



Red Hat JBoss Fuse Service Works 6.0

Installation Guide

Use this guide to install Red Hat JBoss Fuse Service Works, and perform initial configuration.

Red Hat JBoss Fuse Service Works 6.0 Installation Guide

Use this guide to install Red Hat JBoss Fuse Service Works, and perform initial configuration.

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Abstract

This guide teaches you how to install the Red Hat JBoss Fuse Service Works product.

Table of Contents

CHAPTER 1. RED HAT JBOSS FUSE SERVICE WORKS	3
1.1. WHAT IS RED HAT JBOSS FUSE SERVICE WORKS?	3
1.2. CORE FUNCTIONALITY	3
1.3. SYSTEM INTEGRATION	4
1.4. CORE AND COMPONENTS	4
1.5. COMPONENTS OF RED HAT JBOSS FUSE SERVICE WORKS	4
1.6. RED HAT JBOSS FUSE SERVICE WORKS FEATURES	5
1.7. INTEGRATION USE CASE	6
CHAPTER 2. READ ME	7
2.1. BACK UP YOUR DATA	7
2.2. RED HAT DOCUMENTATION SITE	7
2.3. EAP_HOME	7
2.4. MODE	7
CHAPTER 3. DOWNLOADING RED HAT JBOSS FUSE SERVICE WORKS	8
3.1. ABOUT THE RED HAT CUSTOMER PORTAL	8
3.2. JBOSS FUSE SERVICE WORKS 6.0.0 PACKAGES FOR DOWNLOAD	8
3.3. DOWNLOAD FILES FROM THE RED HAT CUSTOMER PORTAL	9
3.4. CHECKSUM VALIDATION	9
3.5. VERIFY DOWNLOADED FILES	10
CHAPTER 4. INSTALLING RED HAT JBOSS FUSE SERVICE WORKS	11
4.1. INSTALLING JBOSS FUSE SERVICE WORKS BY GUI	11
4.1.1. Configure Advanced Settings	14
4.2. INSTALLING JBOSS FUSE SERVICE WORKS BY SCRIPT	15
4.2.1. Supplying Passwords to the Installation Script	16
CHAPTER 5. STARTING AND STOPPING THE APPLICATION SERVER	18
5.1. START JBOSS EAP 6 AS A STANDALONE SERVER	18
5.2. STOP JBOSS EAP 6 AS A STANDALONE SERVER	18
CHAPTER 6. MAVEN REPOSITORIES	20
6.1. ABOUT MAVEN	20
6.2. ABOUT THE PROVIDED MAVEN REPOSITORIES	20
6.3. CONFIGURING MAVEN TO USE THE FILE SYSTEM REPOSITORIES	21
6.4. CONFIGURING MAVEN TO USE THE ONLINE REPOSITORIES	28
6.5. DEPENDENCY MANAGEMENT	31
CHAPTER 7. INSTALLING JBOSS DEVELOPER STUDIO	33
7.1. INSTALL JBOSS DEVELOPER STUDIO	33
7.2. INSTALLING JBOSS DEVELOPER STUDIO INTEGRATION STACK	36
APPENDIX A. PREREQUISITE SOFTWARE	39
A.1. INSTALL OPEN JDK ON RED HAT LINUX	39
A.2. INSTALL MAVEN	39
APPENDIX B. VERIFY YOUR RED HAT JBOSS FUSE SERVICE WORKS INSTALLATION	42
APPENDIX C. REVISION HISTORY	43

CHAPTER 1. RED HAT JBOSS FUSE SERVICE WORKS

1.1. WHAT IS RED HAT JBOSS FUSE SERVICE WORKS?

Red Hat JBoss Fuse Service Works is a platform for developing enterprise application integration (EAI) and service-oriented architecture (SOA) solutions. It consists of a service component framework, business rules/complex event processing, life-cycle governance, runtime governance, and process automation. Red Hat JBoss Fuse Service Works is built on the same core as JBoss Fuse, and includes enterprise messaging, Apache Camel, and Apache CXF. Red Hat JBoss Fuse Service Works enables users to design, deploy, integrate, and orchestrate business services.

Red Hat JBoss Fuse Service Works can be used to integrate your major business systems into a cohesive infrastructure. Development is simplified with a transparent, lightweight service framework which uses Enterprise Integration Platform (EIP) technology. This allows developers to work with familiar technologies such as Apache Camel, BPEL, BPMN, or POJOs. To reduce the operational costs of production and maintenance, the platform utilizes an automatable, content-aware repository and service activity monitoring. These support the entire service life cycle.

[Report a bug](#)

1.2. CORE FUNCTIONALITY

Red Hat JBoss Fuse Service Works provides the following core functionality:

Enterprise Integration Pattern (EIP) Based Development

The versatile EIP framework is implemented in routing and transformation processes for faster and more efficient integration solutions.

High Performance Messaging

A high performance messaging broker supports messaging patterns such as publish-subscribe, point-to-point, and store-forward, and multiple cross language clients.

Service Development

The web services framework exposes integration assets as services and calls external services, supporting all major web services standards. It also supports RESTful calls.

Structured Service Development

A lightweight service development framework provides full life-cycle support for developing, deploying, and managing service-based applications.

Automatable Registry with Workflow

Manage the life-cycle of services from design, development, and deployment by defining, exposing, and enforcing rules or policies.

Business Transaction Monitoring

Capture service activity information, define and collect metrics, and define alerts and SLAs.

[Report a bug](#)

1.3. SYSTEM INTEGRATION

Integrating your major business systems into a cohesive infrastructure can be a challenge, especially when you have legacy applications. Red Hat JBoss Fuse Service Works provides a number of ways that enable you to integrate both new and legacy applications. Development is simplified with a transparent, lightweight service framework which uses the EIP technology. This allows developers to focus on higher order concepts while still working with familiar technologies such as Apache Camel, BPEL, BPMN, or POJOs. To reduce the operational costs of production and maintenance, the platform utilizes an automatable, content-aware repository and service activity monitoring. These support the entire service life cycle and development, QA, and production teams with runtime and design time visibility, monitoring, and alerting.

[Report a bug](#)

1.4. CORE AND COMPONENTS

Red Hat JBoss Fuse Service Works provides an environment for easily applying SOA concepts to integrated applications. A SwitchYard application consists of components such as composite services and composite references. These provide service definitions and accessibility.

Along with SwitchYard, Red Hat JBoss Fuse Service Works is made up of a number of components including a rules-based router (Apache Camel), a web service framework (Apache CXF), and a message broker (HornetQ).

[Report a bug](#)

1.5. COMPONENTS OF RED HAT JBOSS FUSE SERVICE WORKS

Red Hat JBoss Fuse Service Works ships with a number of components which enable its multi-functional capabilities.

Table 1.1. Red Hat JBoss Fuse Service Works Components

Component	Function
SwitchYard	Service delivery framework
JBoss Rules	Business rules engine with complex event processing
Design Time Governance	A service registry/repository
Runtime Governance	Service activity monitoring
JBoss Operations Network	Operations, administration, and management tools
JBoss EAP	A full JavaEE application server
Apache Camel	Rules Based Router
Smooks	Framework for processing XML and non-XML data using Java
ModeShape	Data Store

Component	Function
HornetQ	Messaging and Integration Patterns Server
Apache CXF	Services Framework

These components can be used in Red Hat JBoss Fuse Service Works to enable developers to build the required functionality using reliable and familiar tools. Some examples of how the components can be used are shown below:

Bean Services with CDI

SwitchYard leverages the power of Java EE6 and CDI to allow Java objects become services by adding an `@Service` annotation to your bean. Beans are automatically registered at runtime and references to other services can be injected as CDI beans using the `@Inject` annotation. Use CDI in your JSP and JSF applications to inject enterprise services into the web tier.

Declarative Transformation

With declarative transformation in SwitchYard, you can define the transformation and types to which it applies. SwitchYard automatically registers and executes the transformation. Choose from Smooks, Java, XSLT, JSON, and more.

Decision Services with JBoss Rules

Encapsulate business rules as decision services using the JBoss Rules component in SwitchYard. Each service has a well-defined contract with protocol binding details and marshaling details abstracted away by SwitchYard.

Smooks

This transformation engine can be used in conjunction with Red Hat JBoss Fuse Service Works to process messages.

Business Process Execution Language (BPEL)

You can use web services to orchestrate business rules using this language. It is included with Red Hat JBoss Fuse Service Works for the execution of business process instructions.

JBoss Rules

This is the rules engine that is packaged with Red Hat JBoss Fuse Service Works. It can infer data from the messages it receives to determine which actions need to be performed.

Testing

Comprehensive unit test support is provided to allow you to test services as you develop them.

[Report a bug](#)

1.6. RED HAT JBOSS FUSE SERVICE WORKS FEATURES

SwitchYard

SwitchYard is a lightweight service delivery framework providing full life-cycle support for developing, deploying, and managing service-oriented applications.

Business Process Execution Language (BPEL)

You can use web services to orchestrate business rules using this language. It is included with Red Hat JBoss Fuse Service Works for the execution of business process instructions.

Smooks

This transformation engine can be used in conjunction with Red Hat JBoss Fuse Service Works to process messages. It can also be used to split messages and send them to the correct destination.

JBoss Rules

This is the rules engine that is packaged with Red Hat JBoss Fuse Service Works. It can infer data from the messages it receives to determine which actions need to be performed.

[Report a bug](#)

1.7. INTEGRATION USE CASE

Acme Equity is a large financial service. The company possesses many databases and systems. Some are older, COBOL-based legacy systems and some are databases obtained through the acquisition of smaller companies in recent years. It is challenging and expensive to integrate these databases as business rules frequently change. The company wants to develop a new series of client-facing e-commerce websites, but these may not synchronize well with the existing systems as they currently stand.

