



Red Hat Fuse 7.11

Installing on JBoss EAP

Install Red Hat Fuse 7.11 on Red Hat JBoss Enterprise Application Platform 7.4.5

Red Hat Fuse 7.11 Installing on JBoss EAP

Install Red Hat Fuse 7.11 on Red Hat JBoss Enterprise Application Platform 7.4.5

Legal Notice

Copyright © 2023 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

Installation and management procedures to enable development of Fuse applications that run on JBoss EAP.

Table of Contents

PREFACE	3
MAKING OPEN SOURCE MORE INCLUSIVE	4
CHAPTER 1. INSTALLING FUSE ON JBOSS EAP	5
CHAPTER 2. STARTING FUSE ON JBOSS EAP SERVER	7
2.1. STARTING JBOSS EAP IN STANDALONE MODE	7
2.2. STARTING JBOSS EAP IN DOMAIN MODE	7
CHAPTER 3. VERIFYING INSTALLATION OF FUSE ON JBOSS EAP	8
CHAPTER 4. ADDING FUSE USERS TO JBOSS EAP	9
CHAPTER 5. STOPPING JBOSS EAP	10
CHAPTER 6. PATCHING A FUSE ON JBOSS EAP INSTALLATION	11
6.1. HOTFIX PATCHES FOR FUSE ON JBOSS EAP	11
6.2. INSTALLING A FUSE HOTFIX PATCH ON JBOSS EAP	12
CHAPTER 7. SETTING UP MAVEN LOCALLY	14
7.1. PREPARING TO SET UP MAVEN	14
7.2. ADDING RED HAT REPOSITORIES TO MAVEN	14
7.3. USING LOCAL MAVEN REPOSITORIES	16
7.4. SETTING MAVEN MIRROR USING ENVIRONMENTAL VARIABLES OR SYSTEM PROPERTIES	16
7.4.1. About Maven mirror	16
7.4.2. Adding Maven mirror to settings.xml	17
7.4.3. Setting Maven mirror using environmental variable or system property	17
7.4.4. Using Maven options to specify Maven mirror url	17
7.5. ABOUT MAVEN ARTIFACTS AND COORDINATES	17

PREFACE

An administrator can install Red Hat Fuse on Red Hat JBoss Enterprise Application Platform so that programmers can develop Fuse applications that run on JBoss EAP. For information about JBoss EAP, see [Introduction to JBoss EAP](#).

To install Fuse on JBoss EAP, see:

- [Chapter 1, *Installing Fuse on JBoss EAP*](#)
- [Chapter 2, *Starting Fuse on JBoss EAP server*](#)
- [Chapter 3, *Verifying installation of Fuse on JBoss EAP*](#)
- [Chapter 4, *Adding Fuse users to JBoss EAP*](#)
- [Chapter 5, *Stopping JBoss EAP*](#)
- [Chapter 6, *Patching a Fuse on JBoss EAP installation*](#)
- [Chapter 7, *Setting up Maven locally*](#)

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our [CTO Chris Wright's message](#).

CHAPTER 1. INSTALLING FUSE ON JBOSS EAP

To develop Fuse applications on JBoss EAP, install JBoss EAP, and then install Fuse.

Prerequisites

- Red Hat Customer Portal login credentials.
- A supported version of a Java runtime is installed.
[Red Hat Fuse Supported Runtimes](#) lists the Java runtimes that are supported on different operating systems.

Procedure

1. Install JBoss EAP 7.4.5:
 - a. Go to the [Enterprise Application Platform Software Downloads](#) page on the Red Hat Customer Portal.
 - b. When prompted, log in to your customer account.
 - c. In the **Version** dropdown menu, select **7.4**.
 - d. Click the **Download** link for the **Red Hat JBoss Enterprise Application Platform 7.4 Installer** package.
 - e. Run the downloaded installer. In the following example, replace **DOWNLOAD_LOCATION** with the location of the JBoss EAP installer on your system:

```
java -jar DOWNLOAD_LOCATION/jboss-eap-7.4.0-installer.jar
```
 - f. Accept the terms and conditions.
 - g. Choose your preferred installation path, which is represented by **EAP_HOME** for the JBoss EAP runtime.
 - h. Create an administrative user and be sure to note this administrative user's credentials. You need them to log in to the Fuse Management Console.
 - i. Accept the default settings on the remaining screens.
2. Check the [Red Hat Fuse Supported Configurations](#) page for any notes or advice on the compatibility of JBoss EAP patches with Red Hat Fuse. If relevant, install any additional JBoss EAP patches.
3. Install Fuse 7.11 on JBoss EAP 7.4.5:
 - a. Go to the [Red Hat Fuse Software Downloads](#) page on the Red Hat Customer Portal.
 - b. When prompted, log in to your customer account.
 - c. In the **Version** dropdown menu, select **7.11.0**.
 - d. Click the **Download** link for the **Red Hat Fuse 7.11.0 on EAP Installer** package.
 - e. Open a shell prompt (or a command prompt on Windows).

