



Red Hat JBoss BPM Suite 6.0

Installation Guide

For Red Hat JBoss Administrators

Red Hat JBoss BPM Suite 6.0 Installation Guide

For Red Hat JBoss Administrators

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 provides the steps necessary for administrators to install Red Hat JBoss Enterprise BPM Suite Platform, the plug-ins for Red Hat JBoss Developer Studio, and provides instructions for running example projects.

Table of Contents

CHAPTER 1. INTRODUCTION	3
1.1. ABOUT RED HAT JBOSS BPM SUITE	3
1.2. SUPPORTED PLATFORMS	3
1.3. USE CASE: PROCESS-BASED SOLUTIONS IN THE LOAN INDUSTRY	3
PART I. RED HAT JBOSS ENTERPRISE BPM SUITE INSTALLATION	5
CHAPTER 2. INSTALLATION OPTIONS	6
2.1. THE RED HAT JBOSS BPM SUITE INSTALLER INSTALLATION	6
2.2. THE EAP6 BUNDLE INSTALLATION	16
2.3. THE GENERIC DEPLOYABLE BUNDLE INSTALLATION	20
CHAPTER 3. SPECIAL SETUPS	30
3.1. SETTING UP PERSISTENCE FOR BUSINESS CENTRAL	30
3.2. SETTING UP PERSISTENCE FOR DASHBUILDER	31
3.3. SPECIAL SETUP FOR IBM DB2 DATABASE	31
CHAPTER 4. ROLES AND USERS	33
4.1. DEFINING ROLES	33
4.2. CREATING USERS	33
CHAPTER 5. TESTING THE INSTALLATION	35
5.1. STARTING THE SERVER	35
5.2. JAVA SECURITY MANAGER AND PERFORMANCE MANAGEMENT	36
5.3. LOGGING ON TO BUSINESS CENTRAL	36
CHAPTER 6. CLUSTERING	37
GIT REPOSITORY CLUSTERING MECHANISM	37
CLUSTERING MAVEN REPOSITORIES	38
6.1. SETTING UP A CLUSTER	38
6.2. SETTING UP QUARTZ	40
6.3. CONFIGURING CLUSTERING ON RED HAT JBOSS EAP	42
CHAPTER 7. MAVEN REPOSITORIES	46
7.1. ABOUT MAVEN	46
7.2. ABOUT THE PROVIDED MAVEN REPOSITORIES	46
7.3. CONFIGURING MAVEN TO USE THE FILE SYSTEM REPOSITORIES	46
7.4. CONFIGURING MAVEN TO USE THE ONLINE REPOSITORIES	51
7.5. DEPENDENCY MANAGEMENT	55
CHAPTER 8. RED HAT JBOSS DEVELOPER STUDIO	56
8.1. RED HAT JBOSS DEVELOPER STUDIO	56
8.2. INSTALLING THE JBOSS DEVELOPER STUDIO PLUG-INS	56
8.3. SETTING THE DROOLS RUNTIME	56
8.4. CONFIGURING THE JBOSS BPM SUITE SERVER	57
8.5. IMPORTING PROJECTS FROM A GIT REPOSITORY INTO JBOSS DEVELOPER STUDIO	58
APPENDIX A. REVISION HISTORY	62

CHAPTER 1. INTRODUCTION

1.1. ABOUT RED HAT JBOSS BPM SUITE

Red Hat JBoss BPM Suite is an open source business process management suite that combines Business Process Management and Business Rules Management and enables business and IT users to create, manage, validate, and deploy Business Processes and Rules.

Red Hat JBoss BRMS and Red Hat JBoss BPM Suite use a centralized repository where all resources are stored. This ensures consistency, transparency, and the ability to audit across the business. Business users can modify business logic and business processes without requiring assistance from IT personnel.

To accommodate Business Rules component, Red Hat JBoss BPM Suite includes integrated Red Hat JBoss BRMS.

Business Resource Planner is included as a technical preview with this release.

[Report a bug](#)

1.2. SUPPORTED PLATFORMS

Red Hat JBoss BPM Suite and Red Hat JBoss BRMS are supported on the following containers:

- Red Hat JBoss Enterprise Application Platform 6.1.1
- Red Hat JBoss Web Server 2.0 (Tomcat 7)
- IBM WebSphere Application Server 8

[Report a bug](#)

1.3. USE CASE: PROCESS-BASED SOLUTIONS IN THE LOAN INDUSTRY

Red Hat JBoss BPM Suite (BPMS) can be deployed to automate business processes, such as automating the loan approval process at a retail bank. This is a typical 'Specific Process-Based' deployment that might be the first step in a wider adoption of BPM throughout an enterprise. It leverages both the BPM and business rules features of BPMS.

A retail bank offers several types of loan products each with varying terms and eligibility requirements. Customers requiring a loan must file a loan application with the bank, which then processes the application in several steps, verifying eligibility, determining terms, checking for fraudulent activity, and determining the most appropriate loan product. Once approved, the bank creates and funds a loan account for the applicant, who can then access funds. The bank must be sure to comply with all relevant banking regulations at each step of the process, and needs to manage its loan portfolio to maximize profitability. Policies are in place to aid in decision making at each step, and those policies are actively managed to optimize outcomes for the bank.

Business analysts at the bank model the loan application processes using the BPMN2 authoring tools (Process Designer) in BPM Suite:

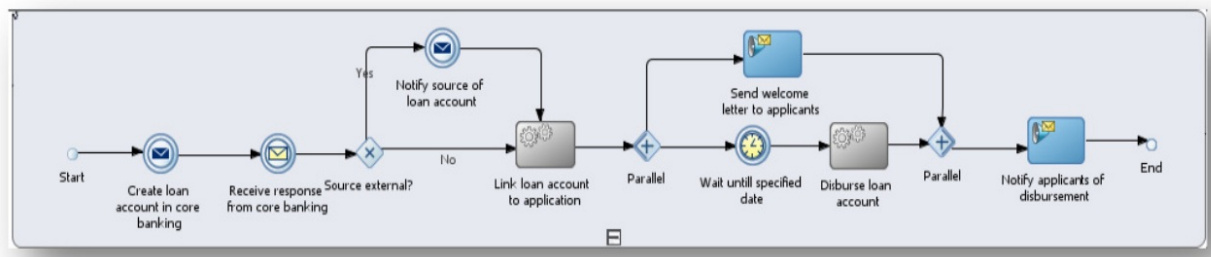


Figure 1.1. High-level loan application process flow

Business rules are developed with the rule authoring tools in BPM Suite to enforce policies and make decisions. Rules are linked with the process models to enforce the correct policies at each process step.

The bank's IT organization deploys the BPM Suite so that the entire loan application process can be automated.

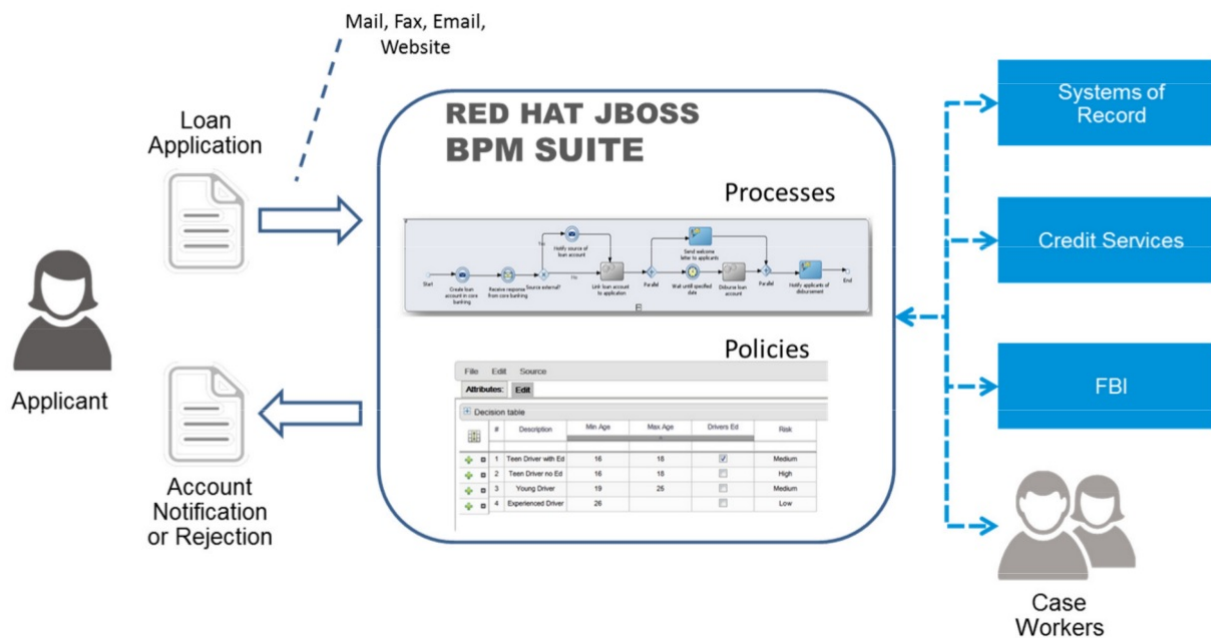


Figure 1.2. Loan Application Process Automation

The entire loan process and rules can be modified at any time by the bank's business analysts. The bank is able to maintain constant compliance with changing regulations, and is able to quickly introduce new loan products and improve loan policies in order to compete effectively and drive profitability.