The company wants an inexpensive solution but one that adheres to the strict regulations and security requirements of the financial sector. What the company does not want to do is to have to write and maintain “glue code” to connect their legacy databases and systems.

Red Hat JBoss Fuse Service Works was selected as a middleware layer to integrate these legacy systems with the new customer websites. It provides a bridge between front-end and back-end systems. Business rules implemented with Red Hat JBoss Fuse Service Works can be updated quickly and easily.

As a result, older systems can now synchronize with newer ones due to the unifying methods of Red Hat JBoss Fuse Service Works. There are no bottlenecks, even with tens of thousands of transactions per month. Various integration types, such as XML, JMS and FTP, are used to move data between systems. Any one of a number of enterprise-standard messaging systems can be plugged into Red Hat JBoss Fuse Service Works providing further flexibility.

An additional benefit is that the system can now be scaled upwards easily as more servers and databases are added to the existing infrastructure.

[Report a bug](#)

CHAPTER 2. READ ME

2.1. BACK UP YOUR DATA



WARNING

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

[Report a bug](#)

2.2. RED HAT DOCUMENTATION SITE

Red Hat's official documentation site is at <https://access.redhat.com/site/documentation/>. There you will find the latest version of every book, including this one.

[Report a bug](#)

2.3. EAP_HOME

EAP_HOME refers to the root directory of the Red Hat JBoss Enterprise Application Platform installation on which JBoss Fuse Service Works is deployed.

[Report a bug](#)

2.4. MODE



NOTE

Please note that Red Hat JBoss Fuse Service Works 6.0 does not support Red Hat JBoss Enterprise Application Platform run in Domain mode.

MODE will either be **standalone** or **domain** depending on whether your instance of JBoss Enterprise Application Platform is running in standalone or domain mode. Substitute one of these whenever you see **MODE** in a file path in this documentation.

[Report a bug](#)

CHAPTER 3. DOWNLOADING RED HAT JBOSS FUSE SERVICE WORKS

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. JBOSS FUSE SERVICE WORKS 6.0.0 PACKAGES FOR DOWNLOAD

Table 3.1.

Package	Description
Red Hat JBoss Fuse Service Works 6.0.0 Javadocs for Runtime Governance	Javadocs for Runtime Governance.
Red Hat JBoss Fuse Service Works 6.0.0 Javadocs for S-RAMP	Javadocs for S-RAMP.
Red Hat JBoss Fuse Service Works 6.0.0 Javadocs for SwitchYard	Javadocs for SwitchYard.
Red Hat JBoss Fuse Service Works 6.0.0 Maven Repository	Red Hat provides a Maven repository containing artifacts required to build applications for Red Hat JBoss Fuse Service Works. This package enables users to setup the repository offline.
Red Hat JBoss Fuse Service Works 6.0.0 Source Code	The source code package contains the complete source code for the Red Hat JBoss Fuse Service Works product.
Red Hat JBoss Fuse Service Works 6.0.0 Installer	Red Hat provides this tool to simplify product installation. The installer can be run with a graphical interface or from a script on the command line.

[Report a bug](#)

3.3. DOWNLOAD FILES FROM THE RED HAT CUSTOMER PORTAL

Prerequisites

- Before you begin this task, you need a Customer Portal account. Browse to <https://access.redhat.com> and click the **Register** link in the upper right corner to create an account.

Procedure 3.1. Log in and Download Files from the Red Hat Customer Portal

1. Browse to <https://access.redhat.com> and click the **Log in** link in the top right corner. Enter your credentials and click **Log In**.

Result

You are logged into RHN and you are returned to the main web page at <https://access.redhat.com>.

2. **Navigate to the Downloads page.**

Use one of the following options to navigate to the **Downloads** page.

- Click the **Downloads** link in the top navigation bar.
- Navigate directly to <https://access.redhat.com/downloads/>.

3. **Select the product and version to download.**

Use one of the following ways to choose the correct product and version to download.

- Step through the navigation one level at a time.
- Search for your product using the search area at the top right-hand side of the screen.

4. **Download the appropriate file for your operating system and installation method of choice.**

Depending on the product you choose, you may have the choice of a Zip archive, RPM, or native installer for a specific operating system and architecture. Click either the file name or the **Download** link to the right of the file you want to download.

Result

The file is downloaded to your computer.

[Report a bug](#)

3.4. CHECKSUM VALIDATION

Checksum validation is used to ensure a downloaded file has not been corrupted. Checksum validation employs algorithms that compute a fixed-size datum (or checksum) from an arbitrary block of digital data. If two parties compute a checksum of a particular file using the same algorithm, the results will be identical. Therefore, when computing the checksum of a downloaded file using the same algorithm as the supplier, if the checksums match, the integrity of the file is confirmed. If there is a discrepancy, the file has been corrupted in the download process.

[Report a bug](#)

3.5. VERIFY DOWNLOADED FILES

Procedure 3.2. Verify File Checksums on Red Hat Enterprise Linux

1. Obtain checksum values for the downloaded file

- a. Navigate to <https://access.redhat.com/jbossnetwork/>. Login if required.
- b. Select your **Product** and **Version**.
- c. Select the package you want to verify to navigate to the **Software Details** page.
- d. Take note of the **MD5** and **SHA - 256** checksum values.

2. Run a checksum tool on the file

- a. Navigate to the directory containing the downloaded file in a terminal window.
- b. Run `md5sum downloaded_file`.
- c. Run `sha256sum downloaded_file`.

Example output:

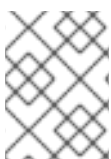
```
[localhost]$ md5sum downloaded_file
4564d1a5190110dbe8170e50d7353a97 downloaded_file
[localhost]$ sha256sum downloaded_file
25b6bd3c5f47b316639b014d041cdb6a515e3a4a32d30a479141cd8ceecb853e
downloaded_file
```

3. Compare the checksum values

- a. Compare the checksum values returned by the `md5sum` and `sha256sum` commands with the corresponding values displayed on the **Software Details** page.
- b. Download the file again if the two checksum values are not identical. A difference between the checksum values indicates that the file has either been corrupted during download or has been modified since it was uploaded to the server. Contact Red Hat Support for assistance if after several downloads the checksum does not successfully validate.
- c. The downloaded file is safe to use if the two checksum values are identical.

Result

The integrity of the downloaded file is verified.



NOTE

No checksum tool is included with **Microsoft Windows**. Download a third-party MD5 application such as **MD5summer** from <http://www.md5summer.org/>.

[Report a bug](#)

CHAPTER 4. INSTALLING RED HAT JBOSS FUSE SERVICE WORKS

JBoss Fuse Service Works provides an installer, `jboss-fsw-installer-6.0.0.GA-redhat-4.jar`, which can be used to initiate either a graphical or script-based installation.

With the installer, users can choose to install one or more of the following:

- JBoss EAP (mandatory)
 - including HornetQ
- SwitchYard - selecting this component also installs the following components:
 - jBPM
 - Drools
 - RiftSaw (BPEL)
 - Smooks
 - Apache Camel
 - Apache CXF
- Runtime Governance Client (must be accompanied by SwitchYard)
- Runtime Governance Server and Client
- Design Time Governance (must be accompanied by S-RAMP Repository)
- S-RAMP Repository
- Red Hat JBoss Fuse Service Works quickstarts

[Report a bug](#)

4.1. INSTALLING JBOSS FUSE SERVICE WORKS BY GUI

Prerequisites

- You must have already downloaded the Red Hat JBoss Fuse Service Works installer file from [Red Hat Customer Portal](#).
- You must have a supported JDK installed. See [Red Hat JBoss Fuse Service Works Supported Configurations](#).
- If you would like to configure use of the online Maven repository during this installation, you must have Maven already installed.



IMPORTANT

The set of online remote repositories is a technology preview source of components. As such, it is not in scope of patching and is supported only for use in development environment. Using the set of online repositories in production environment is a potential source of security vulnerabilities and is therefore not a supported use case. For more information see <https://access.redhat.com/site/maven-repository>.



NOTE

If you have multiple versions of Java installed on your server, Red Hat recommends setting `PATH`, `JAVA_HOME`, and `JBOSS_HOME` to ensure the correct version of Java is used at runtime:

- Set `JAVA_HOME` to the version of Java. For example, `/usr/lib/jvm/java-1.7.0-openjdk-1.7.0.VERSION/`.
- Set `PATH` to include the same version of Java.
- Set `JBOSS_HOME` to your `jboss-eap-6.1` directory.

Procedure 4.1. Install JBoss Fuse Service Works

1. Open a new terminal and navigate to the folder where you have downloaded the JBoss Fuse Service Works installer file.
2. Run the following command, replacing `6.0.0.GA-redhat-4` for the version you downloaded if required:

```
java -jar jboss-fsw-installer-6.0.0.GA-redhat-4.jar
```

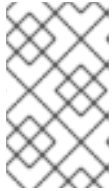
3. You are presented with the End User License Agreement. If you agree with the terms, select **I accept the terms of this license agreement** and then select **Next** to continue.
4. You are prompted for the location in which you want to install JBoss Fuse Service Works. Specify a location and select **Next**. You will be notified that an instance of JBoss EAP will be placed in the chosen location. Select **OK** to proceed.



WARNING

A warning message is displayed if the directory already exists. Red Hat does not recommend installing over the top of an existing installation. However, if you wish to proceed, select **Yes** to continue.

5. You are prompted for the installation packs to install. Select **Next** to continue with the default options.

**NOTE**

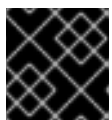
Depending on your deployment, you can select packs other than the default options. For example, if you do not wish to install every pack, you could select SwitchYard and Runtime Governance Client only, for a basic installation.

