.6 Web Service Applications which can be based on Stateless EJBs or Plain Old Java Objects (POJOs) deployed as Java Servlets.

Report a bug

6.2. 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*.
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.
- The enterprise bean must not use Simple and Protected GSSAPI Negotiation Mechanism (SPNEGO).
- The enterprise bean must not use annotations from PicketBox. The following annotations that modify the behavior of the JAAS module must not be used:

- **`@AuthenticationMechanism`**
- **`@SecurityMapping`**
- **`@Authentication`**
- **`@Authorization`**
- **`@SecurityConfig`**
- **`@SecurityAudit`**

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.

Report a bug

6.3. 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 6.1. 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 6.2. Retrieving Principal and Credential

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

Report a bug

CHAPTER 7. OVERVIEW OF THE SECURITY FUNCTIONS

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

Report a bug

7.1. ACCESS CONTROL

JBoss EAP 6.2.2 has access control mechanisms to restrict access for the following request types:

HTTP

URLs and paths provided with URLs, as well as Plain Old Java Objects (POJOs) deployed as Servlets and Session Beans, can be protected from access by subjects.

EJB

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

HornetQ

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

For more information see the *Administration and Configuration Guide*.

Report a bug

7.2. ROLE-BASED ACCESS CONTROL FOR MANAGEMENT INTERFACES

The management interfaces of JBoss EAP 6.2.2 (the command line interface and the web-based administrative interface) allow access to the JBoss EAP system configuration in order to manage all configurable aspects of JBoss EAP 6.2.2. Administrators can access general system aspects, such as network port configurations, and container configurations. In addition, configuration aspects for services offered by containers are managed as well.

The configuration of applications, such as the application access control, are addressed in the deployment descriptors shipped with the application. Therefore, application configuration is generally not accessible via the management interfaces.

The administrative interfaces can be bound to a specific network interface. This allows for the management interfaces to be restricted to an administration LAN in order to prevent untrusted users from accessing the management interfaces. In order for administrators to interact with administrative interfaces, they must log in. Administration accounts are maintained separately from other user accounts.

Each action on an object that an administrative user can perform is subject to a role-based access control mechanism. The actions are classified into:

- Model operations, whose the main function is to read/write from the model, although there will often be associated runtime services started/stopped as a consequence.
- RPC operations, which invokes some runtime that affects runtime state only. This may either read runtime state or change it. The model is not affected.

- A resource.
- An attribute residing in a resource.

A set of object-action capabilities are mapped to a management role. This mapping defines the allowed access for the management role. A set of pre-defined management roles is included with JBoss EAP 6.2.2 and is available after installation. The pre-configured roles are detailed below.

Role-based access control pre-configured management roles

Monitor

The monitor role has the fewest permissions and restricts the user to viewing the configuration and the current state. The monitor role does not have permission to view sensitive data.

Configurator

The configurator role has the same permissions as the monitor role, and can change the configuration. For example, the configurator can deploy an application. The configurator role does not have permission to view sensitive data.

Operator

The operator role has monitor permissions and can also change the runtime state, but not the persistent configuration. For example, the operator can start or stop servers. The operator role does not have permission to view sensitive data.

Administrator

The administrator role has the combined permissions of the operator and the configurator. This role has also permission to access sensitive data, including passwords. The administrator role is the superuser of the Application Server and can modify administrative users and roles.

Deployer

The deployer role has the combined permissions of the operator and the configurator, but with those permissions constrained to operating on deployments.

Auditor

The auditor role can view and modify the configuration settings for the security auditing system. The auditor role includes the monitor role, allowing the auditor to view but not change the rest of the security configuration.

A role is a named set of permissions. These permissions include constraints, for example the read permissions for the Monitor role is constrained to non-sensitive actions and targets. Redefinition of the permissions and constraints associated with the above mentioned standard roles is not permitted.

A limited form of creation of new roles is allowed. These new roles are equivalent to the standard roles, but with an additional constraint applied to all permission, for example the target must be related to a particular host or server group.