[Report a bug](#)

PART I. RED HAT JBOSS ENTERPRISE BPM SUITE INSTALLATION

CHAPTER 2. INSTALLATION OPTIONS

Red Hat JBoss BPM Suite comes in two versions:

- Executable jar installer for installation on Red Hat JBoss Enterprise Application Platform (EAP) 6.1.1.
- Zip file install which itself comes in two versions:
 - **jboss-bpms-6.MINOR_VERSION-redhat-x-deployable-eap6.x.zip**: version adapted for deployment on Red Hat JBoss Enterprise Application Platform (EAP 6.1.1).
 - **jboss-bpms-6.MINOR_VERSION-redhat-x-deployable-generic.zip**: the deployable version with additional libraries adapted for deployment on Red Hat JBoss Web Server (WS), Apache Tomcat 6, and Apache Tomcat 7.

Depending on your environment, you may choose the installation option best suited for your project needs.



NOTE

Red Hat JBoss BPM Suite is designed to work with UTF-8 encoding. If a different encoding system is used by the underlying JVM, unexpected errors might occur. To ensure UTF-8 is used by the JVM, use the following system property "-Dfile.encoding=UTF-8".

[Report a bug](#)

2.1. THE RED HAT JBOSS BPM SUITE INSTALLER INSTALLATION

This section describes the steps required to install Red Hat JBoss BPM Suite using the jar file installer installation method. The jar file is an executable file that installs EAP6 if you do not already have it, or you can use it to install BPMS on an existing EAP6 installation.



NOTE

Due to IBM JDK not being able to use keystores generated on other JDKs, it is not possible to install JBoss BPM Suite into an existing JBoss EAP running on IBM JDK with a keystore generated on another JDK.

[Report a bug](#)

2.1.1. Downloading Red Hat JBoss BPM Suite Installer

1. Go to the [Red Hat Customer Portal](#) and log in.
2. Click **Downloads** → **Products Downloads**.
3. In the **Product Downloads** page that opens, click **Red Hat JBoss BPM Suite**.
4. From the **Version** drop-down menu, select version 6.0.3.

5. Select **Red Hat JBoss BPM Suite 6.0.3 Installer for EAP 6.1.1** and then click **Download**.

[Report a bug](#)

2.1.2. Installing Red Hat JBoss BPM Suite using the Installer

The installer for Red Hat JBoss BPM Suite is an executable Java jar file. You can use it to install BPMS on an existing EAP 6.1.1 installation or a brand new one. The installer gives you the option to install EAP 6.1.1 if you do not already have it.



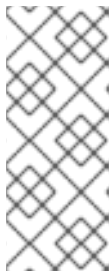
NOTE

For security reasons, you should run the installer as a **non-root user**.

1. Setup Location and Users

Navigate to the folder where you downloaded the installer file in a command prompt and execute the following command (replace the VERSION number and x with the actual file name).

```
java -jar jboss-bpms-installer-VERSION.GA-redhat-x.jar
```



NOTE

When running the installer on Windows, you may be prompted to provide administrator credentials during the installation. To prevent this, add the **izpack.mode=privileged** option to the installation command: **java - Dizpack.mode=privileged -jar jboss-bpms-installer-VERSION.GA-redhat-x.jar**

2. The graphical installer will execute and display a splash screen and a license agreement page. Accept the license to proceed.
3. In the next screen, provide the parent location of an existing EAP where BPMS needs to be installed. If you instead prefer to install BPMS in a brand new EAP, then point this location to a parent directory where there is no existing EAP. BPMS will then be installed on top of a fresh EAP server bundled with the installer. The screenshot below depicts an example directory path:

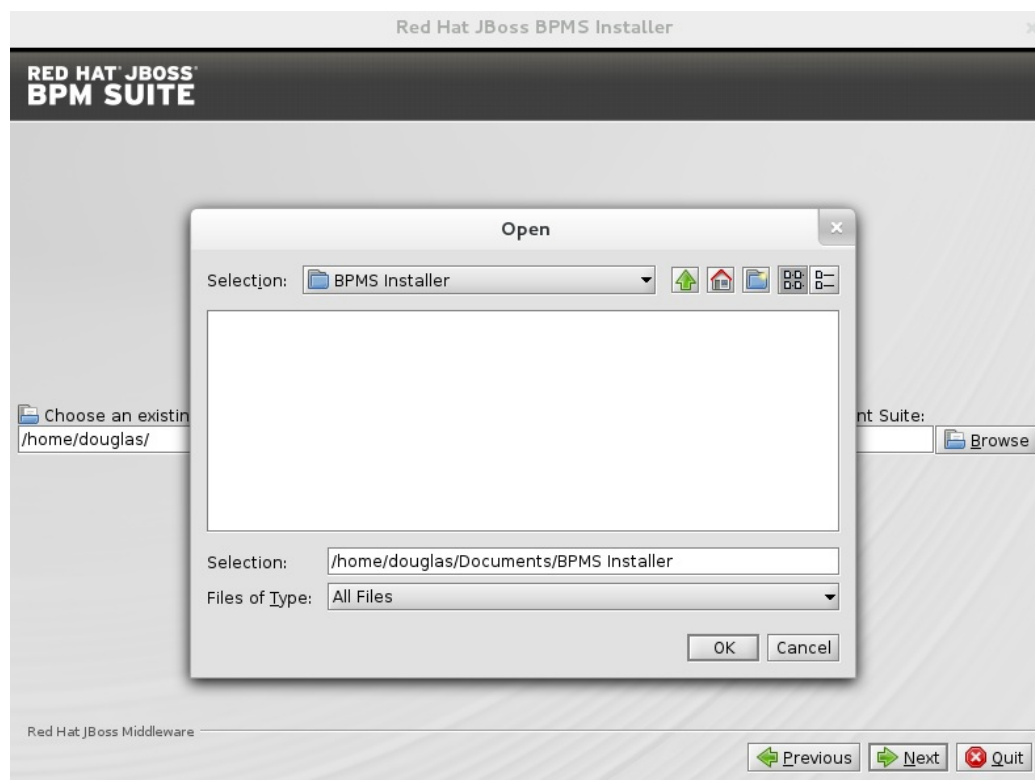


Figure 2.1. BPM Suite Installer Directory Path

4. In the next two screens, create two users: the first one for the management console of the EAP (ManagementRealm) and the second one for managing BPMS itself (ApplicationRealm). Make a note of these usernames and passwords as you will need them to access the EAP server and the BPMS application respectively.



NOTE

The passwords that you create must have at least 8 characters and must contain atleast one number and one alphanumeric character.

Depicted below is a screenshot of the username and password page:

Red Hat JBoss BPMS Installer

RED HAT JBOSS BPM SUITE

Create a Business Process Management Suite User

Create a BPM Suite user. The user will be added to the ApplicationRealm, and can be used to access the Business Central Console. The User will be assigned the 'analyst, developer, user, manager, admin' application roles.

The password must have at least 8 characters, and contain at least one number and one non-alphanumeric symbol.

BPMS username:

BPMS password:

Confirm BPMS password:

Red Hat JBoss Middleware

Previous Next Quit

Figure 2.2. BPM Suite Installer Username and Password Screen



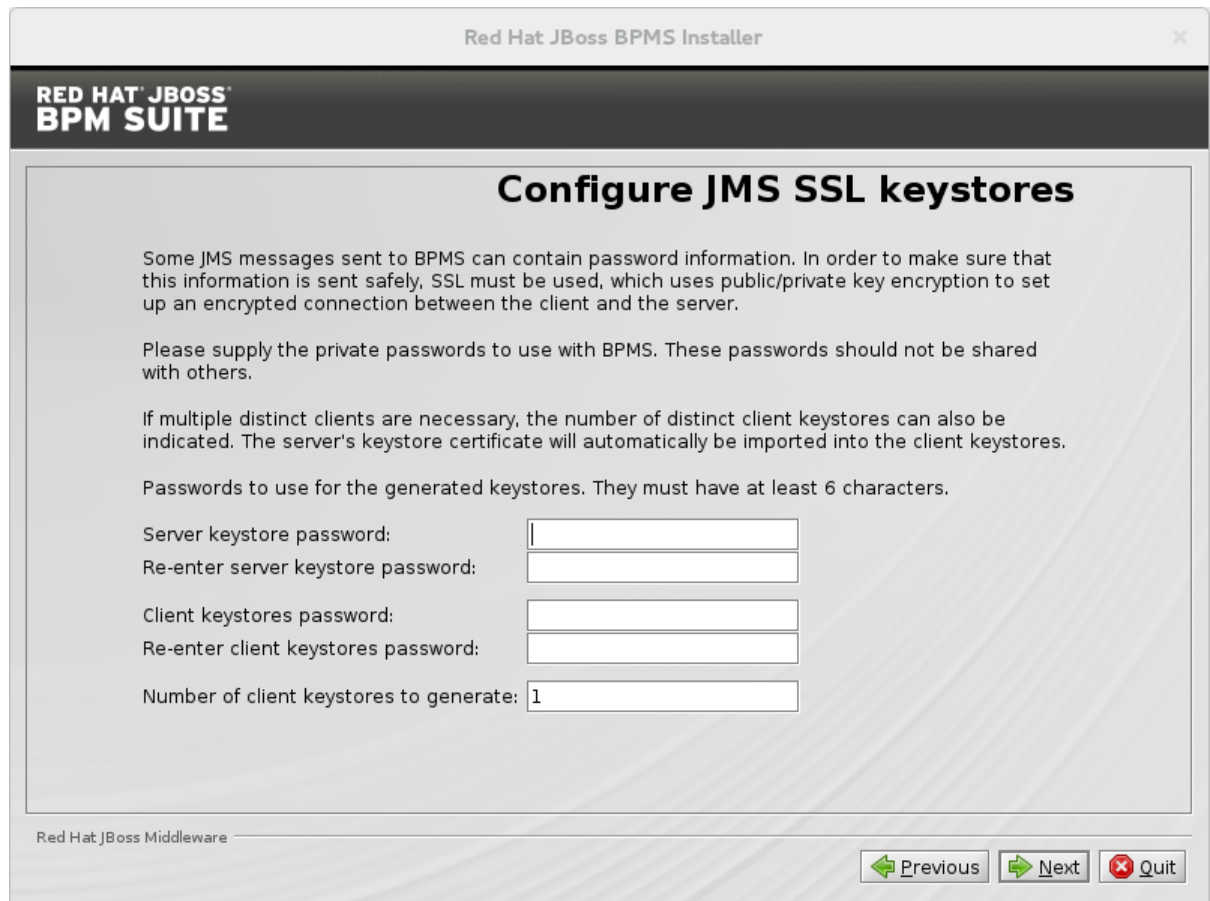
NOTE

The application role assigned to the second user that you create is the admin role. This is the only role that can be assigned to this newly created user. You can create more users with narrow roles afterwards by using the command line.

5. Setup Security Environment

Next, you will setup the security environment of your new BPMS install. Decide to enable or disable the Java Security Manager in this step by clicking on the checkbox. The Java Security Manager makes your system more secure but may downgrade performance. You need to make a decision based on your environment.

- Next, configure the passwords for the SSL Keystores for both server and client side for JMS messages. This is to make sure that the messages sent to BPMS via JMS can be encrypted as it may contain sensitive information.



7. Final Configuration Steps

Choose whether you want to setup pure IPv6 configuration on the server that the installation is taking place. This will allow you to setup runtime IPv6 specific configurations later.

8. Choose default configuration for the runtime environment in the next step and click next to review the installation details. If you are happy with the details, click next to start the actual installation or click previous to go back and make changes.
9. The installer will go through the steps to install BPMS and will perform post installation configuration steps when you click next. The installer will also start the BPMS server and connect to it to validate the installation. Click next to get to the last screen where you can generate the installation script and properties file. Click done to quit the installer.

You have successfully installed Red Hat JBoss BPM Suite using the installer.

[Report a bug](#)

2.1.3. Installing Red Hat JBoss BPM Suite using the Installer in CLI Mode

The installer for Red Hat JBoss BPMS can also be executed through the command-line interface (CLI). The procedure below demonstrates the steps that you are likely to encounter using this option to install BPMS.

1. Navigate to the folder where you downloaded the installer file in a command prompt and execute the following command (replace the VERSION number and x with the actual file name).

```
java -jar jboss-bpms-installer-VERSION.GA-redhat-x.jar -console
```

2. The command-line interactive process will start and display the End-User license agreement. You will be prompted to select an option at the end of this license:

```
press 1 to continue, 2 to quit, 3 to redisplay.
```

3. Enter 1 to begin the installation and type in the parent directory of an existing EAP installation. If you specify a directory where EAP does not exist, the installer will install a new version of EAP for you.

```
If you wish to install on an existing EAP installation, the parent
directory of that EAP installation should be specified. [/home/user]
```

If you decide to create a new folder for the installation to take place, you will be prompted to select the following:

```
press 1 to continue, 2 to quit, 3 to redisplay.
```

4. Create a user for the management console of EAP (Management Realm):

```
Create an administrative user
This user will be added to the host container's management realm for
administrative purposes. It can be used to access the management
console, the management CLI or other applications secured in this
realm.
```

```
Admin username: [admin]
```

5. Create and confirm a password for the user of the EAP management console:

```
The password must have at least 8 characters, and contain at least
one number and one non-alphanumeric symbol.
```

```
Admin password: []
```

```
Confirm admin password: [*****]
```

6. After the passwords have been entered, choose an option from the prompt below:

```
press 1 to continue, 2 to quit, 3 to redisplay.
```

7. Enter 1 then create a BPMS user:

```
Create a Business Process Management Suite User
Create a BPM Suite user. The user will be added to the
ApplicationRealm, and can be used to access the Business Central
Console. The User will be assigned the 'admin' application roles.
The BPMS username cannot be any of the following: 'admin',
'analyst', 'user', 'manager' or 'developer'.
```

```
BPMS username: [bpmsAdmin]
```

8. Create and confirm a password for the BPMS user:

```
The password must have at least 8 characters, and contain at least
one number and one non-alphanumeric symbol.
```

```
BPMS password: []
```

```
Confirm BPMS password: [*****]
```

9. After the passwords have been entered, choose an option from the prompt below:

```
press 1 to continue, 2 to quit, 3 to redisplay.
```

10. Configure the Java Security Manager by either pressing 1 to select it or 0 to deselect it.

```
Configure the Java Security Manager
```

```
A Java security manager offers JVM level security beyond what is
provided by the application container. It enforces access rules at
the JVM runtime based on one or more security policies.
```

```
This installer will place two security policies in the installation
directory with the filenames 'security.policy' and 'kie.policy'
regardless of choice. Those policies will be enabled at runtime if
the option below is selected.
```

```
Please note that a security manager imposes a significant
performance overhead when enabled. It is suggested the included
policies be applied in production if user requirements call for a
stronger measure than what is already provided by the application
container's authentication and authorization mechanism.
```

```
Please see the JBoss Business Process Management Suite
administrative documentation for further details and consideration.
```

```
[x] Enable the Java security manager
```

```
Input 1 to select, 0 to deselect:
```

11. After the Java Security Manager choice, choose an option from the prompt below:

```
press 1 to continue, 2 to quit, 3 to redisplay.
```

12. The Configure JMS SSL keystores will display on the CLI:

```
Configure JMS SSL keystores
```

```
Some JMS messages sent to BPMS can contain password information. In
order to make sure that this information is sent safely, SSL must be
used, which uses public/private key encryption to set up an
encrypted connection between the client and the server.
```


Please supply the private passwords to use with BPMS. These passwords should not be shared with others.

If multiple distinct clients are necessary, the number of distinct client keystores can also be indicated. The server's keystore certificate will automatically be imported into the client keystores.

Passwords to use for the generated keystores. They must have at least 6 characters.

13. Enter the server keystore password:

Server keystore password: []

Re-enter server keystore password: [*****]

14. Enter the client keystores password:

Client keystores password: [*****]

Re-enter client keystores password: [*****]

15. Enter the number of client keystores to generate:

Number of client keystores to generate: [1]

16. After the keystores configuration, choose an option from the prompt below:

press 1 to continue, 2 to quit, 3 to redisplay.

17. Next, select whether to enable the IPv6 configuration.

IPv6 configuration

If this computer is using a pure IPv6 configuration, please check the box below. A pure IPv6 setup requires additional configuration at runtime to ensure the proper bindings of the management and http interfaces.

[] Enable pure IPv6 configuration
Input 1 to select, 0 to deselect:

18. After the IPv6 configuration choice, choose an option from the prompt below:

press 1 to continue, 2 to quit, 3 to redisplay.

19. Configure the runtime environment by either choosing the default configuration or inputting advanced options.

```
Configure runtime environment
Red Hat JBoss Business Process Management Suite can be further
customized at this time.
0  [x] Perform default configuration
1  [ ] Perform advanced configuration
Input Selection:
```

If you select 1, "Perform advanced configuration," complete the following configurations:

```
[ ] Install password vault
Input 1 to select, 0 to deselect:

[ ] Enable SSL security
Input 1 to select, 0 to deselect:

[ ] Enable LDAP authentication
Input 1 to select, 0 to deselect:

[ ] Add a security-domain
Input 1 to select, 0 to deselect:
```

20. Next, choose an option from the prompt below:

```
press 1 to continue, 2 to quit, 3 to redisplay.
```

21. Configure the password vault by creating a password and pressing 1 to continue:

```
Configure password vault
A password vault encrypts sensitive strings and stores them in an
encrypted keystore. The vault mechanism manages decrypting the
strings for use with security domains, security realms, or other
verification systems.

Please make note of your entry below in order to mask any subsequent
passwords. See EAP 6 documentation for further details.

The password must have no fewer than 6 characters.

Vault keystore password: []

Re-enter vault keystore password: [*****]

press 1 to continue, 2 to quit, 3 to redisplay.
```

22. The .jar file will begin to unpack and configure.

23. After a successful installation, the command-line will ask you if you would like to generate an automatic installation script and properties file.

```
Installation has completed successfully.
Application installed on /home/user/Documents/BPMS_Installer
Would you like to generate an automatic intallation script and
properties file?
(y/n) [n]:
```

24. If you select [y], provide a path for the automatic installation script:

```
Select path for the automatic installation script:
[/home/user/Documents/BPMS_Installer/<auto script filename>]
```

This generated script will allow the user to run the installer in the following way for future installations:

```
java -jar jboss-bpms-installer.jar <auto script filename>
```

NOTE

Running the installer in this way will result in an installation identical to the installation from which the auto script was generated. Note that sensitive values, such as passwords, will need to be provided from an external file or provided at auto installation time. The optional argument below allows the user to provide these values automatically:

```
-variablefile <variable filename>
```

Sensitive values can also be provided using the following argument:

```
-variables key1=value1,key2=value2
```

25. The command-line will provide the following message upon a successful auto script creation and/or console installation:

```
XML written successfully.
[ Console installation done ]
[BPMS_Installer]$
```

26. Start JBoss EAP by running **standalone.sh** in the **jboss-eap-[VERSION]/bin** directory.

```
./standalone.sh
```

27. Navigate to <http://localhost:8080/business-central> in a web browser.

28. Login with the correct username/password as given to the BPMS user in the "Create and confirm a password for the BPMS user" step.