- f. Change to the **EAP_HOME** directory, which is the root directory of the fresh Red Hat JBoss Enterprise Application Platform installation.
- g. Run the downloaded installer. In the following example, replace **DOWNLOAD_LOCATION** with the location of the downloaded Fuse installer on your system:

```
java -jar DOWNLOAD_LOCATION/fuse-eap-installer-7.11.1.fuse-7_11_1-00014-redhat-00002.jar
```

4. The installer runs without prompts and logs its activity to the screen.

Next steps

- Start JBoss EAP, verify that Fuse is running, and then add Fuse users to JBoss EAP. The next chapters explain how to perform these tasks.
- Optional but recommended: set up Maven for local use with Fuse projects. This is explained in [Chapter 7, Setting up Maven locally](#).

Additional resources

- For more detailed information about installing JBoss EAP, see the [JBoss EAP Installation Guide](#).
- For more detailed information about installing JBoss EAP patches, see the [JBoss EAP Patching and Upgrading Guide](#).

CHAPTER 2. STARTING FUSE ON JBOSS EAP SERVER

Fuse on JBoss EAP supports both the standalone mode and domain mode. This chapter explains how to start the server in either standalone mode or domain mode.

2.1. STARTING JBOSS EAP IN STANDALONE MODE

The commands in this section Fuse explain how to start JBoss EAP as a standalone server.

Prerequisites

- JBoss EAP 7.4.5 is installed.

Procedure

- For **Red Hat Enterprise Linux**, run the following command:
EAP_HOME/bin/standalone.sh
- For **Microsoft Windows Server**, run the following command:
EAP_HOME\bin\standalone.bat

2.2. STARTING JBOSS EAP IN DOMAIN MODE

The commands in this section explain how to start Fuse on JBoss EAP as a domain server.

Prerequisites

- JBoss EAP 7.4.5 is installed.

Procedure

- For **Red Hat Enterprise Linux**, run the following command:
EAP_HOME/bin/domain.sh
- For **Microsoft Windows Server**, run the following command:
EAP_HOME\bin\domain.bat

Additional resources

- To obtain a list of parameters that you can pass to the start-up scripts, enter the **-h** parameter.
- 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](#).

CHAPTER 3. VERIFYING INSTALLATION OF FUSE ON JBOSS EAP

After installing Fuse on JBoss EAP, verify that the installation was successful.

Prerequisites

- Fuse 7.11 on JBoss EAP 7.4.5 is installed.
- The JBoss EAP server is running.

Procedure

1. Open the following file to check if any error messages have been logged:
EAP_HOME/standalone/log/server.log
2. In a browser, open the [Fuse Management Console](#).
3. Sign in by entering the administrator credentials that you created during JBoss EAP installation.

Results

If the Fuse Management Console runs and you can log in, the Fuse installation was successful.

CHAPTER 4. ADDING FUSE USERS TO JBOSS EAP

Run the JBoss EAP **add-user** script to add Fuse users to JBoss EAP.

Prerequisites

- JBoss EAP is running.

Procedure

1. Navigate to **EAP_HOME/bin**.
2. Run the **add-user** script. For example:
./add-user.sh
3. Respond to the prompts to create a new user:
 - **Management User** is a Fuse administrative user on JBoss EAP.
 - **Application User** is a Fuse non-administrative user on JBoss EAP.

CHAPTER 5. STOPPING JBOSS EAP

The following procedure uses the Management command line interface (CLI) to stop JBoss EAP.