All administrative operations are stored in configuration files (either **`domain.xml`** or **`standalone.xml`** depending on the startup mode). The administrative interfaces are an in-memory image of the data stored in the configuration file. Once the in-memory image is modified, the modified configuration file is stored.

The role-based access control mechanism can only be enforced if the administrator accesses the JBoss

EAP 6.2.2 system configuration via the CLI or management interface. If an administrator has shell access to the host, the underlying operating system may grant direct read or write access to the JBoss EAP system configuration files. Such access would imply that the role-based access control mechanism is not enforced. It is assumed that the host is located in a protected environment where direct access to the JBoss EAP system configuration files is not allowed.

Report a bug

7.3. AUDIT

JBoss EAP 6.2.2 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 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. Audit records can also be forwarded to a syslog server for additional audit controls.

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 explicitly record 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.

Report a bug

7.4. CLUSTERING

A cluster is a set of nodes. In a JBoss EAP 6 cluster, a node is a JBoss EAP 6 server instance. To build a cluster, several JBoss EAP 6 instances have to be grouped together, also known as a "partition").

Clustering allows the execution of applications on several parallel nodes. Two cluster concepts are possible with JBoss EAP 6: a failover cluster, and a load-distribution cluster. In both cases, the server state is distributed across different servers, and if any server fails, the application is still accessible via other cluster nodes.

Cluster communication establishes data consistency between different cluster nodes. JGroups and Infinispan provide the underlying communication, node replication, and caching services for JBoss EAP 6 clusters. These services are configured as MBeans. There is a set of Infinispan and JGroups MBeans for each type of clustering applications (for example, the Stateful Session EJBs, the distributed entity EJBs, etc.).

Infinispan provides distributed cache and state replication services for the JBoss EAP 6 cluster. A JBoss EAP 6 cluster can have multiple Infinispan MBeans: one for HTTP session replication, one for stateful session beans, one for cached entity beans, and so on.

The following information is replicated as part of cluster communication:

- Replication of the state of a node includes the replication of HTTP sessions, EJB 3.0 session beans, EJB 3.0 entity beans, as well as Hibernate persistence objects (distributed state replication service using Infinispan).
- Replication of the state of a node covering the replication of HTTP sessions, and EJB 2.x session beans.
- Replication of HornetQ queues. Messages sent to a distributed queue or topic on one node are consumable on other nodes.

JBoss EAP 6 does not perform an automated replication of the JNDI state. When applications defining JNDI resources are replicated to different cluster nodes, they are newly deployed at the nodes. With this deployment, the JNDI resources are created similar to a regular deployment. System configuration changes that involve modifications of JNDI resources are replicated to the cluster nodes, and applied similarly to a local reconfiguration. The JNDI registry maintaining the JNDI mappings is managed consistently between the different cluster nodes. As JNDI does not maintain a state other than the JNDI registry, this is sufficient to ensure cluster-wide consistency of the JNDI service.

Report a bug

7.5. IDENTIFICATION AND AUTHENTICATION

Users are assigned unique user identifiers which are used as the basis for access control decisions and auditing. JBoss EAP 6.2.2 authenticates the identity of the user before allowing the user to perform any further security-mediated actions. JBoss EAP 6.2.2 internally maintains the identifier associated with the thread spawned for a user after a successful authentication.

JBoss EAP 6.2.2 provides different identification and authentication mechanisms for different request types:

HTTP and webservices

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

EJB

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

HornetQ

Username and password-based authentication.

JBoss EAP 6.2.2 implements identification and authentication using Java Authentication and Authorization Service (JAAS) with the PicketBox framework. JAAS is provided by the Java virtual machine in the operational environment. The PicketBox framework uses only the authentication capabilities of JAAS to implement the declarative role-based Java EE security model.

JAAS authentication is performed in a pluggable fashion. This permits Java applications to remain independent from the underlying authentication technologies, and allows the PicketBox security manager to work in different security infrastructures. Integration with a security infrastructure can be achieved without changing the PicketBox security manager implementation. This is done by changing the configuration of the authentication stack that JAAS uses. JBoss EAP 6.2.2 provides the JAAS modules which are called by the JAAS framework to perform identification and authentication.