2.2. THE EAP6 BUNDLE INSTALLATION

This section describes installing the Red Hat JBoss BPM Suite package deployable for Red Hat JBoss Enterprise Application Platform (EAP).



NOTE

The minimum supported configuration of Red Hat JBoss EAP for Red Hat JBoss BPM Suite installation is 6.1.1 and not 6.1.0.

[Report a bug](#)

2.2.1. Downloading the EAP6 package

To download the deployable Red Hat JBoss BPM Suite package for JBoss Enterprise Application Platform, do the following:

1. Go to the [Red Hat Customer Portal](#) and log in.
2. Click **Downloads** → **Products Downloads**.
3. In the **Product Downloads** page that opens, click **Red Hat JBoss BPM Suite**.
4. From the **Version** drop-down menu, select version 6.0.3.
5. On the **Software Downloads** page that opens, navigate to the **Red Hat JBoss BPM Suite 6.0.3 Deployable for EAP 6.1.1** row and click **Download**.

[Report a bug](#)

2.2.2. Installing the EAP6 package

Installation on a fresh EAP instance

To install the deployable package for an EAP that has yet to be configured, do the following:

1. Extract the zip package deployable for EAP you downloaded from the [Red Hat Customer Portal](#).
2. Merge the extracted zip package deployable for EAP into the EAP `SERVER_HOME` directory.



WARNING

This step must be performed by the same user account that was used to install EAP. This account must not be a superuser account.

3. In this process, multiple files in the EAP `SERVER_HOME` directory will be overwritten and you must let the unzip process overwrite these files. An example of a file that is overwritten is the `SERVER_HOME/bin/product.conf` file. After a successful merge, this file must contain the string `slot=bpms`. You can open this file to verify that the files have been overwritten successfully.

In Red Hat Enterprise Linux, you can use the following command to extract the BPMS zip file and merge it into your server directory correctly in one step, if you execute this command in the directory where you have downloaded the zip file:

```
unzip -u jboss-bpms-VERSION-TYPE.zip -d SERVER_HOME_PARENT_DIR
```

Example 2.1. The unzip command

```
unzip -u jboss-bpms-6.0.3-redhat-7-deployable-eap6.x.zip -d  
/home/john/myServers/
```

On server start-up, Red Hat JBoss BPM Suite will be deployed.

Installation on an existing EAP configuration



WARNING

These instructions are for installing, and *NOT* for updating an existing BPMS instance. Make sure that there is no existing BPMS install in the target EAP.

To install the deployable package for a previously configured EAP, do the following:

1. Extract the zip package deployable for EAP you downloaded from the [Red Hat Customer Portal](#).
2. Unzip the downloaded zip archive; however, do not overwrite all of the files. Manually merge the following files into the *SERVER_HOME* directory.
 - **jboss-eap-6.1/domain/configuration/*** - (please be aware that BPMS requires JMS, so JMS is added by default into all profiles in **domain.xml** provided by BPMS distribution.)
 - **jboss-eap-6.1/standalone/configuration/*** - (please be aware that BPMS requires JMS, so JMS is added by default into all profiles config files (especially into **standalone.xml** and **standalone-ha.xml**) provided by BPMS distribution.)
 - **jboss-eap-6.1/modules/layers.conf**
 - **jboss-eap-6.1/bin/product.conf**
3. Ensure the target EAP does not include a deployment with a colliding name. Copy the folder **jboss-eap-6.1/standalone/deployments** into the *EAP_HOME* directory from the BPMS distribution.
4. Make sure no EAP module layer is already called BPMS and copy the folder **jboss-eap-6.1/modules/system/layers/bpms** into the EAP 6.1.1 folder.

[Report a bug](#)

2.2.3. Installing BPM Suite in the Domain Mode

To install the deployable package for an EAP in the domain mode, do the following steps:

1. Extract the downloaded EAP Deployable ZIP file from [Red Hat Customer Portal](#) and copy the following directories into your local installation of EAP 6.1:

- **bin**
- **domain**
- **modules**

Skip the **standalone** directory.

2. On the command line, move to the **/domain** directory and start the domain:

In a Unix environment, run:

```
./domain.sh
```

In a Windows environment, run:

```
./domain.bat
```

3. Deploy the archive either via **\${jboss-eap-home}/bin/jboss-cli.sh** / **\${jboss-eap-home}/bin/jboss-cli.bat**, or via management web UI (localhost:9990/):



NOTE

The web applications `business-central.war` and `dashbuilder.war` supplied in the EAP deployable binaries are directories, but for deployment into the domain, you have to use WAR archives. To create them, simply zip the content of the `business-central.war` and the `dashbuilder.war` directories.

- a. To deploy the archive via **\${jboss-eap-home}/bin/jboss-cli.sh** or **\${jboss-eap-home}/bin/jboss-cli.bat**, move into the **\${jboss-eap-home}/bin** directory and deploy the WAR file:

In a Unix environment, run:

```
./jboss-cli.sh
```

In a Windows environment, run:

```
./jboss-cli.bat
```

- b. To deploy the archive via management web UI (localhost:9990/):

- log in using your EAP management account
- select **Domain** -> **Manage Deployments** -> **Content Repository** -> **Add**

- select the web archive from the file system, upload the web archive
- select the deployment, click the **Assign** button
- select the server group



NOTE

In order to log in to Business Central deployed on Host Controller (HC) machines, the user created on the Domain Controller Machine has to be created on the Host Controller machines as well, by following the steps in the [Section 4.2, “Creating users”](#) section.

Installing Multiple BPMS Server Instances

In many situations, users may want to group together a set of EAP 6 nodes on the same machine and give them a meaningful name for easy maintenance. Unique values need to be incorporated for the system properties for each server instance. Listed below are the common properties that can be specified with a single BPMS node to change the default configuration; however, they should be specified for multiple nodes running on a single machine so every node can point to a different directory:

- **org.uberfire.nio.git.dir**
- **org.uberfire.metadata.index.dir**
- **org.uberfire.nio.git.ssh.cert.dir**

When multiple BPMS nodes are used on a single machine, the below properties need to be specified:

- **org.uberfire.nio.git.daemon.host** - can be left on default to bind to localhost.
- **org.uberfire.nio.git.daemon.port**
- **org.uberfire.nio.git.ssh.host** - can be left on default to bind to localhost.
- **org.uberfire.nio.git.ssh.port**



NOTE

Both the **org.uberfire.nio.git.daemon.port** and the **org.uberfire.nio.git.ssh.port** require different port values in order to avoid port conflicts.

Incorporate the previous properties in the **\$EAP_HOME/domain/configuration/host.xml** file as illustrated in the two nodes below:

Node A:

```
<system-properties>
  <property name="org.uberfire.nio.git.dir" value="/valid/path/.." boot-
time="false"/>
  <property name="org.uberfire.metadata.index.dir"
value="/valid/path/.." boot-time="false"/>
  <property name="org.uberfire.nio.git.ssh.cert.dir"
value="/valid/path/.." boot-time="false"/>
```

```

    <property name="org.uberfire.nio.git.daemon.host" value="10.10.10.10"
boot-time="false"/>
    <property name="org.uberfire.nio.git.daemon.port" value="9417" boot-
time="false"/>
    <property name="org.uberfire.nio.git.ssh.host" value="10.10.10.10"
boot-time="false"/>
    <property name="org.uberfire.nio.git.ssh.port" value="8002" boot-
time="false"/>
</system-properties>

```

Node B:

```

<system-properties>
    <property name="org.uberfire.nio.git.dir" value="/valid/path/.." boot-
time="false"/>
    <property name="org.uberfire.metadata.index.dir"
value="/valid/path/.." boot-time="false"/>
    <property name="org.uberfire.nio.git.ssh.cert.dir"
value="/valid/path/.." boot-time="false"/>

    <property name="org.uberfire.nio.git.daemon.host" value="10.10.10.10"
boot-time="false"/>
    <property name="org.uberfire.nio.git.daemon.port" value="9418" boot-
time="false"/>
    <property name="org.uberfire.nio.git.ssh.host" value="10.10.10.10"
boot-time="false"/>
    <property name="org.uberfire.nio.git.ssh.port" value="8003" boot-
time="false"/>
</system-properties>

```

The system properties depicted above should indicate the host, port, or location of the **.index** or **.nio.git** files. These files, which should be used by a respective node, would then be grouped in a particular domain.

[Report a bug](#)

2.3. THE GENERIC DEPLOYABLE BUNDLE INSTALLATION

To install Red Hat JBoss BPM Suite on Red Hat JBoss Web Server (WS) , you need to use the generic deployable package of the product.

