



Red Hat JBoss Web Framework Kit 2.7 Install Red Hat JBoss Web Framework Kit

Information for users installing JBoss Web Framework Kit

Red Hat Customer Content Services

Red Hat JBoss Web Framework Kit 2.7 Install Red Hat JBoss Web Framework Kit

Information for users installing JBoss Web Framework Kit

Red Hat Customer Content Services

Legal Notice

Copyright © 2015 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Abstract

This document details how to install and set up JBoss Web Framework Kit.

Table of Contents

| | |
|--|-----------|
| Chapter 1. Introduction to Red Hat JBoss Web Framework Kit | 2 |
| 1.1. About Red Hat JBoss Web Framework Kit | 2 |
| 1.2. About the JBoss Web Framework Kit Tiers | 2 |
| 1.3. About the JBoss Web Framework Kit Distribution | 3 |
| Chapter 2. System Requirements | 4 |
| 2.1. Platforms | 4 |
| 2.2. Supported Operating Systems and Java Developer Kits | 4 |
| 2.3. Apache Maven | 4 |
| Chapter 3. Use the JBoss Web Framework Kit Maven Repository | 5 |
| 3.1. About the JBoss Web Framework Kit Maven Repository | 5 |
| 3.2. Download and make Available the Maven Repository | 6 |
| 3.3. Configure Maven to Use the Repository | 7 |
| Chapter 4. Use the JBoss Stacks | 9 |
| 4.1. Available JBoss Stacks | 9 |
| 4.2. Add a JBoss Stack to a Project | 10 |
| Chapter 5. Use the Archetypes | 11 |
| 5.1. Available Archetypes | 11 |
| 5.2. Use the Archetypes with the Command Line | 11 |
| 5.3. Use the Archetypes with JBoss Developer Studio | 12 |
| Chapter 6. Use the Examples | 14 |
| 6.1. Available Examples | 14 |
| 6.2. Use the Examples with the Command Line | 14 |
| 6.3. Use the Examples with JBoss Developer Studio | 15 |
| Chapter 7. Upgrade from Earlier Versions of JBoss Web Framework Kit | 17 |
| 7.1. Upgrade from Earlier Versions of JBoss Web Framework Kit | 17 |
| 7.2. Upgrade JBoss Stacks from JBoss Web Framework Kit 2.3.0 and Earlier | 17 |
| Revision History | 19 |

Chapter 1. Introduction to Red Hat JBoss Web Framework Kit

1.1. About Red Hat JBoss Web Framework Kit

Red Hat JBoss Web Framework Kit is a set of enterprise-ready versions of popular open source frameworks. Together, these frameworks provide a solution for developing light and rich Java-based web applications.

JBoss Web Framework Kit comprises enterprise distributions of JBoss community web application frameworks and tested third-party frameworks. These leading frameworks support fast and easy client-side and server-side application development and testing, with frameworks including RichFaces, jQuery, jQuery Mobile, Hibernate Search, Spring, and Arquillian.

The breadth of JBoss Web Framework Kit frameworks provides choice for Java application development. Each framework has been tested and certified for use in applications deployed to Red Hat JBoss Enterprise Application Platform, Red Hat JBoss Web Server, or Red Hat OpenShift JBoss EAP cartridge. Inclusion in JBoss Web Framework Kit ensures stable versions of frameworks are available over long-term enterprise product life cycles, with regular releases for fixes and nonintrusive feature updates. Further, Red Hat JBoss Developer Studio (an Eclipse-based development environment) provides integrated project templates, quickstarts and tooling for many of the JBoss Web Framework Kit frameworks.

For the complete list of the frameworks composing JBoss Web Framework Kit and the certified platform and framework configurations, see <https://access.redhat.com/site/articles/112543> on the Red Hat Customer Portal.

See Also:

- » [Section 1.2, “About the JBoss Web Framework Kit Tiers”](#)

[Report a bug](#)

1.2. About the JBoss Web Framework Kit Tiers

The frameworks composing JBoss Web Framework Kit are categorized in four distinct tiers. A description of each tier and the associated Red Hat support is detailed here.

Tier 1 - Included Components

These are components that are based wholly or partly on open source technologies that support broad collaboration and where Red Hat maintains a leadership role; as such Red Hat is able to support these components and provide upgrades and fixes under our standard support terms and conditions.

Tier 2 - Tested Frameworks

These are third-party frameworks where Red Hat does not have sufficient influence and does not provide upgrades and fixes under our standard support terms and conditions. Commercially reasonable support is provided by Red Hat Global Support Services for these frameworks.