6. You are prompted to create an administrative user. Once created, this user will be added to the ManagementRealm and can be used to access the Management Console and other applications secured using ManagementRealm. Enter the new username and password in the appropriate fields and select **Next**.
7. You are prompted to create a governance user. It will give you access to S-RAMP, DTGov and RTGov consoles along with the S-RAMP Command Line Interface. Enter a username and password and select **Next**.
8. You are then prompted to setup your Maven repository. This will enable you to build the quickstarts provided. Specify the location of an existing or new location for your Maven settings file (typically `~/.m2/settings.xml`). If you do not currently have a settings file, a new `settings.xml` file will be created for you in the location you provide. Select **Next**.

**NOTE**

If you choose to skip the maven repository setup, you will be asked to confirm this decision. Select **Yes** to continue.

9. You are prompted to configure the SAML keystore. It is used by a Java KeyStore to sign authentication tokens. Enter the SAML keystore password and select **Next**.
10. You are then prompted to enable the Java Security Manager. It enforces access rules at the JVM runtime based on one or more security polices. Select **Next** when you are ready to proceed.

**IMPORTANT**

Enabling the Java Security Manager may reduce server performance.

11. At this point you are asked to choose between the default or advanced configuration. Select the default configuration for a basic configuration, and select **Next** to continue.
12. You are prompted to configure the password vault. Enter the Vault keystore password in the appropriate fields and select **Next**.
13. You are presented with a screen specifying default database settings. The installer creates a database that is used to hold data for various components of JBoss Fuse Service Works, including BPEL, jBPM and S-RAMP.

Change the username and password to something secure and select **Next** to continue.



WARNING

The default database is an **H2** database. Red Hat recommends configuring an alternative database for production systems, using the advanced configuration options.

14. You are presented with a list of packages that will be installed. Select **Next** to proceed.
15. The installation will commence. A status bar for each component will display its progress. Once this is complete, select **Next**.
16. Additional processing of tasks will commence. When the status bar indicates this has been completed, select **Next**.
17. The console displays a message to let you know that the application has been successfully installed. Also, it offers you a list of URLs for accessing the Administration and BPEL Consoles, S-RAMP and DTGov User Interfaces and Gadget Server.

If you want to repeat the same installation on other machines, select **Generate installation script and properties file** and choose a location and filename for the script.

18. Select **Done**.

Result

The basic installation is complete.



NOTE

When running the installer on Windows, you may be prompted to provide administrator credentials during installation. To prevent this from happening, add the **izpack.mode=privileged** option to the installation command:

```
java -Dizpack.mode=privileged -jar jboss-fsw-installer-6.0.0.GA-redhat-4.jar
```

[Report a bug](#)

4.1.1. Configure Advanced Settings

During the JBoss Fuse Service Works installation, you can choose between a default or advanced configuration. This section provides additional instructions if you choose to proceed with the advanced configuration.

Procedure 4.2. Task

1. After choosing **Perform advanced configuration**, select the items you want to configure and select **Next**.

2. If you chose the **Install password vault** option, the **Configure password vault** screen appears. This mechanism encrypts sensitive strings and stores them in an encrypted keystore.

Enter your keystore information in the required fields and select **Next**.

3. If you chose the **Enable SSL security** option, the **SSL Security** screen appears. Enter the location and password of a keystore under the respective fields. Confirm the password and select **Next**.
4. If you chose the **Enable LDAP authentication** option, the **LDAP Configuration** screen appears. You can use the LDAP server as an authentication and authorization authority, both for applications and the management interface.

Enter the required information to suit your system requirements and select **Next**.

5. A screen displaying the new security realm configurations appears. Select **Next** to continue.
6. If you chose the **Install JDBC Driver** option, the **JDBC Driver Setup** screen appears. You can setup your database connection using this screen.

Type the Database Username and Password under the respective fields. Confirm the password and select **Next**.



NOTE

JBoss Fuse Service Works requires a database to hold data for various components, including BPEL, jBPM and S-RAMP. It is used to hold state information that must be persisted from action to action so that recovery is possible in case of a server crash.

Clustering does not rely upon this database in the default configuration. Instead, Infinispan is used as a memory grid.

[Report a bug](#)

4.2. INSTALLING JBOSS FUSE SERVICE WORKS BY SCRIPT

In addition to running the installer GUI, you can also run the installer by script. This method of installation removes the need to step through the graphical installer for every installation, enabling simplified repeat installations or installations in bulk.

Prerequisites

In order to install JBoss Fuse Service Works by script, you must have already run the installer in graphical mode, and saved the installation script when prompted at the end of the installation. The script will have been saved in **EAP_HOME**, along with an associated **.variables** file; for example, **InstallConfigRecord.xml** and **InstallConfigRecord.xml.variables**.

Procedure 4.3.

1. You can change the installation directory by editing the script and modifying the value specified within the **<installpath>** element.

**NOTE**

You can specify either a full directory path or a path relative to the directory in which you run the script.

**WARNING**

Red Hat does not support installing over an existing instance of Fuse Service Works. Therefore be sure to select a new or empty directory.

2. Run the script using the following command, replacing *6.0.0.GA-redhat-4* for the version you downloaded if required:

```
java -jar jboss-fsw-installer-6.0.0.GA-redhat-4.jar
InstallConfigRecord.xml
```

The script runs and prompts you to provide Governance, SAML keystore, and Vault keystore passwords, as shown below:

```
[ Starting automated installation ]
Read pack list from xml definition.
Try to add to selection [Name: Red Hat JBoss Enterprise Application
Platform and Index: 0]
Try to add to selection [Name: Red Hat JBoss Fuse Service Works and
Index: 1]
Try to add to selection [Name: Switchyard quickstarts and Index: 2]
Try to remove from selection [Name: RT-Gov Client-only and Index: 3]
Try to add to selection [Name: RT-Gov Server+Client and Index: 4]
Try to add to selection [Name: RTGov Server quickstarts and Index:
5]
Try to add to selection [Name: DT-Gov run-time and Index: 6]
Try to add to selection [Name: DT-Gov quickstarts and Index: 7]
Try to add to selection [Name: S-RAMP run-time and Index: 8]
Try to add to selection [Name: S-RAMP quickstarts and Index: 9]
Try to add to selection [Name: Red Hat JBoss Fuse Service Works
quickstarts and Index: 10]
Modify pack selection.
Pack [Name: RT-Gov Client-only and Index: 3] removed from selection.
Governance password:
Confirm Governance password:
SAML keystore password:
Re-enter SAML keystore password:
Vault keystore password:
Re-enter vault keystore password:
```

[Report a bug](#)

4.2.1. Supplying Passwords to the Installation Script

When installing JBoss Fuse Service Works by script, the user will be prompted to supply various passwords during installation by default.

If you would like a fully automated installation, add the passwords to the associated **.variables** file and run the installer with the **-variablefile** option. For example, assuming the script is named **InstallConfigRecord.xml**:

```
java -jar jboss-fsw-installer-6.0.0.GA-redhat-4.jar InstallConfigRecord.xml  
-variablefile InstallConfigRecord.xml.variables
```

You can also supply them directly on the command line as key/value pairs by using the **-variables** option:

```
java -jar jboss-fsw-installer-6.0.0.GA-redhat-4.jar InstallConfigRecord.xml  
-variables KEY_1=VALUE_1 KEY_2=VALUE_2 ...
```



WARNING

Red Hat recommends you remove plaintext passwords from the **.variables** file after installation to avoid compromise.

[Report a bug](#)

CHAPTER 5. STARTING AND STOPPING THE APPLICATION SERVER

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



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 Administration and Configuration Guide*.
- Note that JBoss Fuse Service Works 6.0 does not support JBoss Enterprise Application Platform in domain mode.

[Report a bug](#)

5.1. START JBOSS EAP 6 AS A STANDALONE SERVER

Summary

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

Procedure 5.1. 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](#)

5.2. STOP JBOSS EAP 6 AS A STANDALONE SERVER

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

1. **Stopping JBoss Enterprise Application Platform using the Management CLI.**
 - o Run the `EAP_HOME/bin/jboss-cli.sh` command to launch the Management CLI.

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

- o Run the `connect` command to connect to the server.

-

```
█ [disconnected /] connect
```

- Run the **shutdown** command to stop the server.

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

- Run the **quit** command to close the Management CLI.

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

2. Stopping JBoss Enterprise Application Platform by pressing CTRL+C.

Press **Ctrl+C** in the *server window* (the terminal window where JBoss Enterprise Application Platform was started).

Result

Each of these alternatives stops JBoss Enterprise Application Platform.

[Report a bug](#)

CHAPTER 6. MAVEN REPOSITORIES

6.1. ABOUT MAVEN

Apache Maven is a distributed build automation tool used in Java application development to build and manage software projects. Maven uses configuration XML files called POM (Project Object Model) to define project properties and manage the build process. POM files describe the project's module and component dependencies, build order, and targets for the resulting project packaging and output. This ensures that projects are built in a correct and uniform manner.

Maven uses repositories to store Java libraries, plug-ins, and other build artifacts. Repositories can be either local or remote. A local repository is a download of artifacts from a remote repository cached on a local machine. A remote repository is any other repository accessed using common protocols, such as **http://** when located on an HTTP server, or **file://** when located on a file server. The default repository is the public remote [Maven 2 Central Repository](#).

Configuration of Maven is performed by modifying the **settings.xml** file. You can either configure global Maven settings in the **M2_HOME/conf/settings.xml** file, or user-level settings in the **USER_HOME/.m2/settings.xml** file.

For more information about Maven, see [Welcome to Apache Maven](#).

For more information about Maven repositories, see [Apache Maven Project - Introduction to Repositories](#).

For more information about Maven POM files, see the [Apache Maven Project POM Reference](#).



NOTE

Red Hat JBoss Fuse Service Works has been built with maven 3.0.x Therefore, this is the recommended maven version for building your own SwitchYard applications.