For installation on WS, the generic deployable package contains additional transaction manager and security libraries that are not part of Red Hat JBoss WS.

Note that the generic deployable package contains the following zip archives:

- **jboss-bpms-engine.zip**: supported execution engine libraries needed if you are embedding the engine into your application
- **jboss-bpms-manager.zip**: the **business-central.war** and **dashbuilder.war** web applications

[Report a bug](#)

2.3.1. Downloading the generic deployable package

To download the generic deployable Red Hat JBoss BPM Suite package for JBoss Web Server, do the following:

1. Go to the [Red Hat Customer Portal](#) and log in.
2. Select **Downloads** → **Product Downloads**.
3. From the list of products click on **Red Hat JBoss BPM Suite**.
4. From the **Version** drop-down menu, select version 6.0.3.
5. In the Software Downloads section that comes up, navigate to the **Red Hat JBoss BPM Suite 6.0.3 Deployable for all supported containers** row and then click **Download**.

[Report a bug](#)

2.3.2. Installing the generic deployable package

To install the generic deployable package, you need to set up the following after you have installed the underlying platform (Red Hat JBoss WS):

- Set up the database driver and the transaction manager (Bitronix) (refer to [Section 2.3.2.1, “Setting up transaction manager for Red Hat JBoss Web Server 2.0 \(Tomcat 7\)”](#)).
- Set up the Business Central application: set up users and roles and set up persistence (refer to [Section 2.3.2.2, “Setting up Business Central for Red Hat JBoss Web Server 2.0 \(Tomcat 7\)”](#)).
- Set up the Dashbuilder application: set up users and roles and set up persistence (refer to [Section 2.3.2.3, “Setting up Dashbuilder for Red Hat JBoss Web Server 2.0 \(Tomcat 7\)”](#)).

[Report a bug](#)

2.3.2.1. Setting up transaction manager for Red Hat JBoss Web Server 2.0 (Tomcat 7)

1. Extract the generic deployable zip package you downloaded from [Red Hat Customer Portal](#). This zip package contains two more zip files: **jboss-bpms-engine.zip** and **jboss-bpms-manager.zip**.
2. Extract the contents of the **jboss-bpms-manager.zip** file to a temporary location. This zip file contains two web application archive folders: **business-central.war** and **dashbuilder.war** in exploded formats and these are now in your temporary location. Rename these folders to remove the **.war** extension.

Copy both these folders directly under the **\$TOMCAT_DIR/webapps** folder.

You should end up with two folders in exploded format: **\$TOMCAT_DIR/webapps/business-central** and **\$TOMCAT_DIR/webapps/dashbuilder**.



NOTE

\$TOMCAT_DIR stands for the home directory where your web server is located. Replace it with the actual path to your web server home directory, for example: **/home/john/jboss-ews-2.0/tomcat7/**

3. Extract the **jboss-bpms-engine** folder from the **jboss-bpms-engine.zip** archive to a temporary location from where you can copy the required libraries. This folder now contains all the core BPMS libraries under the extracted folder and a **lib** folder.
4. **Install the transaction manager.**

**WARNING**

Please note that the following section describes the setup of a transaction manager, Bitronix that is not officially supported by Red Hat.

Copy the following transaction manager jar libraries from the **lib** folder where you just extracted the **jboss-bpms-engine** libraries to **\$TOMCAT_DIR/lib/** directory:

- `btm-VERSION.jar`
- `btm-tomcat55-lifecycle-VERSION.jar`
- `jta-VERSION.jar`
- `slf4j-api-VERSION.jar`
- `slf4j-ext-VERSION.jar`

In addition, download the following library and copy it into the **\$TOMCAT_DIR/lib/** folder as well:

- [javax.security.jacc-api.jar](#)

5. Install the driver to your database: copy the jar file with the relevant database driver to **\$TOMCAT_DIR/lib/**.

**NOTE**

If using the embedded H2 database, the driver is available in **business-central/WEB-INF/lib/**.

6. Create the transaction manager configuration files in **\$TOMCAT_DIR/conf/**:

- `btm-config.properties`

```
bitronix.tm.serverId=tomcat-btm-node0
bitronix.tm.journal.disk.logPart1Filename=${btm.root}/work/btm1.t
log
bitronix.tm.journal.disk.logPart2Filename=${btm.root}/work/btm2.t
log
bitronix.tm.resource.configuration=${btm.root}/conf/resources.pro
perties
```

- resources.properties (the resource.ds1.uniqueName defines the datasource name used in tomcat resource definition later - make a note of this value).

Make sure to change the values in the following definitions to match your environment.

Example 2.2. H2 datasource definition

```
resource.ds1.className=bitronix.tm.resource.jdbc.lrc.LrcXADataSource
resource.ds1.uniqueName=jdbc/jbpm
resource.ds1.minPoolSize=10
resource.ds1.maxPoolSize=20
resource.ds1.driverProperties.driverClassName=org.h2.Driver
resource.ds1.driverProperties.url=jdbc:h2:file:~/jbpm
resource.ds1.driverProperties.user=sa
resource.ds1.driverProperties.password=
resource.ds1.allowLocalTransactions=true
```

Example 2.3. MySQL 5.5 datasource definition

```
resource.ds1.className=com.mysql.jdbc.jdbc2.optional.MysqlXADataSource
resource.ds1.uniqueName=jdbc/jbpm
resource.ds1.minPoolSize=0
resource.ds1.maxPoolSize=10
resource.ds1.driverProperties.URL=jdbc:mysql://localhost:3306/sampled
resource.ds1.driverProperties.user=dbuser
resource.ds1.driverProperties.password=dbpassword
resource.ds1.allowLocalTransactions=true
```

Example 2.4. DB2 Type 4 datasource definition

```
resource.ds1.className=com.ibm.db2.jcc.DB2Driver
resource.ds1.uniqueName=jdbc/jbpm
resource.ds1.minPoolSize=0
resource.ds1.maxPoolSize=10
resource.ds1.driverProperties.URL=jdbc:db2://localhost:50000/sampled
resource.ds1.driverProperties.user=dbuser
resource.ds1.driverProperties.password=dbpassword
resource.ds1.allowLocalTransactions=true
```

Example 2.5. Oracle datasource definition

```
resource.ds1.className=oracle.jdbc.xa.client.OracleXADataSource
resource.ds1.uniqueName=jdbc/jbpm
resource.ds1.minPoolSize=0
resource.ds1.maxPoolSize=10
resource.ds1.driverProperties.URL=jdbc:oracle:thin:@//localhost
```

```
:1521/bpms
resource.ds1.driverProperties.user=dbuser
resource.ds1.driverProperties.password=dbpassword
resource.ds1.allowLocalTransactions=true
```

Example 2.6. Microsoft SQL Server datasource definition

```
resource.ds1.className=com.microsoft.sqlserver.jdbc.SQLServerDr
iver
resource.ds1.uniqueName=jdbc/jbpm
resource.ds1.minPoolSize=0
resource.ds1.maxPoolSize=10
resource.ds1.driverProperties.URL=jdbc:sqlserver://localhost:14
33;databaseName=bpms;
resource.ds1.driverProperties.user=dbuser
resource.ds1.driverProperties.password=dbpassword
resource.ds1.allowLocalTransactions=true
```

7. Set up the transaction manager listener in **\$TOMCAT_DIR/conf/server.xml** to start and stop Bitronix on container startup and shutdown:

Add the following element as the last <Listener> element into the <Server> element:

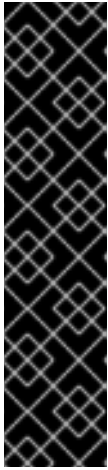
```
<Listener
  className="bitronix.tm.integration.tomcat55.BTMLifecycleListener" />
```

8. Define the btm.root system property and location where bitronix config file is placed:

In **\$TOMCAT_DIR/bin/**, create the **setenv.sh** file with the following content:

```
CATALINA_OPTS="-Xmx512M -XX:MaxPermSize=512m -
Dbtm.root=$CATALINA_HOME -
Dbitronix.tm.configuration=$CATALINA_HOME/conf/btm-config.properties
-Dorg.jbpm.designer.perspective=RuleFlow -
Djbpm.tsr.jndi.lookup=java:comp/env/TransactionSynchronizationRegist
ry"
```

The property **org.jbpm.designer.perspective** is set to **RuleFlow** to allow the default perspective for the designer to be **RuleFlow** rather than **Full**. Grant the file execute permissions if applicable.



IMPORTANT

On Microsoft Windows systems, replace the **\$CATALINA_HOME** value in the content of the file with the equivalent environment variable name or use the absolute path and add the values in **setenv.bat** file as shown here in the following example:

```
set "CATALINA_OPTS=-Xmx512M -XX:MaxPermSize=512m -
Dbtm.root=C:/Tomcat -
Dbitronix.tm.configuration=C:/Tomcat/conf/btm-
config.properties -
Dorg.jbpm.designer.perspective=RuleFlow"
```

[Report a bug](#)

2.3.2.2. Setting up Business Central for Red Hat JBoss Web Server 2.0 (Tomcat 7)

To set up Business Central, do the following:

1. Set up a **Valve** so that the Business Central web application can load the users set up in Tomcat:
 - a. Define users and roles in **\$TOMCAT_DIR/conf/tomcat-users.xml**. Note that Business Central requires users to have the roles specified as **admin** and/or **analyst** (for more information about user and role definitions, refer to the Tomcat 7 documentation).