The basic security interfaces required for implementation of the JAVA EE security model are not heavily dependent on JAAS. The PicketBox framework is an implementation of the basic security plug-in

interfaces that are based on JAAS. PicketBox provides an abstraction layer, which is based on JAAS, to other containers of JBoss EAP 6.2.2. An administrator is able to replace the JAAS-based PicketBox implementation classes with an individual custom security manager implementation that does not use JAAS, however the evaluated configuration prohibits the replacement of PicketBox.

The following authentication backends and corresponding JAAS modules are allowed to be configured:

- File-based authentication using **`UsersRolesLoginModule`**.
- File-based authentication for EJB Remoting Framework using **`RemotingLoginModule`**.
- Certificate-based authentication using **`BaseCertLoginModule`**.
- LDAP-based authentication using **`LdapLoginModule`**.
- Advanced LDAP-based authentication using **`LdapExtLoginModule`**.
- Database based-authentication using **`DatabaseServerLoginModule`**.

Password quality used can be enforced with configuration options for the JAAS modules provided by JBoss EAP 6.2.2.

If the JAAS login authenticates the user, a JAAS Subject is created that contains the following in its **`PrincipalsSet`**:

- A **`java.security.Principal`** that corresponds to the client identity as known in the deployment security environment.
- A **`java.security.acl.Group`** named **`Roles`** that contains the role names from the application domain to which the user has been assigned. **`org.jboss.security.SimplePrincipal`** objects, or custom objects registered as **`principalClass`**, are used to represent the role names. **`SimplePrincipal`** is a simple string-based implementation of **`Principal`**. These roles are used to validate the roles assigned to methods in **`ejb-jar.xml`** and the **`EJBContext.isCallerInRole(String)`** method implementation.

The above mentioned network protocols tunnel client requests to JBoss EAP 6.2.2. After identification and authentication checks are performed, the request is forwarded to the intended application. As JBoss EAP 6.2.2 uses only the credential information from the network request, only the aspect of communicating the user credentials is relevant for the enforcement of the identification and authentication policy.

JBoss EAP 6.2.2 allows the management of authorization independently for each application and service. The mentioned deployment descriptors and annotations can be used by authorized administrators to configure the identification and authentication mechanism. JBoss EAP 6.2.2 provides the interfaces for managing identification and authentication policy, however it does not restrict the use of the interfaces to authorized administrators. These settings are stored in the JBoss EAP 6.2.2 system configuration. This configuration file could be accessed by users who have access to write permissions on the host system.

Report a bug

7.6. TRANSACTION ROLLBACK

JBoss EAP 6.2.2 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 EAP 6.2.2 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.

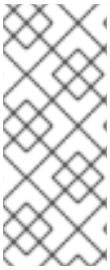
JBoss EAP 6.2.2 supports both local and distributed transactions. A transaction is considered to be distributed if it spans multiple process instances, i.e. virtual machines (VMs). Typically a distributed transaction will contain participants that are located within multiple VMs but the transaction is 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.

Report a bug

APPENDIX A. NETWORK PORTS USED BY JBOSS EAP 6

The ports used by the JBoss EAP 6 default configuration depend on several factors:

- Whether your server groups use one of the default socket binding groups, or a custom group.
- The requirements of your individual deployments.



NOTE

A numerical port offset can be configured, to alleviate port conflicts when you run multiple servers on the same physical server. If your server uses a numerical port offset, add the offset to the default port number for its server group's socket binding group. For instance, if the HTTP port of the socket binding group is **`8080`**, and your server uses a port offset of **`100`**, its HTTP port is **`8180`**.

Unless otherwise stated, the ports use the TCP protocol.