[Report a bug](#)

6.2. ABOUT THE PROVIDED MAVEN REPOSITORIES

A set of repositories containing artifacts required to build applications based on Red Hat JBoss Fuse Service Works is provided with this release. Maven must be configured to use these repositories and the Maven Central Repository in order to provide correct build functionality.

Two interchangeable sets of repositories ensuring the same functionality are provided. The first set is available for download and storage in a local file system, the second set is hosted online for use as remote repositories. If you provided the location of Maven's **settings.xml** file during installation, Maven is already configured to use the online repositories. If you did not provide the location during installation, you need to configure Maven manually by following the procedure in [Section 6.4](#), “Configuring Maven to Use the Online Repositories” or [Section 6.3](#), “Configuring Maven to Use the File System Repositories”.



IMPORTANT

The set of online remote repositories is a technology preview source of components. As such, it is not in scope of patching and is supported only for use in development environment. Using the set of online repositories in production environment is a potential source of security vulnerabilities and is therefore not a supported use case. For more information see <https://access.redhat.com/site/maven-repository>.

[Report a bug](#)

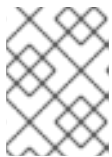
6.3. CONFIGURING MAVEN TO USE THE FILE SYSTEM REPOSITORIES

Overview

In situations where you cannot use the online repositories, you will have to download and configure the required repositories locally.

Procedure 6.1.

1. Download the following ZIP archives containing the required repositories:
 - <http://maven.repository.redhat.com/techpreview/fsw6/6.0.0/fsw-6.0.0.GA-redhat-2-repository.zip>
 - <http://maven.repository.redhat.com/techpreview/eap6/6.1.1/jboss-eap-6.1.1-maven-repository.zip>



NOTE

Alternatively, the ZIP archives can be downloaded from <http://access.redhat.com/jbossnetwork/>.

2. Unzip the downloaded ZIP files into an arbitrary location in a local file system.
3. Add entries for the unzipped repositories to Maven's **settings.xml** file. The following code sample contains a profile with the repositories and an activation entry for the profile:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/xsd/settings-1.0.0.xsd">
  <localRepository/>
  <profiles>
    <!-- Profile with local repositories required by Fuse Service
    Works -->
    <profile>
      <id>fsw-local-repos</id>
      <repositories>
        <repository>
          <id>fsw-6.0.0.GA-redhat-2-repository</id>
          <name>FSW 6.0.0.GA Repository</name>
          <url>file://<!-- path to the repository -->/fsw-6.0.0.GA-
redhat-2-repository</url>
          <layout>default</layout>
```

```

    <releases>
      <enabled>true</enabled>
      <updatePolicy>never</updatePolicy>
    </releases>
  </snapshots>
    <enabled>>false</enabled>
    <updatePolicy>never</updatePolicy>
  </snapshots>
</repository>
<repository>
  <id>jboss-eap-6.1.1.GA-maven-repository</id>
  <name>EAP 6.1.1.GA Repository</name>
  <url>file://<!-- path to the repository -->/jboss-eap-
6.1.1.GA-maven-repository</url>
  <layout>default</layout>
  <releases>
    <enabled>true</enabled>
    <updatePolicy>never</updatePolicy>
  </releases>
  <snapshots>
    <enabled>>false</enabled>
    <updatePolicy>never</updatePolicy>
  </snapshots>
</repository>
</repositories>
<pluginRepositories>
  <pluginRepository>
    <id>fsw-6.0.0.GA-redhat-2-repository</id>
    <name>FSW 6.0.0.GA Repository</name>
    <url>file://<!-- path to the repository -->/fsw-6.0.0.GA-
redhat-2-repository</url>
    <layout>default</layout>
    <releases>
      <enabled>true</enabled>
      <updatePolicy>never</updatePolicy>
    </releases>
    <snapshots>
      <enabled>>false</enabled>
      <updatePolicy>never</updatePolicy>
    </snapshots>
  </pluginRepository>
  <pluginRepository>
    <id>jboss-eap-6.1.1.GA-maven-repository</id>
    <name>EAP 6.1.1 GA Repository</name>
    <url>file://<!-- path to the repository -->/jboss-eap-
6.1.1.GA-maven-repository</url>
    <layout>default</layout>
    <releases>
      <enabled>true</enabled>
      <updatePolicy>never</updatePolicy>
    </releases>
    <snapshots>
      <enabled>>false</enabled>
      <updatePolicy>never</updatePolicy>
    </snapshots>
  </pluginRepository>

```

```

        </pluginRepositories>
    </profile>
</profiles>
<activeProfiles>
    <!-- Activation of the Fuse Service Works profile -->
    <activeProfile>fsw-local-repos</activeProfile>
</activeProfiles>
</settings>

```

IMPORTANT

The **bpm-service** quickstart application will not build without two additional repositories. If you want to build this application, download the following repositories and add them to your **setting.xml** file in the same way as that shown above.

- <http://maven.repository.redhat.com/techpreview/eap6/6.0.1/jboss-eap-6.0.1-maven-repository.zip>
- <http://maven.repository.redhat.com/techpreview/wfk2/2.5.0/jboss-wfk-2.5.0-maven-repository.zip>

Result

The Maven repositories are downloaded, unzipped in a local file system, registered in Maven's **settings.xml** file, and ready to be used when performing Maven builds.

Troubleshooting

Q: Why does the bpm-service quickstart application fail on build?

A: Issue

When you build or deploy the **bpm-service** quickstart application, it fails with the following error:

```

[ERROR] Failed to execute goal on project switchyard-quickstart-bpm-
service: Could not resolve dependencies for project
org.switchyard.quickstarts:switchyard-quickstart-bpm-
service:jar:1.1.1-p5-redhat-1: The following artifacts could not be
resolved: org.jboss.netty:netty:jar:3.2.6.Final-redhat-2,
org.apache.lucene:lucene-queryparser:jar:3.6.2-redhat-4: Could not
find artifact org.jboss.netty:netty:jar:3.2.6.Final-redhat-2 in fsw-
6.0.0.GA-redhat-2-repository (file:///home/belong/maven_repos/fsw-
6.0.0.GA-redhat-2-repository) -> [Help 1]

```

Cause

The **bpm-service** quickstart application requires an additional two dependencies, provided by the JBoss EAP 6.0.1 and JBoss Web Framework Kit 2.5.0 Maven repositories.

Resolution

1. Download the following ZIP archives, and unzip them into a location of your choice:

<http://maven.repository.redhat.com/techpreview/eap6/6.0.1/jboss-eap-6.0.1-maven-repository.zip>

<http://maven.repository.redhat.com/techpreview/wfk2/2.5.0/jboss-wfk-2.5.0-maven-repository.zip>

2. Add entries for the unzipped repositories to your Maven `settings.xml` file:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/xsd/settings-
1.0.0.xsd">
  <localRepository/>
  <profiles>
    <!-- Profile with local repositories required by Fuse
Service Works -->
    <profile>
      <id>fsw-local-repos</id>
      <repositories>
        <repository>
          <id>fsw-6.0.0.GA-redhat-2-repository</id>
          <name>FSW 6.0.0.GA Repository</name>
          <url>file://<!-- path to the repository -->/fsw-
6.0.0.GA-redhat-2-repository</url>
          <layout>default</layout>
          <releases>
            <enabled>>true</enabled>
            <updatePolicy>never</updatePolicy>
          </releases>
          <snapshots>
            <enabled>>false</enabled>
            <updatePolicy>never</updatePolicy>
          </snapshots>
        </repository>
        <repository>
          <id>jboss-eap-6.1.1.GA-maven-repository</id>
          <name>EAP 6.1.1.GA Repository</name>
          <url>file://<!-- path to the repository -->/jboss-
eap-6.1.1.GA-maven-repository</url>
          <layout>default</layout>
          <releases>
            <enabled>>true</enabled>
            <updatePolicy>never</updatePolicy>
          </releases>
          <snapshots>
            <enabled>>false</enabled>
            <updatePolicy>never</updatePolicy>
          </snapshots>
        </repository>
        <repository>
          <id>jboss-wfk-2.5.0-maven-repository</id>
          <name>JBoss WFK 2.5.0 Repository</name>
          <url>file://<!-- path to the repository -->/jboss-
wfk-2.5.0-maven-repository</url>
          <layout>default</layout>
          <releases>
            <enabled>>true</enabled>
            <updatePolicy>never</updatePolicy>
```

```

        </releases>
        <snapshots>
            <enabled>>false</enabled>
            <updatePolicy>never</updatePolicy>
        </snapshots>
    </repository>
    <repository>
        <id>jboss-eap-6.0.1-maven-repository</id>
        <name>EAP 6.0.1 Repository</name>
        <url>file://<!-- path to the repository -->/jboss-
eap-6.0.1-maven-repository</url>
        <layout>default</layout>
        <releases>
            <enabled>>true</enabled>
            <updatePolicy>never</updatePolicy>
        </releases>
        <snapshots>
            <enabled>>false</enabled>
            <updatePolicy>never</updatePolicy>
        </snapshots>
    </repository>
</repositories>
<pluginRepositories>
    <pluginRepository>
        <id>fsw-6.0.0.GA-redhat-2-repository</id>
        <name>FSW 6.0.0.GA Repository</name>
        <url>file://<!-- path to the repository -->/fsw-
6.0.0.GA-redhat-2-repository</url>
        <layout>default</layout>
        <releases>
            <enabled>>true</enabled>
            <updatePolicy>never</updatePolicy>
        </releases>
        <snapshots>
            <enabled>>false</enabled>
            <updatePolicy>never</updatePolicy>
        </snapshots>
    </pluginRepository>
    <pluginRepository>
        <id>jboss-eap-6.1.1.GA-maven-repository</id>
        <name>EAP 6.1.1 GA Repository</name>
        <url>file://<!-- path to the repository -->/jboss-
eap-6.1.1.GA-maven-repository</url>
        <layout>default</layout>
        <releases>
            <enabled>>true</enabled>
            <updatePolicy>never</updatePolicy>
        </releases>
        <snapshots>
            <enabled>>false</enabled>
            <updatePolicy>never</updatePolicy>
        </snapshots>
    </pluginRepository>
    <pluginRepository>
        <id>jboss-wfk-2.5.0-maven-repository</id>
        <name>JBoss WFK 2.5.0 Repository</name>