The program listing below shows an example of how these two roles would be added and how a user named **bpmsadmin** will be assigned these roles.

```
<role rolename="admin"/>
<role rolename="analyst" />
<user username="bpmsadmin" password="P@ssw0rd"
roles="admin,analyst"/>
```

- b. Copy **kie-tomcat-integration-VERSION.jar** from **\$TOMCAT_DIR/webapps/business-central/WEB-INF/lib/** to **\$TOMCAT_DIR/lib/**.
- c. Copy **jaxb-api-VERSION.jar** from **\$TOMCAT_DIR/webapps/business-central/WEB-INF/lib/** to **\$TOMCAT_DIR/lib/**.
- d. In **\$TOMCAT_DIR/conf/server.xml**, add the Tomcat Valve declaration in the relevant **<host>** element:

```
<Valve className="org.kie.integration.tomcat.JACCValve" />
```

- e. In **\$TOMCAT_DIR/webapps/business-central/WEB-INF/web.xml**, uncomment entries marked with the **TOMCAT-JEE-SECURITY** comments.
- f. Set up the tomcat authentication source: in the **\$TOMCAT_DIR/webapps/business-central/WEB-INF/classes/META-INF/services/** directory, rename the following files: **org.uberfire.security.auth.AuthenticationSource** to

org.uberfire.security.auth.AuthenticationSource-ORIGIN and **org.uberfire.security.auth.AuthenticationSource-TOMCAT-JEE-SECURITY** to **org.uberfire.security.auth.AuthenticationSource**:

```
# Example command if you run this from $TOMCAT_DIR/webapps
directory
$ mv business-central/WEB-INF/classes/META-
INF/services/org.uberfire.security.auth.AuthenticationSource
business-central/WEB-INF/classes/META-
INF/services/org.uberfire.security.auth.AuthenticationSource-
ORIGIN
$ mv business-central/WEB-INF/classes/META-
INF/services/org.uberfire.security.auth.AuthenticationSource-
TOMCAT-JEE-SECURITY business-central/WEB-INF/classes/META-
INF/services/org.uberfire.security.auth.AuthenticationSource
```

- g. In **\$TOMCAT_DIR/webapps/business-central/WEB-INF/beans.xml**, uncomment **JAASUserGroupInfoProducer** and comment **org.jbpm.kie.services.cdi.producer.DefaultUserGroupInfoProducer** (optional). The **alternatives** part of this file should now look like this:

```
<alternatives>
  <!--
<class>org.jbpm.kie.services.cdi.producer.DefaultUserGroupInfoPro
ducer</class> -->
  <!-- uncomment JAASUserGroupInfoProducer when using JEE
security on Tomcat -->

<class>org.jbpm.kie.services.cdi.producer.JAASUserGroupInfoProduc
er</class>
</alternatives>
```

2. If you are using a datasource other than the default provided by the underlying H2 database, you will need to setup persistence. If you are using the default H2 database, then you can ignore the rest of the steps in this procedure.

In this procedure, you configure a datasource with the JNDI name **jdbc/myDatasource** as defined in `uniqueName=jdbc/jbpm` in the bitronix **resources.properties** file earlier (for the MySQL option).

- a. In **business-central/META-INF/context.xml**, replace the datasource JNDI name in the `<Resource>` element. The `uniqueName` attribute refers to the `resource.ds1.uniqueName` property set in **resources.properties**:

```
<Resource name="jdbc/myDatasource" uniqueName="jdbc/jbpm"
auth="Container" removeAbandoned="true"
factory="bitronix.tm.resource.ResourceObjectFactory"
type="javax.sql.DataSource"/>
```

- b. In **business-central/WEB-INF/web.xml**, replace the datasource JNDI name in the `<res-ref-name>` element with your datasource name:

```
<resource-ref>
```

```

        <description>Console DS</description>
        <res-ref-name>jdbc/myDatasource</res-ref-name>
        <res-type>javax.sql.DataSource</res-type>
        <res-auth>Container</res-auth>
    </resource-ref>

```

- c. Change **business-central/WEB-INF/classes/META-INF/persistence.xml**.

In this file, change the name of the hibernate dialect used for your database, if using a different database other than H2. The code below demonstrates the original database information for **persistence.xml**:

```

<property name="hibernate.dialect"
value="org.hibernate.dialect.H2Dialect"/>

```

This information can be updated in the following manner (as demonstrated with MySQL database below):

```

<property name="hibernate.dialect"
value="org.hibernate.dialect.MySQLDialect"/>

```



NOTE

The dialect for DB2 is **org.hibernate.dialect.DB2Dialect**, for DB2 on AS/400 is **org.hibernate.dialect.DB2400Dialect**, for Oracle is **org.hibernate.dialect.Oracle10gDialect** and for Microsoft SQL Server is **org.hibernate.dialect.SQLServerDialect**

- d. Change **business-central/WEB-INF/classes/META-INF/persistence.xml** file so that BPMS process engine can use the new database.

The code below demonstrates the original datasource information for **persistence.xml**:

```

<jta-data-source>java:comp/env/jdbc/jbpm</jta-data-source>

```

Change this value to the datasource defined earlier:

```

<jta-data-source>java:comp/env/jdbc/myDatasource</jta-data-
source>

```

3. You can now start the JBoss Web Server to login to Business Central.

- a. Run **startup.sh** in the **\$TOMCAT_HOME/bin** directory.

```

./startup.sh

```

- b. Navigate to <http://localhost:8080/business-central> in a web browser.

- c. Login with the correct username/password as given in the **tomcat-users.xml** file where you defined user roles.

2.3.2.3. Setting up Dashbuilder for Red Hat JBoss Web Server 2.0 (Tomcat 7)



NOTE

Before setting up Dashbuilder on Red Hat JBoss Web Server, you must ensure that you have correctly installed and started Business Central as described in [Section 2.3.2.2, “Setting up Business Central for Red Hat JBoss Web Server 2.0 \(Tomcat 7\)”](#). Dashbuilder requires the history log database tables to exist, which are only provided by Business Central. If these tables are not present in the database before attempting the steps below, you may get initialization errors.

To set up Dashbuilder on Red Hat JBoss Web Server, do the following:

1. Define users and roles in **`$TOMCAT_DIR/conf/tomcat-users.xml`**. Note that Dashbuilder requires users to have the role specified as **admin** and/or **analyst**. If you have already defined these users earlier for Business-Central, you don't need to define them again.
2. Enable single sign-on between Dashbuilder and Business Central by uncommenting the following lines in **`$TOMCAT_DIR/conf/server.xml`** file:

```
<Valve className="org.apache.catalina.authenticator.SingleSignOn" />
```

3. As with Business Central setup, if you are using a database other than the default and integrated H2 database, you will need to setup persistence.

In this procedure, you configure a datasource with the JNDI name **jdbc/dashbuilderDS** as defined in `uniqueName=jdbc/jbpm` in the bitronix **resources.properties** file:

- a. In **`dashbuilder/META-INF/context.xml`**, replace the datasource JNDI name in the `<Resource>` element. The `uniqueName` attribute refers to the `resource.ds1.uniqueName` property set in **resources.properties**:

```
<Resource name="jdbc/dashbuilderDS" uniqueName="jdbc/jbpm"
auth="Container" removeAbandoned="true"
factory="bitronix.tm.resource.ResourceObjectFactory"
type="javax.sql.DataSource"/>
```



NOTE

Depending upon your database, you may need to define some other properties here as well. For example, in an Oracle environment, this entry may look like the following listing.

```
<Resource name="jdbc/jbpm" uniqueName="jdbc/jbpm"
auth="Container" removeAbandoned="true"
factory="bitronix.tm.resource.ResourceObjectFactory"
type="javax.sql.DataSource" username="username"
password="password"
driverClassName="oracle.jdbc.xa.client.OracleXADataSource"
url="jdbc:oracle:thin:YOUR-URL:1521:YOUR-DB"
maxActive="8" />
```


- b. In **dashbuilder/WEB-INF/web.xml**, add the datasource JNDI name in the `<res-ref-name>` element with your datasource name:

```
<resource-ref>
  <description>Dashboard Builder Datasource</description>
  <res-ref-name>jdbc/dashbuilderDS</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

- c. In **dashbuilder/META-INF/context.xml**, define the transaction factory:

```
<Transaction
factory="bitronix.tm.BitronixUserTransactionObjectFactory"/>
```

- d. Update the datasource JNDI name in **dashbuilder/WEB-INF/etc/hibernate.cfg.xml** in the `<session-factory>` element:

```
<property
name="connection.datasource">java:/comp/env/jdbc/dashbuilderDS</p
roperty>
```

4. Restart Java Web server for these changes to take effect. Once restarted, you can navigate to Dashbuilder from within Business Central or directly via:
http://localhost:8080/dashbuilder.

[Report a bug](#)

CHAPTER 3. SPECIAL SETUPS

3.1. SETTING UP PERSISTENCE FOR BUSINESS CENTRAL

BPM Suite is configured to use an example datasource with Java Naming and Directory Interface (JNDI) name `java:jboss/datasources/ExampleDS`. For Business Central, this example datasource is located in the file `business-central.war/WEB-INF/classes/META-INF/persistence.xml`.

If you want to configure BPM Suite to use an external database, make the following changes. For file `business-central.war/WEB-INF/classes/META-INF/persistence.xml`:

1. Install the respective Java Database Connectivity (JDBC) driver using modular approach for easy subsequent configuration (see [EAP 6 documentation](#)).
2. Create a new datasource according to the example in [EAP 6 documentation, section 6.7.1. Example PostgreSQL Datasource](#). This is the default used H2 database specific datasource configuration:

```
<subsystem xmlns="urn:jboss:domain:datasources:1.1">
  <datasources>
    <datasource jndi-name="java:jboss/datasources/ExampleDS"
pool-name="ExampleDS" enabled="true" use-java-context="true">
      <connection-url>jdbc:h2:mem:test;DB_CLOSE_DELAY=-
1</connection-url>
      <driver>h2</driver>
      <security>
        <user-name>sa</user-name>
        <password>sa</password>
      </security>
    </datasource>
    <drivers>
      <driver name="h2" module="com.h2database.h2">
        <xa-datasource-
class>org.h2.jdbcx.JdbcDataSource</xa-datasource-class>
      </driver>
    </drivers>
  </datasources>
```

3. Use the JNDI name of the datasource to update the following entry inside the `persistence.xml` file, which is by default set to this entry.