Tier 3 - Frameworks in Tested Examples

These are third-party frameworks where Red Hat does not have sufficient influence and does not provide upgrades and fixes under our standard support terms and conditions. Red Hat supports the examples these frameworks are used in and the generic use cases that these examples intend to demonstrate.

Tier 4 - Confirmed Frameworks

These are third-party frameworks that do not receive any support from Red Hat, but Red Hat verifies that the frameworks run successfully on Red Hat JBoss Enterprise Application Platform. Frameworks and versions not listed here have not been explicitly tested and certified, and thus may be subject to support limitations.

For a list of JBoss Web Framework Kit frameworks by tier, see <https://access.redhat.com/site/articles/112543> on the Red Hat Customer Portal.

[Report a bug](#)

1.3. About the JBoss Web Framework Kit Distribution

The frameworks composing JBoss Web Framework Kit are distributed from a range of sources and in a variety of formats:

- The component frameworks are available from the Red Hat Customer Portal. They are distributed in two alternative formats: a binary for each framework or together as one Maven repository. In addition, the source code for each framework is provided for inspection.
- The third party frameworks are not distributed by Red Hat and each must be obtained from its own source.

A number of defined Maven JBoss stacks are provided as part of the JBoss Web Framework Kit distribution. All of the BOMs defining the JBoss stacks are available in the Maven repository .zip file available to download from the Red Hat Customer Portal or from <http://www.jboss.org/developer-materials/> on the JBoss Developer Framework website.

An extensive set of examples are also provided as part of the JBoss Web Framework Kit distribution:

- TicketMonster is a moderately complex application demonstrating a number of the JBoss Web Framework Kit frameworks working together.
- Quickstarts and Maven archetypes illustrate subsets of the JBoss Web Framework Kit frameworks used to create simple applications.
- RichFaces, Snowdrop and Seam demonstrations showcase the power of each framework in web application development.

All of these examples are available from the Red Hat Customer Portal, with TicketMonster, the quickstarts, and the Maven archetypes also available from <http://www.jboss.org/developer-materials/> on the JBoss Developer Framework website.

[Report a bug](#)

Chapter 2. System Requirements

2.1. Platforms

JBoss Web Framework Kit has been tested for use with Red Hat JBoss Enterprise Application Platform, Red Hat JBoss Web Server, and Red Hat OpenShift JBoss EAP cartridge. Note that specific versions of the platforms are supported by this release of JBoss Web Framework Kit and that not all frameworks are certified for use with all platforms. For the complete list of certified platform and framework versions and configurations, see <https://access.redhat.com/site/articles/112543> on the Red Hat Customer Portal.

[Report a bug](#)

2.2. Supported Operating Systems and Java Developer Kits

JBoss Web Framework Kit can be used with the operating systems and Java developer kit combinations certified for use with the platform versions supported by this release of JBoss Web Framework Kit.

For more information on the JBoss EAP supported configurations, see <https://access.redhat.com/articles/111663> on the Red Hat Customer Portal.

For more information on the JBoss Web Server supported configurations, see <https://access.redhat.com/articles/258223> on the Red Hat Customer Portal.

[Report a bug](#)

2.3. Apache Maven

Apache Maven 3.0.4 or later must be available on your system to use the JBoss stacks, archetypes and examples distributed as part of this JBoss Web Framework Kit release.

If you will use Maven from the command line to build and deploy your applications to JBoss EAP, you must download and install Maven. Alternatively, if using JBoss Developer Studio to build and deploy your applications, Maven is distributed with JBoss Developer Studio and it is not necessary to install an independent version.

To obtain a copy of Maven, go to <http://maven.apache.org/> on the Apache Maven website and download the latest distribution for your operating system. For information on installing and configuring Maven, see the Maven documentation.

[Report a bug](#)

Chapter 3. Use the JBoss Web Framework Kit Maven Repository

A Maven repository is provided as part the JBoss Web Framework Kit release. The purpose of the repository is to provide API sources and dependencies for the component frameworks and standardize application development by listing specific source versions to use for all JBoss Web Framework Kit frameworks. The repository is essential for using the JBoss stacks, archetypes and examples distributed with the JBoss Web Framework Kit release.

This section gives more information about the repository and guides you through the installation and configuration necessary to use the JBoss Web Framework Kit repository, together with information about how to use the repository in your projects. Instructions are also provided for achieving these tasks using JBoss Developer Studio when appropriate.