The default socket binding groups

- **`full-ha-sockets`**
- **`full-sockets`**
- **`ha-sockets`**
- **`standard-sockets`**

Table A.1. Reference of the default socket bindings

Name	Port	Multicast Port	Description	full-ha-sockets	full-sockets	ha-socket	standard-socket
`ajp`	8009		Apache JServ Protocol. Used for HTTP clustering and load balancing.	Yes	Yes	Yes	Yes
`http`	8080		The default port for deployed web applications.	Yes	Yes	Yes	Yes
`https`	8443		SSL-encrypted connection between deployed web applications and clients.	Yes	Yes	Yes	Yes

Name	Port	Multicast Port	Description	full-ha-sockets	full-sockets	ha-socket	standard-socket
jacorb	3528		CORBA services for JTS transactions and other ORB-dependent services.	Yes	Yes	No	No
jacorb-ssl	3529		SSL-encrypted CORBA services.	Yes	Yes	No	No
jgroups-diagnostics		7500	Multicast. Used for peer discovery in HA clusters. Not configurable using the Management Interfaces.	Yes	No	Yes	No
jgroups-mping		45700	Multicast. Used to discover initial membership in a HA cluster.	Yes	No	Yes	No
jgroups-tcp	7600		Unicast peer discovery in HA clusters using TCP.	Yes	No	Yes	No
jgroups-tcp-fd	57600		Used for HA failure detection over TCP.	Yes	No	Yes	No
jgroups-udp	55200	45688	Unicast peer discovery in HA clusters using UDP.	Yes	No	Yes	No
jgroups-udp-fd	54200		Used for HA failure detection over UDP.	Yes	No	Yes	No
messaging	5445		JMS service.	Yes	Yes	No	No
messaging-group			Referenced by HornetQ JMS broadcast and discovery groups.	Yes	Yes	No	No

Name	Port	Multicast Port	Description	full-ha-sockets	full-sockets	ha-socket	standard-socket
messaging-throughput	5455		Used by JMS Remoting.	Yes	Yes	No	No
mod_cluster		23364	Multicast port for communication between JBoss EAP 6 and the HTTP load balancer.	Yes	No	Yes	No
osgi-http	8090		Used by internal components which use the OSGi subsystem. Not configurable using the Management Interfaces.	Yes	Yes	Yes	Yes
remoting	4447		Used for remote EJB invocation.	Yes	Yes	Yes	Yes
txn-recovery-environment	4712		The JTA transaction recovery manager.	Yes	Yes	Yes	Yes
txn-status-manager	4713		The JTA / JTS transaction manager.	Yes	Yes	Yes	Yes

Management Ports

In addition to the socket binding groups, each host controller opens two more ports for management purposes:

- **9990** - The Web Management Console port
- **9999** - The port used by the Management Console and Management API

Additionally, if HTTPS is enabled for the Management Console, 9443 is also opened as the default port.

Report a bug

APPENDIX B. MODULES

A Module is a logical grouping of classes used for class loading and dependency management. JBoss EAP 6 identifies two different types of modules, sometimes called static and dynamic modules. However the only difference between the two is how they are packaged. All modules provide the same features.

Static Modules

Static Modules are predefined in the ***EAP_HOME*/modules/** directory of the application server. Each sub-directory represents one module, and contains a configuration file (**module.xml**) and any required JAR files. The name of the module is defined in the **module.xml** file. All the application server provided APIs are provided as static modules, including the Java EE APIs as well as other APIs such as JBoss Logging.

Example B.1. Example module.xml file

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.0" name="com.mysql">
  <resources>
    <resource-root path="mysql-connector-java-5.1.15.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

The module name, **com.mysql**, should match the directory structure for the module.

Creating custom static modules can be useful if many applications are deployed on the same server that use the same third party libraries. Instead of bundling those libraries with each application, a module containing these libraries can be created and installed by the JBoss administrator. The applications can then declare an explicit dependency on the custom static modules.

Dynamic Modules

Dynamic Modules are created and loaded by the application server for each JAR or WAR deployment (or subdeployment in an EAR). The name of a dynamic module is derived from the name of the deployed archive. Because deployments are loaded as modules, they can configure dependencies and be used as dependencies by other deployments.

Modules are only loaded when required. This usually only occurs when an application is deployed that has explicit or implicit dependencies.

Report a bug

APPENDIX C. REVISION HISTORY

Revision 6.2.2-15	**Tue June 24 2014**	**Tom Wells, Russell Dickenson, Lucas Costi**

Red Hat JBoss Enterprise Application Platform Common Criteria Certification