```

```

        <url>file://<!-- path to the repository -->/jboss-
wfk-2.5.0-maven-repository</url>
        <layout>default</layout>
        <releases>
            <enabled>true</enabled>
            <updatePolicy>never</updatePolicy>
        </releases>
        <snapshots>
            <enabled>false</enabled>
            <updatePolicy>never</updatePolicy>
        </snapshots>
    </pluginRepository>
    <pluginRepository>
        <id>jboss-eap-6.0.1-maven-repository</id>
        <name>EAP 6.0.1 Repository</name>
        <url>file://<!-- path to the repository -->/jboss-
eap-6.0.1-maven-repository</url>
        <layout>default</layout>
        <releases>
            <enabled>true</enabled>
            <updatePolicy>never</updatePolicy>
        </releases>
        <snapshots>
            <enabled>false</enabled>
            <updatePolicy>never</updatePolicy>
        </snapshots>
    </pluginRepository>
</pluginRepositories>
</profile>
</profiles>
<activeProfiles>
    <!-- Activation of the Fuse Service Works profile -->
    <activeProfile>fsw-local-repos</activeProfile>
</activeProfiles>
</settings>

```

Q: Why do I still get errors when building or deploying my applications?

A: Issue

When you build or deploy a project, it fails with one or both of the following errors:

```

[ERROR] Failed to execute goal on project PROJECT_NAME
       : Could not find artifact ARTIFACT_NAME

```

Cause

Your cached local Maven repository might contain outdated artifacts.

Resolution

To resolve the issue, delete the cached local repository – the `~/.m2/repository/` directory on Linux or the `%SystemDrive%\Users\USERNAME\.m2\repository\` directory on Windows – and run `mvn clean install -U`. This will force Maven to download correct versions of required artifacts when performing the next build.

Q: Why is JBoss Developer Studio using my old Maven configuration?

A: Issue

You have updated your Maven configuration, but this configuration is not reflected in JBoss Developer Studio.

Cause

If JBoss Developer Studio is running when you modify your Maven `settings.xml` file, this configuration will not be reflected in JBoss Developer Studio.

Resolution

Refresh the Maven settings in the IDE. From the menu, choose **Window** → **Preferences**. In the **Preferences** Window, expand **Maven** and choose **User Settings**. Click the **Update Settings** button to refresh the Maven user settings in JBoss Developer Studio.

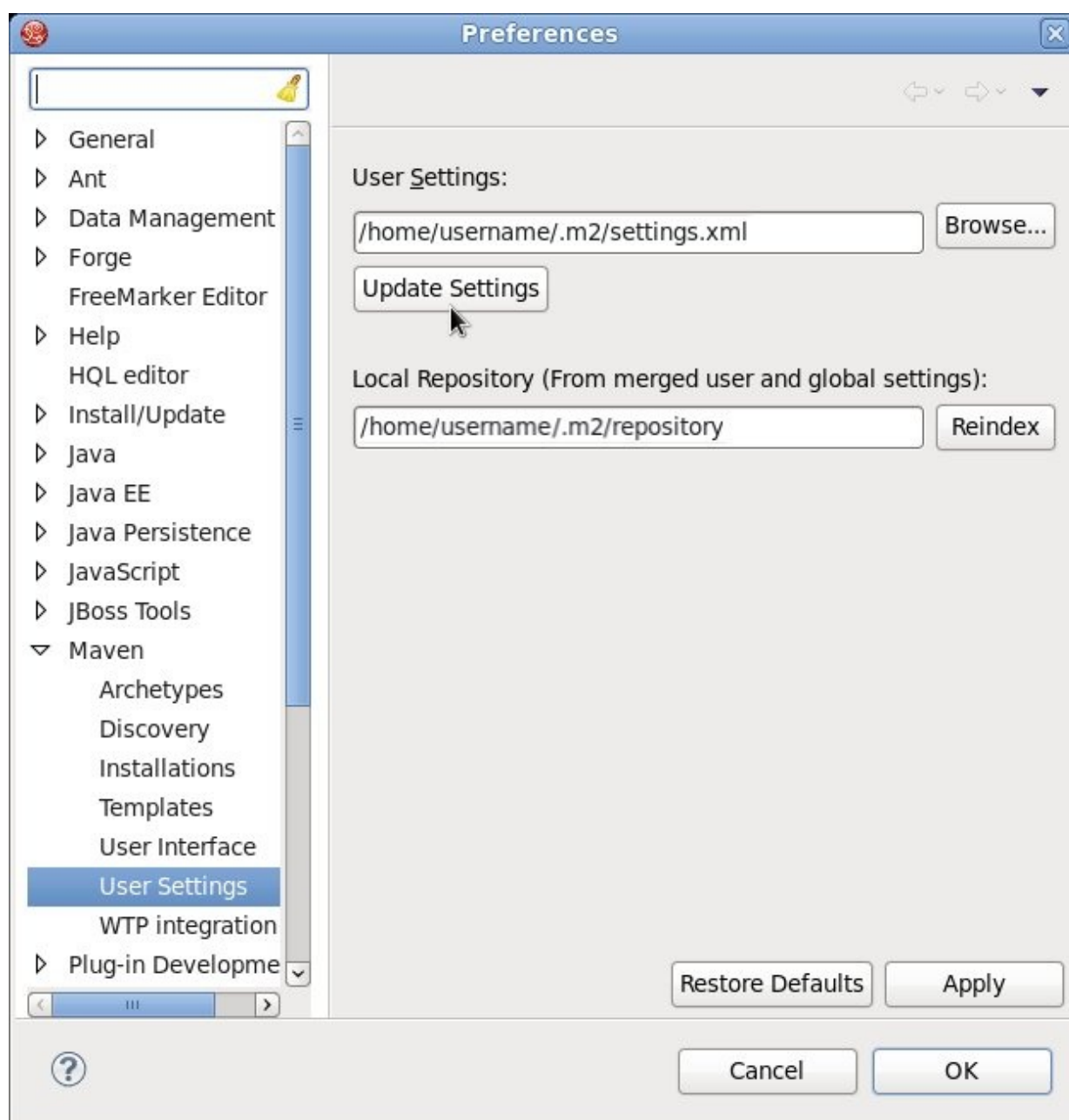


Figure 6.1. Update Maven User Settings

[Report a bug](#)

6.4. CONFIGURING MAVEN TO USE THE ONLINE REPOSITORIES

The online repositories required for Red Hat JBoss Fuse Service Works applications are located at <http://maven.repository.redhat.com/techpreview/all/> and <http://repository.jboss.org/nexus/content/repositories/public/>.

If you provided the location of Maven's `settings.xml` file during installation, Maven is already configured to use the online repositories. If you did not provide the location during installation, you need to configure Maven manually by following the procedure in [Section 6.4, "Configuring Maven to Use the Online Repositories"](#) or [Section 6.3, "Configuring Maven to Use the File System Repositories"](#).

If you did not configure the Maven repository during installation, you can configure it using the following procedure. (It is also possible to do this using the project's POM file, but this is not recommended.)

Procedure 6.2. Configuring Maven to Use the Online Repositories

1. Add entries for the online repositories to Maven's `settings.xml` file as in the code sample below:

```
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
http://maven.apache.org/xsd/settings-1.0.0.xsd">

  <profiles>
    <!-- Profile with online repositories required by Fuse Service
Works -->
    <profile>
      <id>fsw-online-repos</id>
      <repositories>
        <repository>
          <id>jboss-ga-repository</id>

          <url>http://maven.repository.redhat.com/techpreview/all</url>
          <releases>
            <enabled>>true</enabled>
          </releases>
          <snapshots>
            <enabled>>false</enabled>
          </snapshots>
        </repository>
        <repository>
          <id>jboss-public-repository</id>

          <url>http://repository.jboss.org/nexus/content/repositories/public/<
/ur1>
          <releases>
            <enabled>>true</enabled>
          </releases>
          <snapshots>
            <enabled>>false</enabled>
```



```

        </snapshots>
      </repository>
    </repositories>
    <pluginRepositories>
      <pluginRepository>
        <id>jboss-ga-plugin-repository</id>
        <url>http://maven.repository.redhat.com/techpreview/all</url>
        <releases>
          <enabled>>true</enabled>
        </releases>
        <snapshots>
          <enabled>>false</enabled>
        </snapshots>
      </pluginRepository>
      <pluginRepository>
        <id>jboss-public-plugin-repository</id>
        <url>http://repository.jboss.org/nexus/content/repositories/public/<
        /url>
        <releases>
          <enabled>>true</enabled>
        </releases>
        <snapshots>
          <enabled>>false</enabled>
        </snapshots>
      </pluginRepository>
    </pluginRepositories>
  </profile>
</profiles>

<activeProfiles>
  <!-- Activation of the Fuse Service Works profile -->
  <activeProfile>fsw-online-repos</activeProfile>
</activeProfiles>

</settings>

```

2. If you modified the **settings.xml** file while JBoss Developer Studio was running, you must refresh Maven settings in the IDE. From the menu, choose **Window** → **Preferences**. In the **Preferences** Window, expand **Maven** and choose **User Settings**. Click the **Update Settings** button to refresh the Maven user settings in JBoss Developer Studio.