[Report a bug](#)

3.1. About the JBoss Web Framework Kit Maven Repository

The JBoss Web Framework Kit Maven repository is a set resources for use in Maven-based applications that use the JBoss Web Framework Kit frameworks and deploy to JBoss EAP. It assists to simplify and standardize application development by providing a set of centralized sources and specifying tested and compatible source versions.



Important

The JBoss Web Framework Kit 2.7 Maven repository extends the JBoss EAP 6.3.2 Maven repository. In order to use the JBoss Web Framework Kit 2.7 Maven repository, the complete JBoss EAP 6.3.2 Maven repository must be available during application development and Maven configured to use it. Note that the complete JBoss EAP 6.3.2 Maven repository is achieved with the JBoss EAP 6.3.0 Maven repository and JBoss EAP Maven 6.3 CP02 applied.

For information on applying Maven repository patches, see the JBoss EAP 6.3 Development Guide available on the Red Hat Customer Portal. For information on obtaining and configuring the JBoss EAP Maven repositories for use, see the JBoss EAP Development Guide available on the Red Hat Customer Portal.

The JBoss Web Framework Kit Maven repository consists of the following:

- ✦ a complete set of the tier 1 framework APIs and their dependent artifacts
- ✦ several Bill of Materials (BOM) for JBoss stacks

The component framework artifacts can be shared across multiple projects and used by multiple developers, reducing the memory footprint by only being stored once and providing consistency in the build process. Each BOM specifies the versions of a stack (or collection) of artifacts. When used with the JBoss Web Framework Kit archetypes and examples and your Maven-based projects, the BOMs ensure that the correct versions of artifacts are always used.

The JBoss Web Framework Kit Maven repository is provided as an offline repository, available for download from the Red Hat Customer Portal. In this format, the repository can be easily integrated into an enterprise Maven repository or used as a local file system repository, eliminating the need to be connected to the Internet when building projects.



Note

The JBoss Web Framework Kit Maven repository is also provided as part of the Red Hat online Maven repository. This online repository is Technology Preview only; it is not fully supported and not recommended for business-critical use. Red Hat reserves the right to modify the behavior and content of this repository or remove it completely with little or no notice. For more information about using the Red Hat online Maven repository, see <https://access.redhat.com/site/maven-repository> on the Red Hat Customer Portal.

[Report a bug](#)

3.2. Download and make Available the Maven Repository

The JBoss Web Framework Kit Maven repository is available to download from the Red Hat Customer Portal as a **.zip** file. Once downloaded, you must extract the contents of the file and make it available for use by your preferred Maven management system: local file system, web server, Nexus Maven repository manager. The latter two systems host Maven repositories so making them accessible to multiple users and may require additional configuration as noted in the procedure below.

Procedure 3.1. Download and make Available the Maven Repository

1. Log into the Red Hat Customer Portal at <http://access.redhat.com>.
2. Locate the downloads for this release version of JBoss Web Framework Kit.
3. Download the **.zip** file for the Red Hat JBoss Web Framework Kit Maven Repository.
4. Complete the appropriate steps for your Maven management system:
 - A. For a local file system installation, extract the contents of the downloaded **.zip** file to the local file system location where you want to keep them.
 - B. For an Apache Web Server:
 - a. Extract the contents of the downloaded **.zip** file to a web-accessible directory in the Apache Web Server, for example, `/var/www/maven-repositories/`.
 - b. In `/etc/httpd/conf.d/`, create a new text file named **maven-repositories.conf**. This file will be used to enable read access and directory browsing of `/var/www/maven-repositories/`.
 - c. In the **maven-repositories.conf** file, add the following:

```
<Directory "/var/www/maven-repositories">
  Options Indexes FollowSymLinks
  AllowOverride None
  Require all granted
</Directory>

Alias /maven-repositories /var/www/maven-repositories
```

This grants access to the contents of `/var/www/maven-repositories/` and provides an alias for the Maven repository path.

- d. Save and close the file.
- C. For a Nexus Maven Repository Manager:
- a. Extract the contents of the downloaded **.zip** file to a suitable location in the Nexus file system.
 - b. Start the Nexus Repository Manager
 - c. Add the JBoss Web Framework Kit Maven repository as a hosted repository, pointing to the extracted **.zip** file contents.
 - d. Add the JBoss Web Framework Kit Maven hosted repository to the desired repository group to make it available for use.

[Report a bug](#)

3.3. Configure Maven to Use the Repository

For Maven to use the JBoss Web Framework Kit Maven repository, you must configure the Maven configuration **settings.xml** file. There are two options available for configuring this file: manually using a text editor or automatically using JBoss Developer Studio. Both methods are detailed in the following sections.