Prerequisites

- JBoss EAP 7.4.5 is running.

Procedure

1. Launch the Management CLI by running:

```
┆ $ 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
```

Alternative

Here is another way to stop JBoss EAP:

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

CHAPTER 6. PATCHING A FUSE ON JBOSS EAP INSTALLATION

This chapter explains how to apply a Fuse hotfix patch to an existing Fuse on JBoss EAP installation. It includes the following topics:

- [Section 6.1, “Hotfix patches for Fuse on JBoss EAP”](#)
- [Section 6.2, “Installing a Fuse hotfix patch on JBoss EAP”](#)

Upgrading JBoss EAP

You can also upgrade the underlying version of JBoss EAP to another version that is supported by Fuse without needing to reinstall and redeploy Fuse on JBoss EAP. For more details, see the [JBoss EAP Patching and Upgrading Guide](#).



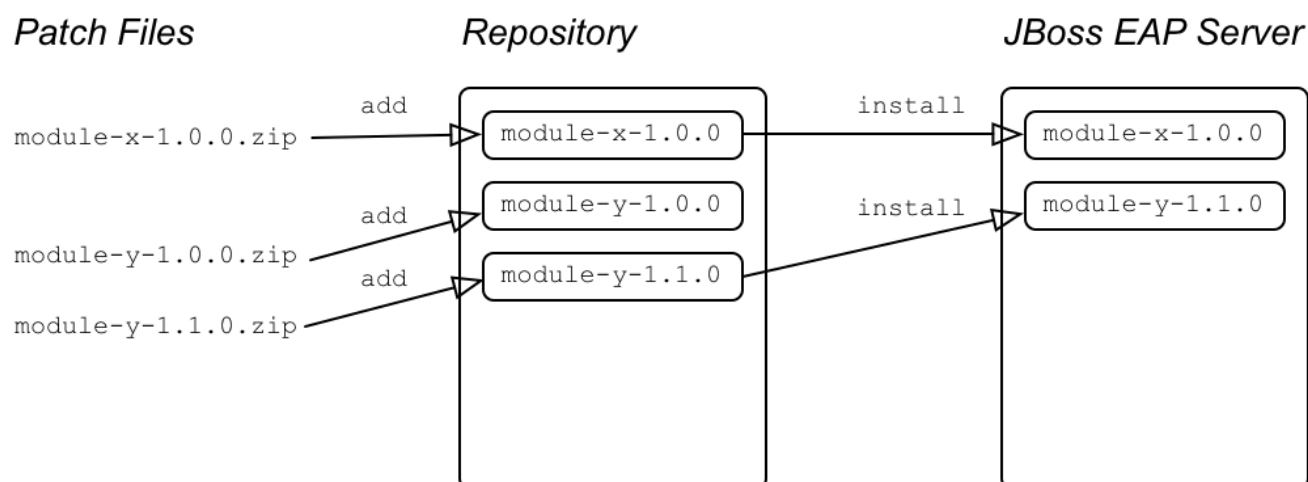
IMPORTANT

You can only upgrade JBoss EAP to a version that is documented as supported on the [Fuse Supported Configurations page](#).

6.1. HOTFIX PATCHES FOR FUSE ON JBOSS EAP

Fuse hotfix patches contain updated versions of specific files in a Fuse on JBoss EAP installation. They typically include only fixes for one or more critical bugs. Hotfix patches are applied on top of your existing Red Hat Fuse distribution and update a subset of the existing Fuse files only.

Applying patches for Fuse on JBoss EAP is a two-stage process where patch files are first added to a patch repository and then installed in the JBoss EAP server. The following diagram shows an overview of the Fuse patching process on JBoss EAP:



Patch repository

The patch repository is a holding area for Fuse on JBoss EAP patches that runs in the same JVM as the JBoss EAP server. When a patch is present in the repository, this does not imply that it has been installed in the JBoss EAP server. You must first add the patch to the repository, and then you can install the patch from the repository into the JBoss EAP server.

fusepatch utility

The **fusepatch** utility is a command-line tool for patching Fuse on JBoss EAP. After installing the Fuse on EAP package, the **fusepatch.sh** script (Linux and UNIX) and the **fusepatch.bat** (Windows) script are available in the **bin** directory of the JBoss EAP server.