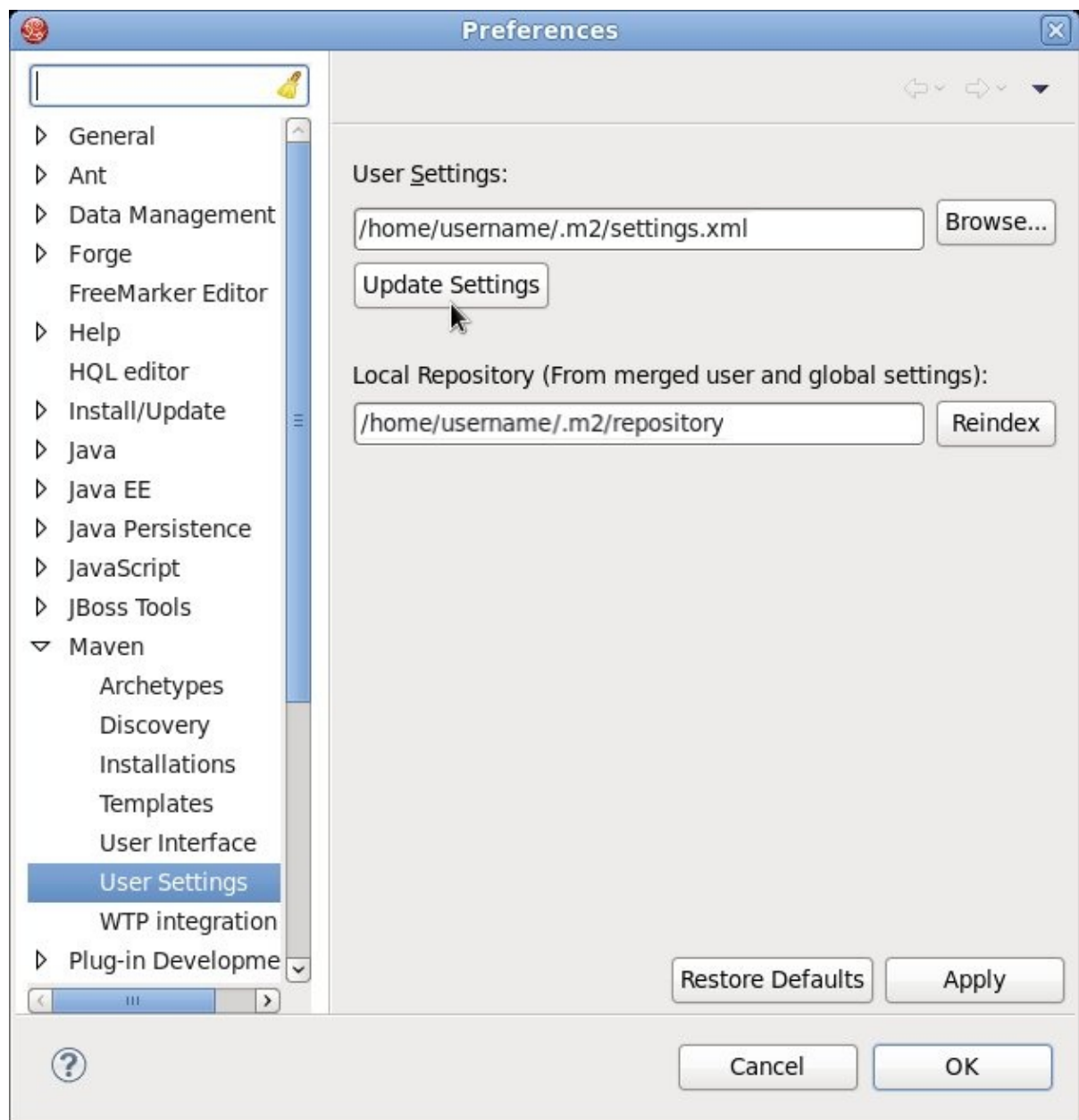
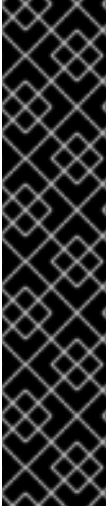


Figure 6.2. Update Maven User Settings

Result

Maven has been configured to use the online repositories provided for Red Hat JBoss Fuse Service Works.



IMPORTANT

If your cached local Maven repository contains outdated artifacts, you may encounter one of the following Maven errors when you build or deploy your project:

- Missing artifact *ARTIFACT_NAME*
- [ERROR] Failed to execute goal on project *PROJECT_NAME*; Could not resolve dependencies for *PROJECT_NAME*

To resolve the issue, delete the cached local repository – the `~/.m2/repository/` directory on Linux or the `%SystemDrive%\Users\USERNAME\.m2\repository\` directory on Windows. This will force Maven to download correct versions of required artifacts during the next build.

[Report a bug](#)

6.5. DEPENDENCY MANAGEMENT

In order to use correct Maven dependencies in your Red Hat JBoss Fuse Service Works project, relevant Bill Of Materials (BOM) and parent POM files must be added to the project's `pom.xml` file. Adding the BOM and parent will ensure that correct versions of plug-ins and transitive dependencies from the provided Maven repositories are included in the project.

To ensure correct dependency usage in your project, declare the following parent in the project's `pom.xml` file:

- `org.jboss.ip.component.management:ip-parent:1.1-redhat-5`

and add the following two BOM files as dependencies in the `dependencyManagement` section:

- `org.jboss.ip.component.management:ip-dependency-management-all:1.1-redhat-5`
- `org.jboss.component.management:jboss-dependency-management-all:6.1.2.Final-redhat-5`

Use the entries from the code sample below for this purpose.

```
<parent>
  <groupId>org.jboss.ip.component.management</groupId>
  <artifactId>ip-parent</artifactId>
  <version>1.1-redhat-5</version>
</parent>

...

<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.jboss.ip.component.management</groupId>
      <artifactId>ip-dependency-management-all</artifactId>
      <type>pom</type>
      <version>1.1-redhat-5</version>
      <scope>import</scope>
```

```
</dependency>
<dependency>
  <groupId>org.jboss.component.management</groupId>
  <artifactId>jboss-dependency-management-all</artifactId>
  <type>pom</type>
  <version>6.1.2.Final-redhat-5</version>
  <scope>import</scope>
</dependency>
</dependencies>
</dependencyManagement>
```

[Report a bug](#)

CHAPTER 7. INSTALLING JBOSS DEVELOPER STUDIO

7.1. INSTALL JBOSS DEVELOPER STUDIO

Procedure 7.1. Install with the Universal Installer

1. Log into the Customer Portal at <https://access.redhat.com>.
2. From the menu bar click **Downloads**.
3. Under **Red Hat JBoss Middleware**, click **Download Software**.



Figure 7.1. Download Red Hat JBoss Middleware Software on the Customer Portal

4. Under **Software Downloads**, select the following options:
 - From the **Product** list, select **JBoss Developer Studio**.
 - From the **Version** list, select **7.1.x**.
5. ◦ For the JBoss Developer Studio universal installer, click **Download** for the **Red Hat JBoss Developer Studio 7.1.x Stand Alone Universal Binary** download file.
 - For the JBoss Developer Studio and JBoss EAP universal installer, click **Download** for the **Red Hat JBoss Developer Studio 7.1.x Universal Binary with JBoss EAP** download file.

This downloads a universal installer **.jar** file.

6. On the command line, navigate to **path/to/.jar** and enter

```
java -jar jbdevstudio-product-universal-version.jar
```

where *version* is substituted to match the name of the **.jar** file.



NOTE

Alternatively, to start the installer you may be able to double-click the **.jar** file.

7. When the **Installer** window opens, click **Next**.
8. After reading and agreeing to the terms of the End User License Agreement, click **I accept the terms of this license agreement** and click **Next**.
9. In the **Select the installation path** field, type the path where you want JBoss Developer Studio to be installed or click **Browse** to navigate to the location. When the **Select the installation path** field shows the correct path, click **Next**. When you are prompted about the specified location being created or overwritten, review the message and, if satisfied, click **OK** or **Yes** as appropriate.