A sample Maven configuration file, **example-settings.xml**, is provided in the JBoss Web Framework Kit Maven repository **.zip** file demonstrating the JBoss EAP and JBoss Web Framework Kit Maven repository declarations. The repository URLs must be specified in the configuration file **<url>** fields and vary depending on the type of Maven repository management system in use. The table below gives example URLs for the JBoss Web Framework Kit Maven repository for various Maven repository management systems.

Table 3.1. Example URLs for Maven Repository Management Systems

| Maven Repository Management System | Example URL |
|------------------------------------|---|
| Local file system | file:///home/user/projects/maven-repositories/jboss-wfk-maven-repository/ |
| Apache Web Server | http://intra.acme.com/maven-repositories/jboss-wfk-maven-repository/ |
| Nexus Repository Manager | https://intra.acme.com/nexus/content/repositories/jboss-wfk-maven-repository/ |

[Report a bug](#)

3.3.1. Configure Maven Manually

You can opt to configure your system to use the JBoss Web Framework Kit Maven repository by editing the Maven configuration **settings.xml** file manually using a text editor, as detailed in the procedure below. Depending on your Maven setup, the **settings.xml** file may be unique to each user or shared by multiple users.

Procedure 3.2. Configure the Maven **settings.xml** File Manually

1. Open the Maven **settings.xml** file in a text editor.
2. Open the provided **/path/to/jboss-wfk-maven-repository/example-settings.xml** for viewing.

3. Copy the JBoss Web Framework Kit `<profile>` and `<activeProfile>` sections from the `example-settings.xml` file to the `settings.xml` file.
4. Replace the placeholder `<url>` text with the path to the JBoss Web Framework Kit Maven repository installation.
5. Save and close the `settings.xml` file.

[Report a bug](#)

3.3.2. Configure Maven in JBoss Developer Studio

JBoss Developer Studio provides actions for automatically configuring your Maven `setting.xml` file for the JBoss Web Framework Kit Maven repository, as detailed in the procedure below.



Note

By default, JBoss Developer Studio expects the Maven configuration `settings.xml` file to be in a user's `.m2` directory. If you are using a `settings.xml` file located elsewhere, change the IDE Maven settings in **Preferences** under **Maven** → **User Settings** before starting this procedure.

Procedure 3.3. Configure Maven in JBoss Developer Studio

1. In JBoss Developer Studio, click **Window** → **Preferences**, expand **JBoss Tools** and select **JBoss Maven Integration**.
2. Click **Configure Maven Repositories**.
3. Click **Add Repository**.
4. Click **Recognize JBoss Maven Enterprise Repositories**.
5. Navigate to `path/to/jboss-wfk-maven-repository` and click **OK**. JBoss Maven Tools recursively scans the path searching for the Maven repository. Once found, the fields in the **Add Maven Repository** wizard are automatically complete.
6. Modify the information in the **ID** and **Name** fields as desired, ensure the **Active by default** check box is selected and click **OK**.
7. Click **Finish** and at the prompt asking if you are sure you want to update the Maven configuration file click **Yes**. If the specified configuration file does not exist, JBoss Maven Tools creates it.
8. Click **Apply** and click **OK** to close the **Preferences** window.

[Report a bug](#)

Chapter 4. Use the JBoss Stacks

A number of defined Maven JBoss stacks are provided as part of the JBoss Web Framework release. The purpose of these stacks is to provide tested and certified collections of version-specific dependencies required by individual frameworks included with this release of JBoss Web Framework Kit. Each stack is defined by a Bill of Materials (BOM) in a `.pom` file. Specifying these BOMs in your project `pom.xml` files enables use of the appropriate stacks and removes the need to explicitly specify the versions of stack artifacts required by your project.

This section details the JBoss stacks available and provides instructions for using them in your projects.

[Report a bug](#)

4.1. Available JBoss Stacks

All of the BOMs defining the JBoss stacks associated with this release of JBoss Web Framework Kit are available in the Maven repository `.zip` file available to download from the Red Hat Customer Portal and individually from the JBoss Developer website.

The JBoss stacks are provided as part of two groups: `com.redhat.jboss.wfk.boms` and `org.jboss.bom.wfk`. These stacks specify versions of all relevant dependencies required when using JBoss Web Framework Kit frameworks with the supported platforms. The stacks in the `org.jboss.bom.wfk` group additionally define versions of all dependencies on Java EE 6 Specification APIs. The tables here list the stacks in each group and the core frameworks of each stack.