6.2. INSTALLING A FUSE HOTFIX PATCH ON JBOSS EAP

A Fuse hotfix patch must be installed on top of an existing Fuse installation. This section explains how to install a hotfix patch, **fuse-eap-distro-VERSION.fuse-MODULE_ID.HOTFIX_ID.zip**, on top of an existing Fuse installation that already includes **fuse-eap-distro-VERSION.fuse-MODULE_ID-redhat-BASE_ID**.

Prerequisites

- [Section 6.1, “Hotfix patches for Fuse on JBoss EAP”](#).
- Download the hotfix patch **.zip** file available on demand from Red Hat Support.
- Read the instructions in the **readme.txt** file accompanying the hotfix patch file, in case there are any extra steps that you must perform to install it.
- Make a full backup of your Fuse on JBoss EAP installation before applying the patch.

Procedure

1. Copy the hotfix patch file to your **EAP_HOME** directory.
2. Make sure that the correct base version has already been added to your patch repository and installed on the JBoss EAP server.

For example, given a base module **fuse-eap-distro-7.11.1.fuse-sb2-7_11_1-00022-redhat-00002**, to check the **MODULE_ID** and **BASE_ID** that are installed in the repository, enter the following command:

```
bin/fusepatch.sh --query-repository
```

The following response should be returned:

```
fuse-eap-distro-7.11.1.fuse-sb2-7_11_1-00022-redhat-00002
```

And to check that the same IDs are installed on the JBoss EAP server, enter the following command:

```
bin/fusepatch.sh --query-server
```

The following response should be returned:

```
fuse-eap-distro-7.11.1.fuse-sb2-7_11_1-00022-redhat-00002
```

3. Given the one-off hotfix patch file, **fuse-eap-distro-7.7.0.fuse-770013.hf1.zip**, add this to your repository and associate it with the base installation by entering the following command:

```
bin/fusepatch.sh --add file:fuse-eap-distro-7.7.0.fuse-770013.hf1.zip --one-off fuse-eap-distro-7.11.1.fuse-sb2-7_11_1-00022-redhat-00002
```


4. Given the base module, **fuse-eap-distro-7.11.1.fuse-sb2-7_11_1-00022-redhat-00002**, update the JBoss EAP server to the latest version:

```
bin/fusepatch.sh --update fuse-eap-distro-7.7.0.fuse-770013.hf1
```

5. Perform any post-installation steps documented in the patch instructions.

Additional resources

- For more details on the **fusepatch** command, enter:

```
bin/fusepatch.sh --help
```

CHAPTER 7. SETTING UP MAVEN LOCALLY

Typical Fuse application development uses Maven to build and manage projects.

The following topics describe how to set up Maven locally:

- [Section 7.1, “Preparing to set up Maven”](#)
- [Section 7.2, “Adding Red Hat repositories to Maven”](#)
- [Section 7.3, “Using local Maven repositories”](#)
- [Section 7.4, “Setting Maven mirror using environmental variables or system properties”](#)
- [Section 7.5, “About Maven artifacts and coordinates”](#)

7.1. PREPARING TO SET UP MAVEN

Maven is a free, open source, build tool from Apache. Typically, you use Maven to build Fuse applications.

Procedure

1. Download the latest version of Maven from the [Maven download page](#).
2. Ensure that your system is connected to the Internet.
While building a project, the default behavior is that Maven searches external repositories and downloads the required artifacts. Maven looks for repositories that are accessible over the Internet.

You can change this behavior so that Maven searches only repositories that are on a local network. That is, Maven can run in an offline mode. In offline mode, Maven looks for artifacts in its local repository. See [Section 7.3, “Using local Maven repositories”](#).

7.2. ADDING RED HAT REPOSITORIES TO MAVEN

To access artifacts that are in Red Hat Maven repositories, you need to add those repositories to Maven’s **settings.xml** file. Maven looks for the **settings.xml** file in the **.m2** directory of the user’s home directory. If there is not a user specified **settings.xml** file, Maven uses the system-level **settings.xml** file at **M2_HOME/conf/settings.xml**.

Prerequisite

You know the location of the **settings.xml** file in which you want to add the Red Hat repositories.