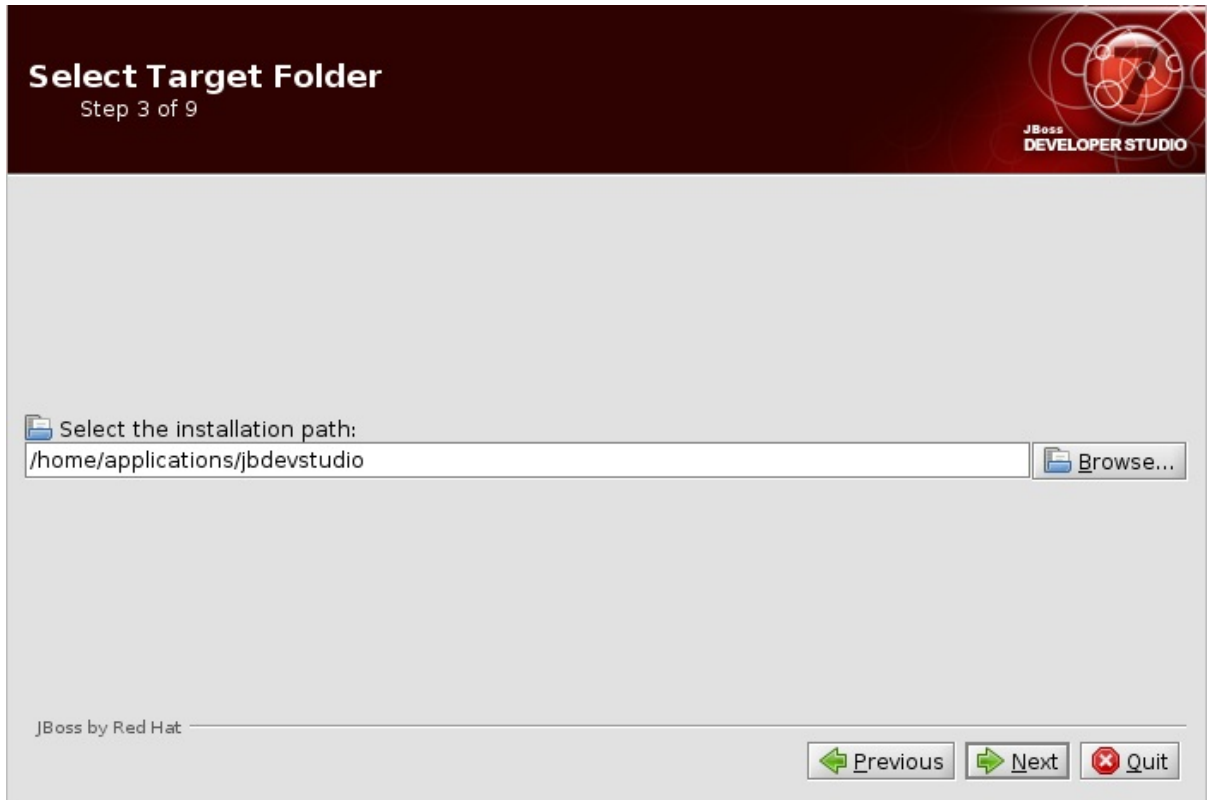


Figure 7.2. Installation Step 3: Select Target Folder

10. In the **Select Java VM** step, **Default Java VM** is automatically selected. Ensure that the disabled text field contains the path of the Java developer kit you want to use. This is based on the default Java developer kit of your system. To change the specified Java developer kit, click **Specific Java VM** and type the path of the Java developer kit in the text field or use the **Browse** button to locate the Java developer kit. When the text field shows the correct Java developer kit path, click **Next**.



IMPORTANT

You must specify a Java developer kit with a 32-bit JRE to install a 32-bit version of JBoss Developer Studio and a 64-bit JRE to install a 64-bit version of JBoss Developer Studio. To change the bit version of the Java developer kit to be used for installing JBoss Developer Studio, complete the appropriate step for your operating system:

- o On OS X operating systems, from the **Installation type** list click the appropriate bit version.
- o On Linux distributions and Microsoft Windows operating systems, in the text field type the path to the appropriate bit version of the Java developer kit.

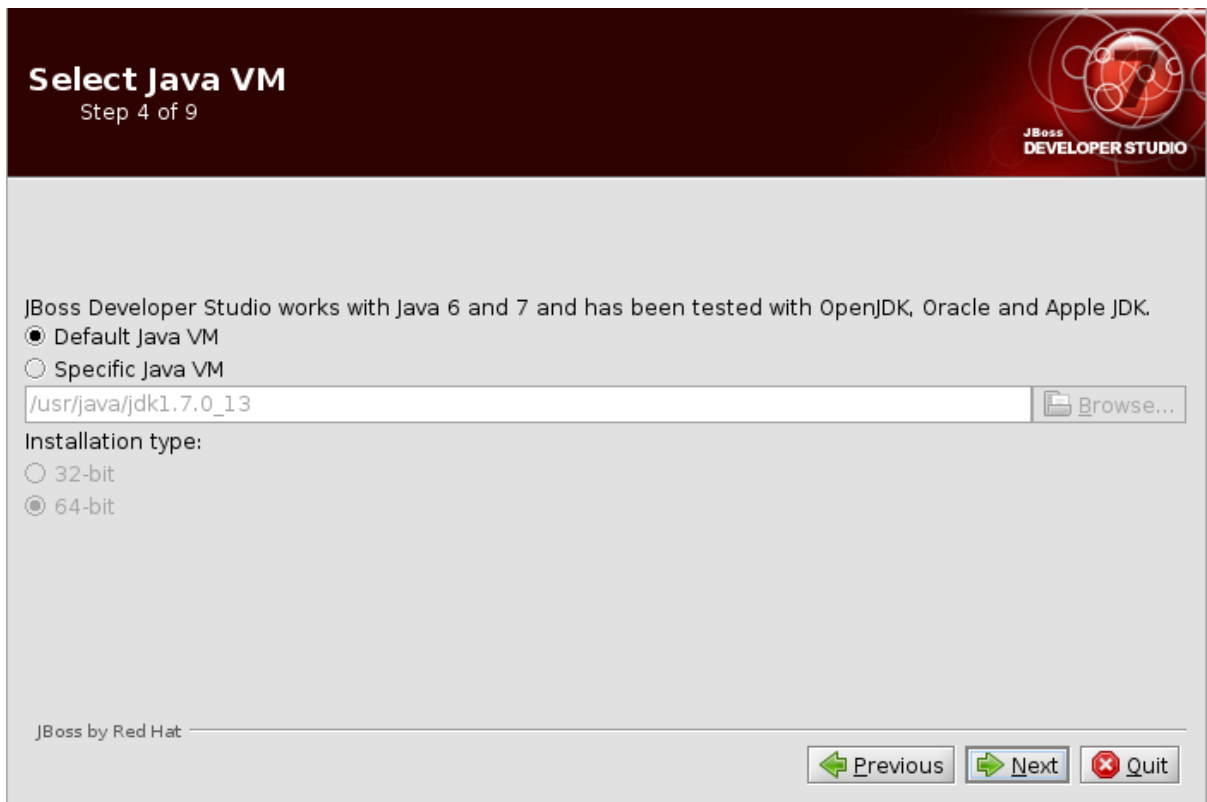


Figure 7.3. Installation Step 4: Select Java VM

11. Review the details in the **Summary Information** window and, if they are correct, click **Next**. Installation commences.
12. When the **Pack installation progress** bar shows **Finished**, click **Next**. The installation process is now complete.

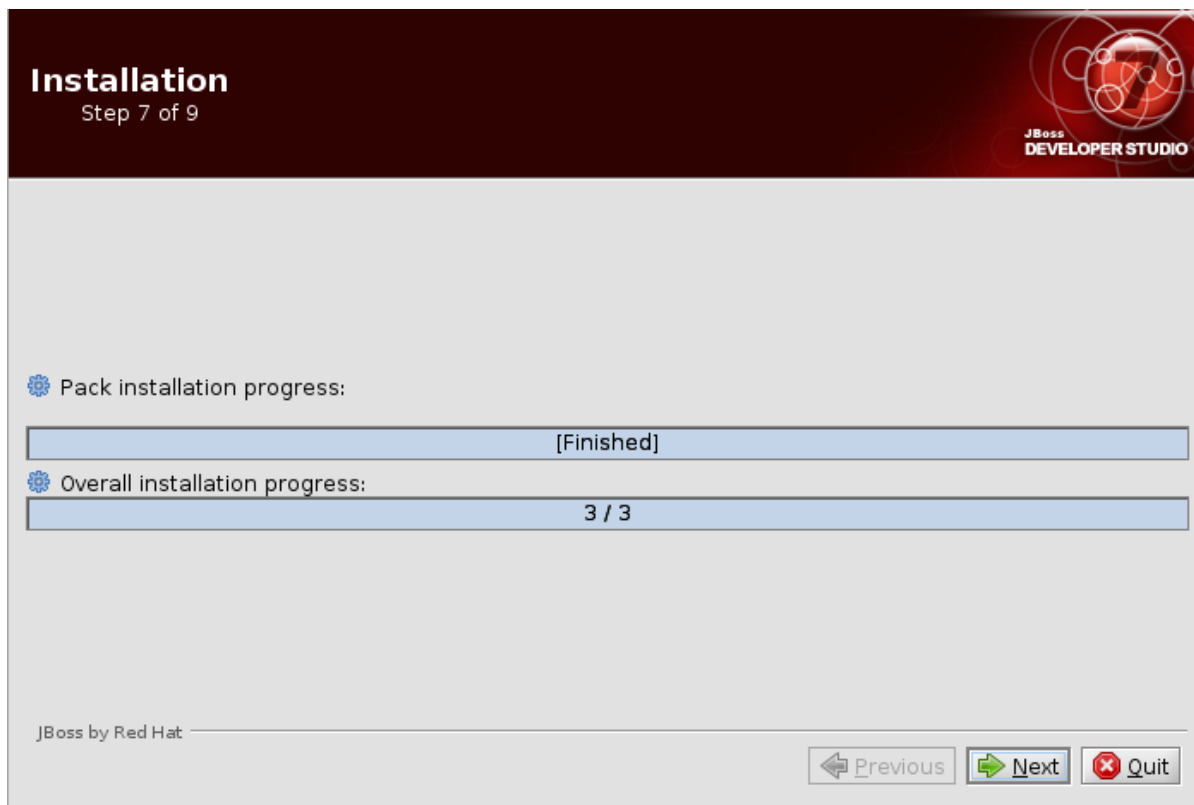


Figure 7.4. Installation Step 7: Installation Progress in Finished State

13. To create shortcuts for starting JBoss Developer Studio, select the **Create shortcuts in the Start-Menu** and **Create additional shortcut on the desktop** check boxes and click **Next**.
14. To automatically start JBoss Developer Studio when the **Installer** window closes, select the **Run JBoss Developer Studio after installation** check box. Click **Done** to close the **Installer** window.



IMPORTANT

Linux distributions have a maximum number of files that a process can have open at one time. If this maximum number of files is set too low, JBoss Developer Studio will not start. You must open the `/etc/security/limits.conf` file and ensure that the **soft nofile** and **hard nofile** variables have values of **9216** at a minimum. If the variables have smaller values, the values must be increased to **9216**. If the variables are not specified, the following lines must be added to the file:

```
* soft nofile 9216
* hard nofile 9216
```

[Report a bug](#)

7.2. INSTALLING JBOSS DEVELOPER STUDIO INTEGRATION STACK

JBoss Developer Studio Integration Stack is not packaged as part of JBoss Developer Studio installations. These plug-ins must be installed independently through JBoss Central, as detailed in the procedure below.

