



Red Hat Fuse 7.2

Installing on JBoss EAP

Install Fuse 7.2 on JBoss EAP 7.1

Red Hat Fuse 7.2 Installing on JBoss EAP

Install Fuse 7.2 on JBoss EAP 7.1

Legal Notice

Copyright © 2019 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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, the Red Hat 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 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

Use this guide to help you install Red Hat Fuse on JBoss EAP.

Table of Contents

CHAPTER 1. INSTALL FUSE ON JBOSS EAP	3
PREREQUISITES	3
INSTALL FUSE USING INSTALLER	3
CHAPTER 2. START AND STOP THE APPLICATION SERVER	4
2.1. START JBOSS EAP 7.1	4
2.2. STOP JBOSS EAP 7.1	4
CHAPTER 3. VERIFY YOUR RED HAT FUSE INSTALLATION	6
ADD A USER ID TO FUSE ON FUSE ON JBOSS EAP	6
APPENDIX A. PREPARING TO USE MAVEN	8
A.1. OVERVIEW	8
A.2. PREREQUISITES	8
A.3. ADDING THE RED HAT MAVEN REPOSITORIES	8
A.4. ARTIFACTS	10
A.5. MAVEN COORDINATES	10

CHAPTER 1. INSTALL FUSE ON JBOSS EAP

PREREQUISITES

Before you install Fuse:

- You must have installed JBoss EAP 7.1 along with any recommended patch version.
 - If you need to install JBoss EAP 7.1, download it from [JBoss EAP 7.1 Installer Download](#). See [JBoss EAP 7.1 Installation Guide](#) for installation instructions.
 - To determine which patch version of JBoss EAP to install, consult the [Supported Configurations](#) page.
- It is recommended that you use Maven with Red Hat Fuse projects. For information about preparing to use Maven, see [Appendix A, Preparing to use Maven](#).



IMPORTANT

Fuse on JBoss EAP is not supported with Domain mode.



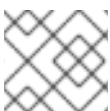
WARNING

Red Hat recommends that you back up your system settings and data before undertaking any of the configuration tasks mentioned in this book.

INSTALL FUSE USING INSTALLER

1. Download the *Red Hat Fuse 7.2 on EAP Installer* package:
 - a. Browse to the [Red Hat Fuse Software Downloads](#) page on the Red Hat Customer Portal and, when prompted, log in to your customer account.
 - b. Select version **7.2** from the **Version** dropdown menu and click the **Download** link for the **Red Hat Fuse 7.2 on EAP Installer** package.
2. Navigate to `$EAP_HOME` in a clean instance of JBoss EAP, where **EAP_HOME** is the root directory of the Red Hat JBoss Enterprise Application Platform installation on which Fuse is deployed.
3. Run the downloaded installer with the following command:

```
java -jar TEMP_LOCATION/fuse-eap-installer-7.2.0.fuse-720018-redhat-00002.jar
```

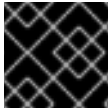


NOTE

Once a datastore has been selected at installation, it cannot be changed.

CHAPTER 2. START AND STOP THE APPLICATION SERVER

You need to start the JBoss Enterprise Application Platform instance for Fuse to run. This is because the Fuse components run on the JBoss Enterprise Application Platform container.



IMPORTANT

Fuse on JBoss EAP is not supported with Domain mode.



NOTE

For more information about starting and stopping JBoss Enterprise Application Platform using alternative and more advanced methods, see the [Red Hat JBoss Enterprise Application Platform Configuration Guide](#).

2.1. START JBOSS EAP 7.1

You can start JBoss EAP 7.1 as a standalone server.

Start the Platform Service as a Standalone Server

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

2.2. STOP JBOSS EAP 7.1

You can stop JBoss EAP using the Management CLI or by pressing **CTRL+C** in the terminal.

To stop JBoss EAP using the Management CLI:

1. Launch the Management CLI by running the **`EAP_HOME/bin/jboss-cli.sh`** command:

```
$ EAP_HOME/bin/jboss-cli.sh
```

2. Connect to the server by running the **`connect`** command:

```
[disconnected /] connect
```

3. Stop the server by running the **`shutdown`** command:

```
[standalone@localhost:9999 /] shutdown
```

4. Close the Management CLI by running the **`quit`** command:

```
[standalone@localhost:9999 /] quit
```


To stop JBoss EAP by pressing **CTRL+C**:

1. Navigate to the terminal where JBoss EAP is running.
2. Press **Ctrl+C** to stop JBoss Enterprise Application Platform.