Procedure

In the **settings.xml** file, add **repository** elements for the Red Hat repositories as shown in this example:

```
<?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>

```

7.3. USING LOCAL MAVEN REPOSITORIES

If you are running a container without an Internet connection, and you need to deploy an application that has dependencies that are not available offline, you can use the Maven dependency plug-in to download the application's dependencies into a Maven offline repository. You can then distribute this customized Maven offline repository to machines that do not have an Internet connection.

Procedure

1. In the project directory that contains the **pom.xml** file, download a repository for a Maven project by running a command such as the following:

```

mvn org.apache.maven.plugins:maven-dependency-plugin:3.1.0:go-offline -
Dmaven.repo.local=/tmp/my-project

```

In this example, Maven dependencies and plug-ins that are required to build the project are downloaded to the **/tmp/my-project** directory.

2. Distribute this customized Maven offline repository internally to any machines that do not have an Internet connection.

7.4. SETTING MAVEN MIRROR USING ENVIRONMENTAL VARIABLES OR SYSTEM PROPERTIES

When running the applications you need access to the artifacts that are in the Red Hat Maven repositories. These repositories are added to Maven's **settings.xml** file. Maven checks the following locations for **settings.xml** file:

- looks for the specified url
- if not found looks for **\${user.home}/.m2/settings.xml**
- if not found looks for **\${maven.home}/conf/settings.xml**
- if not found looks for **\${M2_HOME}/conf/settings.xml**
- if no location is found, empty **org.apache.maven.settings.Settings** instance is created.

7.4.1. About Maven mirror

Maven uses a set of remote repositories to access the artifacts, which are currently not available in local repository. The list of repositories almost always contains Maven Central repository, but for Red Hat Fuse, it also contains Maven Red Hat repositories. In some cases where it is not possible or allowed to

access different remote repositories, you can use a mechanism of Maven mirrors. A mirror replaces a particular repository URL with a different one, so all HTTP traffic when remote artifacts are being searched for can be directed to a single URL.

7.4.2. Adding Maven mirror to `settings.xml`

To set the Maven mirror, add the following section to Maven's `settings.xml`:

```
<mirror>
  <id>all</id>
  <mirrorOf>*</mirrorOf>
  <url>http://host:port/path</url>
</mirror>
```

No mirror is used if the above section is not found in the `settings.xml` file. To specify a global mirror without providing the XML configuration, you can use either system property or environmental variables.

7.4.3. Setting Maven mirror using environmental variable or system property

To set the Maven mirror using either environmental variable or system property, you can add:

- Environmental variable called `MAVEN_MIRROR_URL` to `bin/setenv` file
- System property called `mavenMirrorUrl` to `etc/system.properties` file

7.4.4. Using Maven options to specify Maven mirror url

To use an alternate Maven mirror url, other than the one specified by environmental variables or system property, use the following maven options when running the application:

- `-DmavenMirrorUrl=mirrorId::mirrorUrl`
for example, `-DmavenMirrorUrl=my-mirror::http://mirror.net/repository`
- `-DmavenMirrorUrl=mirrorUrl`
for example, `-DmavenMirrorUrl=http://mirror.net/repository`. In this example, the `<id>` of the `<mirror>` is just a mirror.

7.5. ABOUT MAVEN ARTIFACTS AND COORDINATES

In the Maven build system, the basic building block is an *artifact*. After a build, the output of an artifact is typically an archive, such as a JAR or WAR file.

A key aspect of Maven is the ability to locate artifacts and manage the dependencies between them. A *Maven coordinate* is a set of values that identifies the location of a particular artifact. A basic coordinate has three values in the following form:

groupId:artifactId:version

Sometimes Maven augments a basic coordinate with a *packaging* value or with both a *packaging* value and a *classifier* value. A Maven coordinate can have any one of the following forms:

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

Here are descriptions of the values:

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

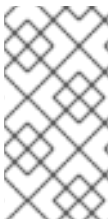
Enables you to distinguish between artifacts that were built from the same POM, but have different content.

Elements in an artifact's POM file define the artifact's group ID, artifact ID, packaging, and version, as shown here:

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

To define a dependency on the preceding artifact, you would add the following **dependency** element to a POM file:

```
<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.