Procedure 7.2. Install JBoss Developer Studio Integration Stack

1. Start JBoss Developer Studio.
2. In JBoss Central, select the **Software/Update** tab. Scroll through the list to locate **JBoss Developer Studio Integration Stack**. Select the check box next to **JBoss Integration and SOA Development** and click **Install**.

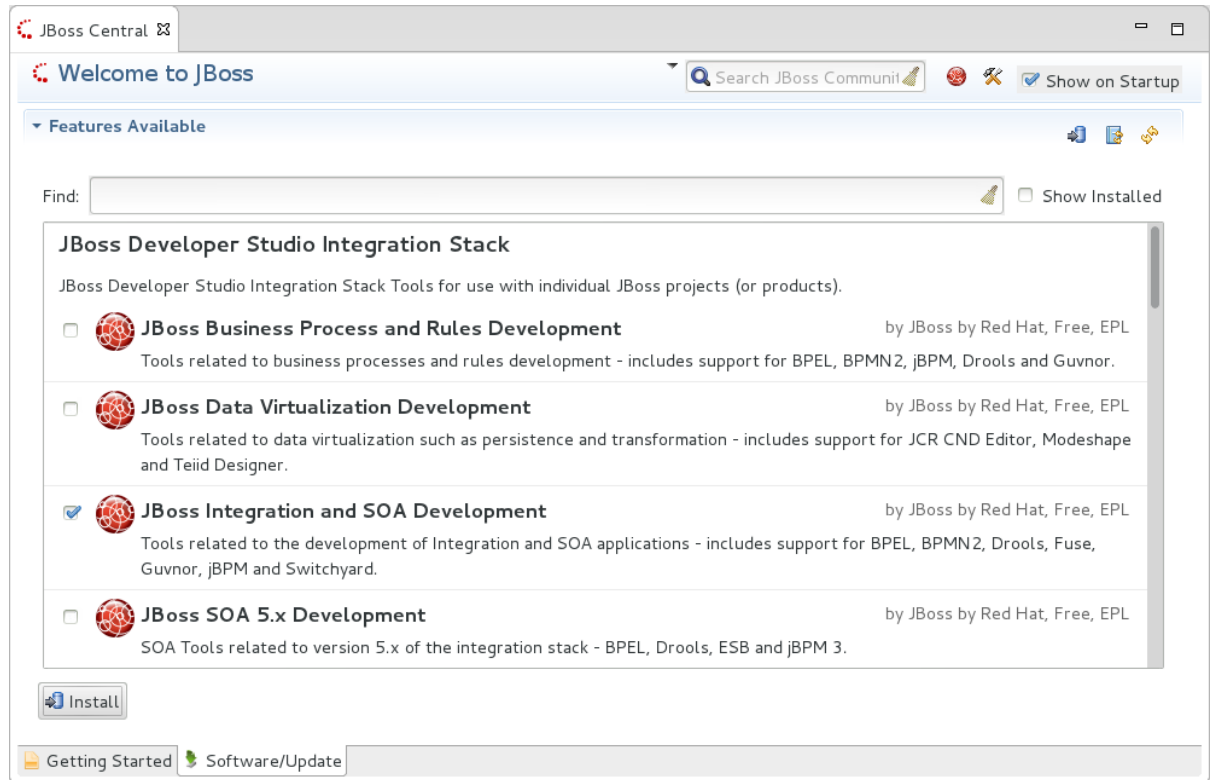


Figure 7.5. Find JBoss Developer Studio Integration Stack in JBoss Central Software/Update Tab

3. In the **Install** wizard, ensure the check boxes are selected for the software you want to install and click **Next**. It is recommended that you install all of the selected components.
4. Review the details of the items listed for install and click **Next**. After reading and agreeing to the license(s), click **I accept the terms of the license agreement(s)** and click **Finish**. The **Installing Software** window opens and reports the progress of the installation.
5. During the installation process you may receive warnings about installing unsigned content. If this is the case, check the details of the content and if satisfied click **OK** to continue with the installation.



Figure 7.6. Warning Prompt for Installing Unsigned Content

6. Once installing is complete, you are prompted to restart the IDE. Click **Yes** to restart now and **No** if you need to save any unsaved changes to open projects. Note that changes do not take effect until the IDE is restarted.

Once installed, you may need to complete additional configuration actions before you can use the individual JBoss Developer Studio Integration Stack components. For plug-in specific configuration information, see the appropriate Red Hat JBoss product documentation available from <https://access.redhat.com/site/documentation> on the Red Hat Customer Portal.



IMPORTANT

The installation method for early releases of JBoss Developer Studio Integration Stack may vary from that given here. For instructions, see the appropriate Red Hat JBoss product documentation available from <https://access.redhat.com/site/documentation> on the Red Hat Customer Portal.

[Report a bug](#)

APPENDIX A. PREREQUISITE SOFTWARE

A.1. INSTALL OPEN JDK ON RED HAT LINUX

This topic covers the steps to install Open JDK on RedHat Linux.

1. Subscribe to the Base Channel Obtain the OpenJDK from the RHN base channel. (Your installation of Red Hat Enterprise Linux is subscribed to this channel by default.)
2. Install the Package. Use the *yum* utility to install OpenJDK: **yum install java-1.7.0-openjdk-devel**
3. Verify that Open JDK is now your system default. You can ensure the correct JDK is set as the system default by following the steps below.
4.
 - a. As a root user, run the alternatives command for java: **/usr/sbin/alternatives --config java**
 - b. Select **/usr/lib/jvm /jre-1.7.0-openjdk/bin/java .**
 - c. Apply the same for javac: **/usr/sbin/alternatives --config javac**
 - d. Select **/usr/lib/jvm /java-1.7.0-openjdk/bin/java .**

Result

Open JDK is installed successfully on your machine.

[Report a bug](#)

A.2. INSTALL MAVEN

Prerequisites

The following software must be installed:

- An archiving tool for extracting the contents of compressed files.
- Open JDK.

Procedure A.1. Install Maven

1. **Download Maven.**
 - a. Enter <http://maven.apache.org/download.cgi> in the address bar of a browser.
 - b. Download **apache-maven-3.0.5** ZIP file and save it to your hard drive.
2. **Install and configure Maven.**
 - o **On Red Hat Enterprise Linux**
 - a. Extract the ZIP archive to the directory where you wish to install Maven.
 - b. Open a terminal.

- c. Add the M2_HOME environment variable by entering the following command:

```
export M2_HOME=/usr/local/apache-maven/apache-maven-3.0.x
```

- d. Add the M2 environment variable by entering the following command:

```
export M2=$M2_HOME/bin
```

- e. Add the M2 environment variable to your path by entering the following command:

```
export PATH=$M2:$PATH
```

- f. Make sure that JAVA_HOME is set to the location of your JDK. For example:

```
export JAVA_HOME=/usr/lib/jvm/java-1.7.0-openjdk.x86_64
```

- g. Make sure that \$JAVA_HOME/bin is in your PATH environment variable.

- h. Run the following command to verify that Maven is installed successfully on your machine:

```
mvn --version
```

o **On Microsoft Windows**

- a. Extract the ZIP archive to the directory where you wish to install Maven. The subdirectory **apache-maven-3.0.x** is created from the archive.

- b. Press **Start+Pause|Break**. The **System Properties** dialog box is displayed.

- c. Click the **Advanced** tab and click **Environment Variables**.

- d. Under System Variables, select **Path**.

- e. Click **Edit** and add the two Maven paths using a semicolon to separate each entry.

- Add the M2_HOME variable and set the path to **C:\Program Files\Apache Software Foundation\apache-maven-3.0.X**.
- Add the M2 variable and set the value to **%M2_HOME%\bin**.

- f. Update or create the Path environment variable:

- Add the **%M2%** variable to allow Maven to be executed from the command line.
- Add the variable **%JAVA_HOME%\bin** to set the path to the correct Java installation.

- g. Click **OK** to close all the dialog boxes including the **System Properties** dialog box.

- h. Open Windows command prompt and run the following command to verify that Maven is installed successfully on your machine:

```
mvn --version
```

-

Result

Maven is successfully installed and configured on your machine.

[Report a bug](#)

APPENDIX B. VERIFY YOUR RED HAT JBOSS FUSE SERVICE WORKS INSTALLATION

After you complete the **JBoss Fuse Service Works** installation, you can verify if the product has been installed successfully.



NOTE

The installer creates a summary of the installation in ***EAP_HOME/InstallationLog.txt***.

1. If the installer encountered any problems during installation, the installer will display an error message to the user. You can find out more information by viewing the ***EAP_HOME/InstallationLog.txt*** file.
2. If no error was reported, you can perform additional verification by performing the following steps:
 - o Start the server by running **`standalone.sh`** command
 - o Open the **`server.log`** file to check if any error messages have been logged and that SwitchYard (and other component subsystems) have loaded correctly
 - o Navigate to the Management Console (<http://localhost:9990/>) check that you can see the SwitchYard Runtime properties.
 - o Compile, deploy and run a quickstart application.

[Report a bug](#)

APPENDIX C. REVISION HISTORY

Revision 6.0.0-57	Wednesday July 22 2015	B Long
Added note about how enabling the Java Security Manager may reduce server performance..		
Revision 6.0.0-56	Friday June 5 2015	B Long
Added note about setting PATH and other variables.		
Revision 6.0.0-55	Tuesday April 14 2015	Petr Penicka
Corrected listing of A-MQ as a Fuse Service Works component, listed HornetQ instead.		
Revision 6.0.0-54	Monday February 23 2015	B Long
Version number updated from 6 to 6.0 for consistency.		
Revision 6.0.0-53	Tuesday January 13 2015	Nilam Shendye
Updated for Red Hat JBoss Fuse Service Works 6.0		
Revision 6.0.0-52	Thursday November 20 2014	Anshu Mahajan