Table 4.1. JBoss Stacks in `com.redhat.jboss.wfk.boms` group

| JBoss Stack | Frameworks |
|--|-------------------------|
| spring-2.5-bom [a] | Spring 2.5 and Snowdrop |
| spring-3.0-bom [b] | Spring 3.0 and Snowdrop |
| spring-3.1-bom [c] | Spring 3.1 and Snowdrop |
| spring-3.2-bom [d] | Spring 3.2 and Snowdrop |
| <p>[a] Spring 2.5 is deprecated as of JBoss Web Framework Kit 2.4.0 but continues to be tested and supported in JBoss Web Framework Kit 2.</p> <p>[b] Spring 3.0 is deprecated as of JBoss Web Framework Kit 2.5.0 but continues to be tested and supported in JBoss Web Framework Kit 2.</p> <p>[c] Spring 3.1 is deprecated as of JBoss Web Framework Kit 2.5.0 but continues to be tested and supported in JBoss Web Framework Kit 2.</p> <p>[d] Use the replacement <code>jboss-javaee-6.0-with-spring</code> stack introduced in JBoss Web Framework Kit 2.4.0 instead of this stack.</p> | |

Table 4.2. JBoss Stacks in `org.jboss.bom.wfk` group

| JBoss Stack | Frameworks |
|----------------------------------|-------------------------|
| jboss-javaee-6.0-with-deltaspike | Deltaspike |
| jboss-javaee-6.0-with-hsearch | Hibernate Search |
| jboss-javaee-6.0-with-richfaces | RichFaces |
| jboss-javaee-6.0-with-spring | Spring 3.2 and Snowdrop |
| jboss-javaee-6.0-with-spring4 | Spring 4 and Snowdrop |
| jboss-javaee-6.0-with-spring4.1 | Spring 4.1 and Snowdrop |

| JBoss Stack | Frameworks |
|-----------------------------|--|
| jboss-javaee-6.0-with-tools | Arquillian, Shrinkwrap, Selenium and other tools |

[Report a bug](#)

4.2. Add a JBoss Stack to a Project

You must include information about the BOM for each JBoss stack in your project `pom.xml` file, as detailed in the procedure below, to take advantage of the certified stacks.

Procedure 4.1. Add a JBoss Stack to a Project

1. Open your project `pom.xml` file for editing.
2. Locate the `<dependencyManagement>` section.
3. Within the `<dependencies>` tags, add the following:

```
<dependency>
  <groupId>stack_group</groupId>
  <artifactId>stack_name</artifactId>
  <type>pom</type>
  <version>stack_version</version>
  <scope>import</scope>
</dependency>
```

4. Replace `stack_group`, `stack_name` and `stack_version` with the appropriate JBoss stacks group and the individual JBoss stack name and version. For example, replace with `com.redhat.jboss.wfk.boms`, `spring-3.2-bom` and `2.7.0-redhat-1` to use the JBoss Web Framework Kit 2.6 JBoss stack for Spring 3.2 and Snowdrop.
5. Within the `<dependencies>` tags that follow the `<dependencyManagement>` section, add the following for each dependency required by your project:

```
<dependency>
  <groupId>dependency_group</groupId>
  <artifactId>dependency_name</artifactId>
</dependency>
```

6. Replace `dependency_group` and `dependency_name` with the appropriate dependency group and name. For example, `org.springframework` and `spring-core` to use the Spring core dependency. The version for the dependency is taken from the included JBoss stack.

[Report a bug](#)

Chapter 5. Use the Archetypes

A number of Maven archetypes are provided as part of the JBoss Web Framework release. The purpose of these archetypes is to provide project patterns that you can customize and use as the starting point for developing your own applications. The archetypes use dependencies from JBoss Web Framework Kit and the generated applications are designed to run with the platforms certified for this release of JBoss Web Framework Kit.

This section details the archetypes available and provides instructions for generating, building and deploying the resulting archetype applications. Instructions are also provided for achieving these tasks using JBoss Developer Studio.

[Report a bug](#)

5.1. Available Archetypes

All of the archetypes associated with this release of JBoss Web Framework Kit are hosted in the Maven Central repository at <http://central.sonatype.org/>. The available archetypes and generated applications are detailed here.