CHAPTER 3. VERIFY YOUR RED HAT FUSE INSTALLATION

After you complete the [application] **Fuse** installation, you can verify if the product has been installed successfully.

If no error was reported, you can verify the installation by performing the following steps:

1. Start the JBoss EAP server by running the **standalone.sh** command
2. Open the **server.log** file to check if any error messages have been logged and that SwitchYard (and other component subsystems) have loaded correctly
3. Open the Fuse Management Console (<http://localhost:8080/hawtio>) in a browser. Sign in using the administrative user that you set up on the JBoss EAP installer. If the Fuse Management Console runs and you can log in then the installation has been successful.

ADD A USER ID TO FUSE ON FUSE ON JBOSS EAP

To add administrative users to JBoss EAP, use the **add-user** utility script provided with JBoss EAP.

1. Navigate to **`$EAP_HOME/bin`**.
2. Run the **add-user** utility script.
3. Press ENTER to select the default option a to add a management user. All the defaults can be selected by pressing ENTER.
4. Enter a User ID and password. Repeat the password.
5. Enter **yes** to indicate that you want to add the new user ID to the Management Realm.
6. Enter **no** to indicate that the new user ID is not for a remote connection of any kind.

```
[userid@localhost bin] $ ./add-user.sh
```

What type of user do you wish to add?

- a) Management User (mgmt-users.properties)
- b) Application User (application-users.properties)

(a):

Enter the details of the new user to add.

Realm (ManagementRealm) :

Username : manageuser

Password requirements are listed below. To modify these restrictions edit the add-user.properties configuration file.

- The password must not be one of the following restricted values {root, admin, administrator}
- The password must contain at least 8 characters, 1 alphabetic character(s), 1 digit(s), 1 non-alphanumeric symbol(s)
- The password must be different from the username

Password :

Re-enter Password :

What groups do you want this user to belong to? (Please enter a comma separated list, or leave blank for none)[]:

About to add user 'manageuser' for realm 'ManagementRealm'

Is this correct yes/no? yes

Added user 'manageuser' to file '`$EAP_HOME/standalone/configuration/mgmt-users.properties`'

Added user 'manageuser' to file '\$EAP_HOME/domain/configuration/mgmt-users.properties'

Added user 'manageuser' with groups to file '\$EAP_HOME/standalone/configuration/mgmt-groups.properties'

Added user 'manageuser' with groups to file '\$EAP_HOME/domain/configuration/mgmt-groups.properties'

Is this new user going to be used for one AS process to connect to another AS process?

e.g. for a slave host controller connecting to the master or for a Remoting connection for server to server EJB calls.

yes/no? no

For more information about creating users on JBoss EAP see [Adding a Management User](#) in the JBoss EAP Administration and Configuration Guide.

APPENDIX A. PREPARING TO USE MAVEN

A.1. OVERVIEW

This section gives a brief overview of how to prepare Maven for building Red Hat Fuse projects and introduces the concept of Maven coordinates, which are used to locate Maven artifacts.

A.2. PREREQUISITES

In order to build a project using Maven, you must have the following prerequisites:

- **Maven installation** – Maven is a free, open source build tool from Apache. You can download the latest version from the [Maven download page](#).
- **Network connection** – whilst performing a build, Maven dynamically searches external repositories and downloads the required artifacts on the fly. By default, Maven looks for repositories that are accessed over the Internet. You can change this behavior so that Maven will prefer searching repositories that are on a local network.



NOTE

Maven can run in an offline mode. In offline mode Maven only looks for artifacts in its local repository.

A.3. ADDING THE RED HAT MAVEN REPOSITORIES

In order to access artifacts from the Red Hat Maven repositories, you need to add them to Maven's **settings.xml** file. Maven looks for your **settings.xml** file in the **.m2** directory of the user's home directory. If there is not a user specified **settings.xml** file, Maven will use the system-level **settings.xml** file at **M2_HOME/conf/settings.xml**.

To add the Red Hat repositories to Maven's list of repositories, you can either create a new **.m2/settings.xml** file or modify the system-level settings. In the **settings.xml** file, add **repository** elements for the Red Hat repositories as shown in [Adding the Red Hat Fuse Repositories to Maven](#).