Table 5.1. Archetypes in org . jboss . archetype . wfk group

| Artifact Id | Generated Application |
|------------------------------------|---|
| jboss-html5-mobile-archetype | This archetype generates a Java EE 6 application that utilizes HTML5 and JAX-RS to support both desktop and mobile web browsers. |
| jboss-html5-mobile-blank-archetype | This archetype generates a blank Java EE 6 application that utilizes HTML5 and JAX-RS to support both desktop and mobile web browsers. This archetype provides only the required dependencies to get started. |
| jboss-spring-mvc-archetype | This archetype generates a starter Spring MVC application with Java EE persistence settings including server bootstrapped JPA and JTA transaction management. |
| richfaces-archetype-kitchensink | This archetype generates a kitchensink application that features annotation-driven CDI and Java Bean Validation on both the client side and the server side, plus JPA and JAX-RS on the server side only. |
| richfaces-archetype-simpleapp | This archetype generates a simple RichFaces application. |

[Report a bug](#)

5.2. Use the Archetypes with the Command Line

You must first generate the project from the archetype using Maven, then you can build the application with Maven and deploy it to a platform. The projects generated by the archetypes are all web applications so you can view the running application in a web browser. The procedures detail the steps for generating, deploying to JBoss EAP and viewing an application.

The procedures assume that JBoss EAP is already installed and Maven is configured to use the JBoss EAP and JBoss Web Framework Kit Maven repositories.

Procedure 5.1. Create a Project from an Archetype Using the Command Line

1. On the command line, navigate to the directory where you want to create the project.
2. On the command line enter

```
$ mvn archetype:generate \  
-DarchetypeGroupId=org.jboss.archetype.wfk \  
-DarchetypeArtifactId=archetype \  
-DarchetypeVersion=version.Final
```

where *archetype* denotes the unique Artifact Id and *version* denotes the complete three-digit version number for this JBoss Web Framework Kit release (for example, **2.7.0.Final** for JBoss Web Framework Kit 2.7.0). This starts the project generation process, downloading the necessary dependencies, and may take some time to complete.

3. At the prompts, accept the default values or enter your own values for groupId, artifactId, version and package of the new project. Once generated, the command line output displays

```
[INFO] BUILD SUCCESS
```

together with summary information about the build process.

Procedure 5.2. Compile and Deploy an Application using the Command Line

1. Compile the Maven-based example to verify it builds successfully

```
$ mvn compile
```

2. Start JBoss EAP

```
For Linux and OS X $ EAP_HOME/bin/standalone.sh  
For Windows      $ EAP_HOME\bin\standalone.bat
```

3. Build and deploy the application to JBoss EAP

```
$ mvn clean install jboss-as:deploy
```

4. View the deployed application in a web browser at http://localhost:8080/application_name/, where *application_name* is replaced by the name of your application.

[Report a bug](#)

5.3. Use the Archetypes with JBoss Developer Studio

You can use the archetypes to generate projects within JBoss Developer Studio using the IDE New Maven Project wizard. This is detailed in the procedure below, together with information on compiling and deploying applications to JBoss EAP using the workflow shortcuts provided by JBoss Developer Studio.

The procedures assume that JBoss EAP is already installed and Maven is configured to use the JBoss EAP and JBoss Web Framework Kit Maven repositories. They further assume that JBoss Developer Studio is installed and a server adapter has been generated for the JBoss EAP instance to enable efficient IDE-server communication. For information on configuring JBoss Developer Studio for use with JBoss EAP and JBoss Web Framework Kit, see the JBoss Web Framework Kit Start Developing guide.



Note

You may also be able to create the archetypes using the project wizards listed in JBoss Central under **Start from scratch** but you must check the archetype version being used before continuing. The archetype version is predetermined in the project wizards and it may not match the version of this JBoss Web Framework Kit release.

Procedure 5.3. Create a Project from an Archetype Using a Maven Project Wizard

1. Click **File** → **New** → **Maven Project**.
2. Ensure the **Create a simple project** check box is not selected and click **Next**.
3. Click **Add Archetype**.
4. Complete the fields about the specific JBoss Web Framework Kit archetype as follows:
 - ✦ In the **Archetype Group Id** field, type `org.jboss.archetype.wfk`.
 - ✦ In the **Archetype Artifact Id** field, type the unique Artifact Id.
 - ✦ In the **Archetype Version** field, type `version.Final` where *version* denotes the complete three-digit version number for this JBoss Web Framework Kit release. For example, `2.7.0.Final` for JBoss Web Framework Kit 2.7.0.
 - ✦ In the **Repository URL** field, type `http://central.sonatype.org/`.
5. Click **OK**.
6. From the archetype list, ensure the added archetype is selected and click **Next**.
7. Complete the fields about the project to be created as follows:
 - ✦ In the **Group Id** field, type a package for the project.
 - ✦ In the **Artifact Id** field, type a name for the project.
8. Click **Finish**. Once the project is created, it is listed in the **Project Explorer** view.

Procedure 5.4. Compile and Deploy an Application Using JBoss Server Tools

1. In the **Project Explorer** view, right-click the project name and click **Run as** → **Run on Server**.
2. Under **How do you want to select the server?**, ensure **Choose an existing server** is selected.
3. In the **Server** table, expand **localhost**, select **jboss-eap-version** where *version* denotes the JBoss EAP version, and click **Next**.
4. Ensure that your project is listed in the **Configured** column and click **Finish**. The **Console** view automatically becomes the view in focus and displays the output from the JBoss EAP server. Once deploying is complete, the web application opens in the IDE default web browser.

[Report a bug](#)

Chapter 6. Use the Examples

An extensive set of examples are provided as part of the JBoss Web Framework Kit distribution. The purpose of these examples is to demonstrate the frameworks at work individually and in conjunction with each other in sample applications. Each example is well documented with source code comments explaining the reasoning for technology and programming choices. The majority of the examples are Maven-based, illustrating usage of this widely used project build system, and they designed to run with the platforms supported by this release of JBoss Web Framework Kit.

This section details the examples available and provides instructions for building and deploying the applications. Instructions are also provided for achieving these tasks using JBoss Developer Studio.

[Report a bug](#)

6.1. Available Examples

All of the examples distributed with this release of JBoss Web Framework Kit can be obtained from the Red Hat Customer Portal. The source for the examples is provided in a number of different **.zip** files, as detailed here.

demo.zip

Contains one or more demo applications, with a demo.zip for several of the Tier 1 frameworks. Each demo illustrates multiple features from a single framework in use in a medium sized application.

quickstarts.zip

Contains many small applications demonstrating one or more frameworks in use together. Some of the quickstarts are part of collections, for example those named `kitchensink` and `helloworld`, and illustrate how to achieve the same web application using different underlying technologies. The source code for all of the quickstarts can also be obtained from <http://www.jboss.org/developer-materials/> on the JBoss Developer website and the source for a subset of quickstarts can also be accessed directly within JBoss Developer Studio.

ticketmonster.zip

Contains one application, TicketMonster. It is a moderately complex application that demonstrates how to build modern web applications optimized for mobile and desktop using an array of different technologies. TicketMonster is representative of an online ticketing broker, providing access to events with an online booking application. The source code for this application can also be obtained from <http://www.jboss.org/ticket-monster/> on the JBoss Developer website.

[Report a bug](#)

6.2. Use the Examples with the Command Line

Once the source for the examples is obtained, you can build the application with Maven and deploy it to a platform. The examples are all web applications so you can view the running application in a web browser. The procedure details the steps for building, deploying to JBoss EAP and viewing an application.

The procedure assumes that JBoss EAP is already installed and Maven is configured to use the JBoss EAP and JBoss Web Framework Kit Maven repositories.

Procedure 6.1. Use the Examples with the Command Line

1. On the command line, navigate to the directory of a specific example, where the `pom.xml` file for the example is located.
2. Compile the Maven-based example to verify it builds successfully

```
$ mvn compile
```

3. Start JBoss EAP

```
For Linux and OS X $ EAP_HOME/bin/standalone.sh
For Windows       $ EAP_HOME\bin\standalone.bat
```

4. Build and deploy the application to JBoss EAP

```
$ mvn clean install jboss-as:deploy
```

5. View the deployed application in a web browser at http://localhost:8080/application_name/, where *application_name* is replaced by the name of your application.

**Note**

Some examples have additional deployment requirements and steps. If the above procedure does not result in successful deployment of an example, check the **README.md** file distributed with the example for any additional deployment information.

[Report a bug](#)

6.3. Use the Examples with JBoss Developer Studio

You can generate a subset of quickstart examples within JBoss Developer Studio; additionally, the source for all examples can be imported into JBoss Developer Studio. The procedures below outline both methods but you only need to use one. Details are also given for compiling and deploying applications to JBoss EAP using the workflow shortcuts provided by JBoss Developer Studio.

The procedures assume that JBoss EAP is already installed and Maven is configured to use the JBoss EAP and JBoss Web Framework Kit Maven repositories. They further assume that JBoss Developer Studio is installed and a server adapter has been generated for the JBoss EAP instance to enable efficient IDE-server communication. For information on configuring JBoss Developer Studio for use with JBoss EAP and JBoss Web Framework Kit, see the JBoss Web Framework Kit Start Developing guide.