Adding the Red Hat Fuse Repositories to Maven

```
<?xml version="1.0"?>
<settings>

<profiles>
<profile>
<id>extra-repos</id>
<activation>
<activeByDefault>true</activeByDefault>
</activation>
<repositories>
<repository>
<id>redhat-ga-repository</id>
<url>https://maven.repository.redhat.com/ga</url>
<releases>
<enabled>true</enabled>
</releases>
```

```

    <snapshots>
      <enabled>>false</enabled>
    </snapshots>
  </repository>
  <repository>
    <id>redhat-ea-repository</id>
    <url>https://maven.repository.redhat.com/earlyaccess/all</url>
    <releases>
      <enabled>>true</enabled>
    </releases>
    <snapshots>
      <enabled>>false</enabled>
    </snapshots>
  </repository>
  <repository>
    <id>jboss-public</id>
    <name>JBoss Public Repository Group</name>
    <url>https://repository.jboss.org/nexus/content/groups/public/</url>
  </repository>
</repositories>
<pluginRepositories>
  <pluginRepository>
    <id>redhat-ga-repository</id>
    <url>https://maven.repository.redhat.com/ga</url>
    <releases>
      <enabled>>true</enabled>
    </releases>
    <snapshots>
      <enabled>>false</enabled>
    </snapshots>
  </pluginRepository>
  <pluginRepository>
    <id>redhat-ea-repository</id>
    <url>https://maven.repository.redhat.com/earlyaccess/all</url>
    <releases>
      <enabled>>true</enabled>
    </releases>
    <snapshots>
      <enabled>>false</enabled>
    </snapshots>
  </pluginRepository>
  <pluginRepository>
    <id>jboss-public</id>
    <name>JBoss Public Repository Group</name>
    <url>https://repository.jboss.org/nexus/content/groups/public</url>
  </pluginRepository>
</pluginRepositories>
</profile>
</profiles>

<activeProfiles>
  <activeProfile>extra-repos</activeProfile>
</activeProfiles>

</settings>

```

A.4. ARTIFACTS

The basic building block in the Maven build system is an *artifact*. The output of an artifact, after performing a Maven build, is typically an archive, such as a JAR or a WAR.

A.5. MAVEN COORDINATES

A key aspect of Maven functionality is the ability to locate artifacts and manage the dependencies between them. Maven defines the location of an artifact using the system of *Maven coordinates*, which uniquely define the location of a particular artifact. A basic coordinate tuple has the form, **{*groupId*, *artifactId*, *version*}**. Sometimes Maven augments the basic set of coordinates with the additional coordinates, *packaging* and *classifier*. A tuple can be written with the basic coordinates, or with the additional *packaging* coordinate, or with the addition of both the *packaging* and *classifier* coordinates, as follows:

```
groupId:artifactId:version
groupId:artifactId:packaging:version
groupId:artifactId:packaging:classifier:version
```

Each coordinate can be explained as follows:

groupId

Defines a scope for the name of the artifact. You would typically use all or part of a package name as a group ID – for example, **org.fusesource.example**.

artifactId

Defines the artifact name (relative to the group ID).

version

Specifies the artifact's version. A version number can have up to four parts: **n.n.n.n**, where the last part of the version number can contain non-numeric characters (for example, the last part of **1.0-SNAPSHOT** is the alphanumeric substring, **0-SNAPSHOT**).

packaging

Defines the packaged entity that is produced when you build the project. For OSGi projects, the packaging is **bundle**. The default value is **jar**.

classifier

Enables you to distinguish between artifacts that were built from the same POM, but have different content.

The group ID, artifact ID, packaging, and version are defined by the corresponding elements in an artifact's POM file. For example:

```
<project ... >
...
<groupId>org.fusesource.example</groupId>
<artifactId>bundle-demo</artifactId>
<packaging>bundle</packaging>
<version>1.0-SNAPSHOT</version>
...
</project>
```

For example, to define a dependency on the preceding artifact, you could add the following **dependency** element to a POM:

-

```
<project ... >
...
<dependencies>
  <dependency>
    <groupId>org.fusesource.example</groupId>
    <artifactId>bundle-demo</artifactId>
    <version>1.0-SNAPSHOT</version>
  </dependency>
</dependencies>
...
</project>
```



NOTE

It is **not** necessary to specify the **bundle** package type in the preceding dependency, because a bundle is just a particular kind of JAR file and **jar** is the default Maven package type. If you do need to specify the packaging type explicitly in a dependency, however, you can use the **type** element.