Procedure 6.2. Create an Example Using a Project Wizard

1. In JBoss Central, under **Start from a sample**, select an application category and click one of the samples.
2. In the **New Project Example** wizard, click **Finish**. Once created, the project is listed in the **Project Explorer** view.

Procedure 6.3. Import an Example Using an Import Wizard

1. Click **File** → **Import**, expand **Maven** and double-click **Existing Maven Projects**.
2. In the **Root Directory** field, enter the path to the example source code. Alternatively, click **Browse** to navigate to the source code location. The wizard recursively searches the path for a **pom.xml** file, which identifies the project as a Maven project. The **pom.xml** file is listed under **Projects** once it is found.
3. Click **Finish**. Once imported, the project is listed in the **Project Explorer** view.

Procedure 6.4. Compile and Deploy an Application Using JBoss Server Tools

1. In the **Project Explorer** view, right-click the project name and click **Run as** → **Run on Server**.
2. Under **How do you want to select the server?**, ensure **Choose an existing server** is selected.
3. In the **Server** table, expand **localhost**, select **jboss-eap-version** where *version* denotes the JBoss EAP version, and click **Next**.
4. Ensure that your project is listed in the **Configured** column and click **Finish**. The **Console** view automatically becomes the view in focus and displays the output from the JBoss EAP server. Once deploying is complete, the web application opens in the default IDE web browser.

[Report a bug](#)

Chapter 7. Upgrade from Earlier Versions of JBoss Web Framework Kit

7.1. Upgrade from Earlier Versions of JBoss Web Framework Kit

This release of JBoss Web Framework Kit is compatible with earlier releases.

To use this release concurrent with earlier JBoss Web Framework Kit releases, add an additional profile to the existing Maven `settings.xml` file enabling the Maven repository corresponding to this release.

To use this release with projects created using earlier JBoss Web Framework Kit releases, update the JBoss stacks BOM versions and any explicit dependency versions in your project `pom.xml` files. Note that upgrading from JBoss Web Framework Kit 2.3.0 requires additional dependency changes because the JBoss stacks were refactored in the 2.4.0 release.

See Also:

- [Section 7.2, “Upgrade JBoss Stacks from JBoss Web Framework Kit 2.3.0 and Earlier”](#)

[Report a bug](#)

7.2. Upgrade JBoss Stacks from JBoss Web Framework Kit 2.3.0 and Earlier

The group, artifact, and version properties of various JBoss stacks distributed with JBoss Web Framework Kit were refactored in the 2.4.0 release. To use this release with projects created using JBoss Web Framework Kit 2.3.0 and earlier, you must update the JBoss stacks BOM versions and any explicit dependency versions and make the following additional changes in your project `pom.xml` files:

- JBoss stacks previously distributed in the `org.jboss.bom` group moved to the `org.jboss.bom.wfk` group. If you use any of the stacks from the previous group in your projects, update the `groupId` in the dependency declarations accordingly.
- The JBoss stack `org.jboss.bom:jboss-javaee-6.0-with-hibernate` previously distributed in the JBoss Web Framework Kit Maven repository has been removed.
 - To use Hibernate in your projects, use the JBoss stack `org.jboss.bom.eap:jboss-javaee-6.0-with-hibernate` from the JBoss EAP Maven repository.
 - To use Hibernate together with Hibernate Search, use the JBoss stack `org.jboss.bom.wfk:jboss-javaee-6.0-with-hsearch` from the JBoss Web Framework Kit Maven repository.
- The JBoss stack `org.jboss.bom:jboss-bom-parent` has been replaced by `org.jboss.bom.wfk:jboss-wfk-bom-parent`. If you use the original stack in your projects, update the `groupId` and `artifactId` in the dependency management declarations accordingly.
- The JBoss stack `org.jboss.bom.wfk:jboss-javaee-6.0-with-spring` has been introduced. If you use the JBoss stack `com.redhat.jboss.wfk.boms:spring-3.2-bom` in your projects, it is recommended to replace it with the JBoss stack `jboss-javaee-6.0-with-spring`.

[Report a bug](#)

Revision History

| | | |
|-------------------------|-------------------------|------------------------|
| Revision 2.7.0-1 | Tues Jan 06 2015 | Michelle Murray |
|-------------------------|-------------------------|------------------------|

Generated for WFK 2.7 release