```
<jta-data-source>java:jboss/datasources/ExampleDS</jta-data-source>
```



IMPORTANT

When configuring your datasource, make sure to enable JTA (typically, by adding `jta="true"` to the `datasource` tag).

4. Replace the following text with the appropriate database specific hibernate dialect name.

```
<property name="hibernate.dialect"
value="org.hibernate.dialect.H2Dialect" />
```

For example, for an Oracle Database Express Edition 11g, it would be:

```
<property name="hibernate.dialect"
value="org.hibernate.dialect.Oracle10gDialect" />
```

[Report a bug](#)

3.2. SETTING UP PERSISTENCE FOR DASHBUILDER

As Dashbuilder is dependent on the configuration of Business Central, ensure Business Central is configured according to [Section 3.1, “Setting up Persistence for Business Central”](#). BPMS 6 is configured to use a datasource with Java Naming and Directory Interface (JNDI) name **java:jboss/datasources/ExampleDS**. If you want to make the application work with a database different from **H2**, for example **Oracle**, **MySQL**, **Postgres**, or **MS SQL Server**, follow these steps.

1. Install the database driver and create a new datasource according to the example in [EAP 6 Documentation in section 6.7.1. Example PostgreSQL Datasource](#). Use modular approach to the installation of JDBC driver for easy subsequent configuration.
2. Create an empty database.
3. Modify file **dashbuilder.war/WEB-INF/jboss-web.xml** whose default entry is:

```
<jboss-web>
<context-root>/dashbuilder</context-root>
<resource-ref>
  <res-ref-name>jdbc/dashbuilder</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <jndi-name>java:jboss/datasources/ExampleDS</jndi-name>
</resource-ref>
```

4. Modify also files **WEB-INF/jboss-deployment-structure.xml** from both the **business-central.war** and **dashbuilder.war** applications, and add a dependency under the dependencies section on the JDBC driver module created earlier during datasource creation. The following snippet shows a sample configuration where **jdbcDriverModuleName** is the name of the JBoss EAP 6 JDBC driver module.

```
<dependencies>
  ...
  <module name="jdbcDriverModuleName" />
  ...
</dependencies>
```

[Report a bug](#)

3.3. SPECIAL SETUP FOR IBM DB2 DATABASE

If you want to use an IBM DB2 database as the underlying data source for Business Central, you will need to increase the page size for the database. The default page size of 4 kB is not sufficient for the Dashbuilder table columns size.

When creating the database, force the page size to 16384 as in the example below:

Example 3.1. Adjusting page size

```
CREATE DATABASE dashb PAGESIZE 16384
```

This increase in page size for the underlying database must be performed before the BPM Suite has been run for the first time.

[Report a bug](#)

CHAPTER 4. ROLES AND USERS

4.1. DEFINING ROLES

Before starting the server and logging onto Business Central, you will need to create some user accounts. This section describes the different user roles that are used in Red Hat JBoss BPM Suite :

- **admin:** The users with admin role are the administrators of the application. Administrators can manage users, manage the repositories (create and clone) and have full access to make the required changes in the application. Admins have access to all areas within the system.
- **developer:** A developer has access to almost all features and can manage rules, models, process flows, forms and dashboards. They can manage the asset repository, they can create, build and deploy projects and they can even use Red Hat JBoss Developer Studio to view processes. Only certain administrative functions like creating and cloning a new repository are hidden for the developer role.
- **analyst:** An analyst role has access to all high-level features to model and execute their projects. However, **Authoring** → **Administration** access is unavailable to users with the analyst role. Certain lower-level features targeted towards developers, like the **Deployment** → **Artifact Repository** view are not accessible for this role. However, the **Build & Deploy** button is available for the analyst role while using the Project Editor.
- **user:** User or a business user work on the business task lists that are used to operate a certain process. A user with this role can access the dashboard and manage processes.
- **manager:** A manager is a viewer of the system and is interested in statistics around the business processes and their performance, business indicators, and other reporting of the system. A user with this role has access to the BAM only.



NOTE

Enter the above mentioned roles during the user creation process.

[Report a bug](#)

4.2. CREATING USERS

To start adding new users, you will need to run the `add-user.sh` script on a Unix system or the `add-user.bat` file on a Windows system from the EAP bin directory.

1. Run `./add-user.sh` on a Unix system or `add-user.bat` on a Windows system from the bin directory.
2. Enter `b` to select an Application User at the type of user prompt and press Enter.
3. Accept the default Realm (ApplicationRealm): by pressing Enter.
4. At the username prompt, enter a user name and confirm. For example: **helloworlduser**.
5. Create the user's password at the password prompt and reenter the password. For example: **HelloWorld@123**.

**NOTE**

The password should be at least 8 characters in length and should contain upper and lower case alphabetic characters (e.g. A-Z, a-z), at least one numerical character (e.g. 0-9) and at least one special character (e.g. ~ ! @ # \$ % ^ * () - _ + =).

6. Enter a comma separate list of roles the user will need at the roles prompt (refer to [Section 4.1, “Defining Roles”](#)).

Business Central users need to have at least the **analyst** role, and dashbuilder users need to have the **admin** role. Roles should be entered as a comma-separated list.

7. Confirm you want to add the user.
8. Enter yes at the next prompt (this is to enable clustering in the future if required).

[Report a bug](#)

CHAPTER 5. TESTING THE INSTALLATION

5.1. STARTING THE SERVER

If you have installed BPMS using either the Installer or via the EAP6 bundle install, you can now start your server in one of two modes.



NOTE

If you installed BPMS using the generic deployable version on Red Hat Java Web Server, the instructions for download and install also contain the instructions for starting the server. You can ignore the following discussion.

The default startup script, **standalone.sh** that Red Hat JBoss BPM Suite ships with is optimized for performance. To run your server in the performance mode, do the following:

1. On the command line, move into the **`$SERVER_HOME/bin/`** directory.
2. In a Unix environment run:

```
./standalone.sh
```

In a Windows environment run:

```
./standalone.bat
```

Red Hat JBoss BPM Suite also ships with a separate script, **standalone-secure.sh** that is optimized for security. This script applies a security policy by default that protects against a known security vulnerability.



NOTE

It is recommended that production environments use **standalone-secure.sh** script.



WARNING

The use of a security manager imposes a significant performance penalty that you should be aware of. The tradeoff between security and performance must be made by taking into consideration individual circumstances. See [Section 5.2, “Java Security Manager and performance management”](#).

To run your server in the secure mode with this script, do the following:

1. On the command line, move into the **`$SERVER_HOME/bin/`** directory.
2. In a Unix environment run:

```
./standalone-secure.sh
```

In a Windows environment run:

```
./standalone-secure.bat
```

[Report a bug](#)

5.2. JAVA SECURITY MANAGER AND PERFORMANCE MANAGEMENT

As noted earlier, enabling the Java Security Manager (JSM) to sandbox the evaluation of MVEL scripts in BPMS introduces a performance hit in high load environments. Environments and performance markers must be kept in mind when deploying a BPMS application. Use the following guidelines to deploy secure and high performance BPMS applications.

- In high load environments where performance is critical it is recommended to only deploy applications that have been developed on other systems and properly reviewed. It is also recommended not to create any users with Analyst role on such systems. If these safeguards are followed, it is safe to leave JSM disabled on these systems so it does not introduce any performance degradation.
- In testing and development environments without high loads, or in environments where rule and process authoring is exposed to external networks, it is recommended to have JSM enabled in order to achieve security benefits of properly sandboxed evaluation of MVEL.

Allowing users with Analyst role to log in to the Business Central console with JSM disabled is not secure and not recommended.

[Report a bug](#)

5.3. LOGGING ON TO BUSINESS CENTRAL

Log into Business Central after the server has successfully started.

1. Navigate to <http://localhost:8080/business-central> in a web browser. If the user interface has been configured to run from a domain name, substitute **localhost** for the domain name. For example <http://www.example.com:8080/business-central>.
2. Log in with the user credentials that were created during installation. For example: User = **helloworlduser** and password = **Helloworld@123**.

[Report a bug](#)

CHAPTER 6. CLUSTERING

When clustering Red Hat JBoss BPM Suite, consider which components need to be clustered. You can cluster the following:

- **GIT repository:** virtual-file-system (VFS) repository that holds the business assets so that all cluster nodes use the same repository
- **Execution Server and Web applications:** the runtime server that resides in the container (such as, Red Hat JBoss EAP) along with BRMS and BPM Suite web applications so that nodes share the same runtime data.

For instructions on clustering the application, refer to the container clustering documentation.

- **Back-end database:** database with the state data, such as, process instances, KIE sessions, history log, etc., for fail-over purposes

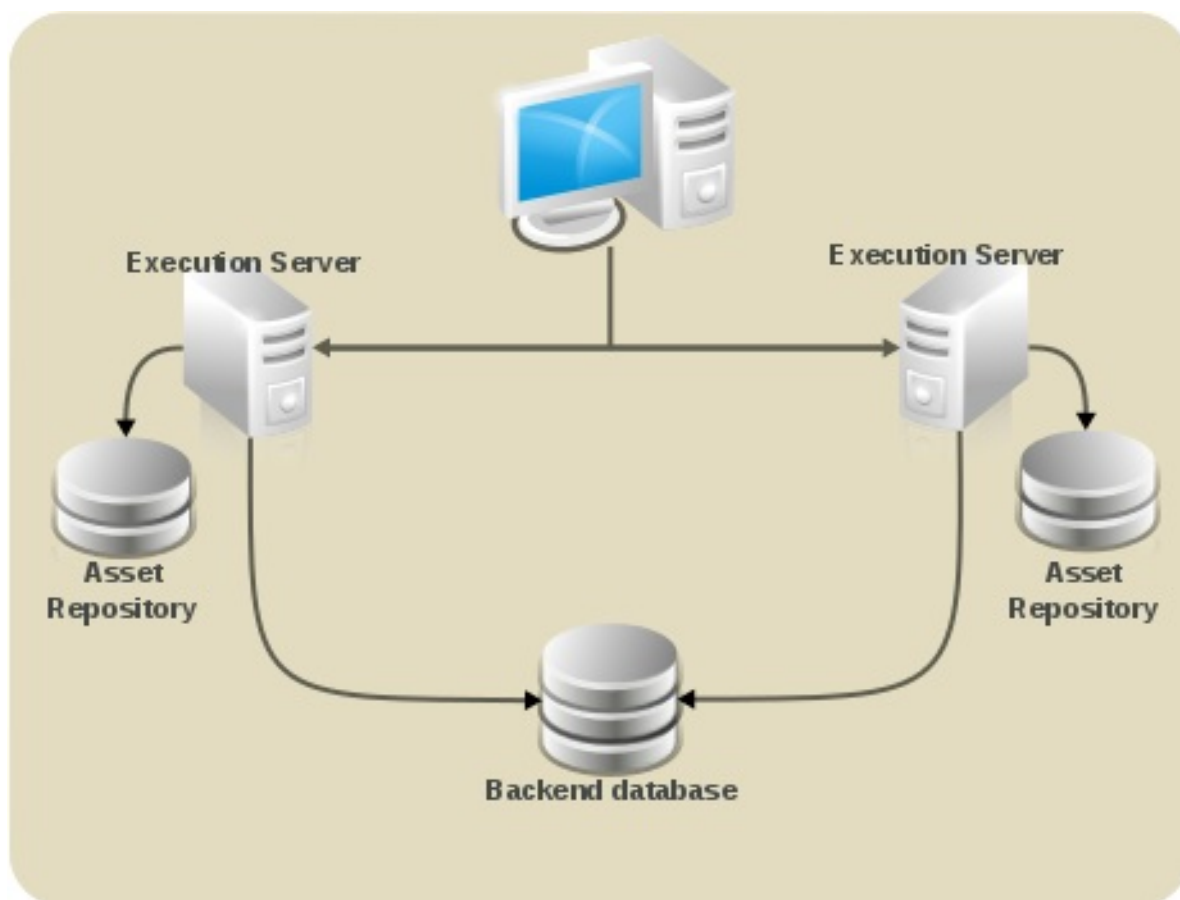


Figure 6.1. Schema of BPMS system with individual system components

GIT REPOSITORY CLUSTERING MECHANISM

To cluster the GIT repository the following is used:

- Apache Zookeeper brings all parts together.
- Apache Helix is the cluster management component that registers all cluster details (the cluster itself, nodes, resources).

The runtime environment, that is the Execution Server, utilizes the following to provide the clustering capabilities:

- kie-commons provides VFS implementation and clustering
- uber fire framework provides backbone of the web applications

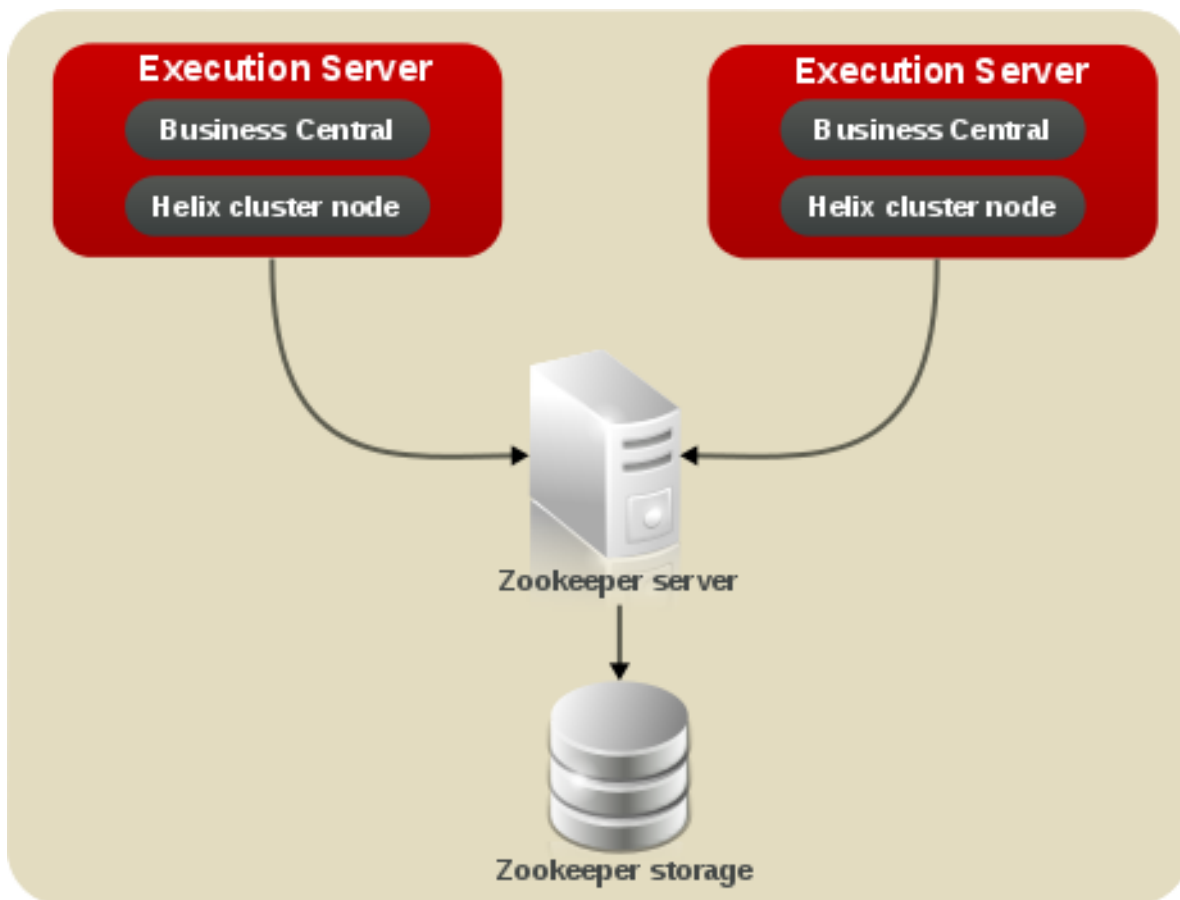


Figure 6.2. Clustering schema with Helix and Zookeeper

A typical clustering setup involves the following:

- Setting up the cluster itself using Zookeeper and Helix
- Setting up the back-end database with Quartz tables and configuration
- Configuring clustering on your container (this documentation provides only clustering instructions for Red Hat JBoss EAP 6)

CLUSTERING MAVEN REPOSITORIES

Various operations within the Business Central publish JARs to the Business Central's internal Maven Repository.

This repository exists on the application server's file-system as regular files and is *not* cluster aware. This folder is not synchronized across the various nodes in the cluster and must be synchronized using external tools like **rsync**.

An alternate to the use of an external synchronization tool is to set the system property `org.guvnor.m2repo.dir` on each cluster node to point to a SAN or NAS. In this case clustering of the Maven repository folder is not needed.

[Report a bug](#)

6.1. SETTING UP A CLUSTER

To cluster your GIT (VFS) repository in Business Central, do the following (If you don't use Business Central, you may skip this section):

1. Download the **jboss-bpms-brms-*VERSION*-redhat-supplementary-tools.zip**, which contains Apache Zookeeper, Apache Helix, and quartz DDL scripts. After downloading, unzip the archive: the **Zookeeper** directory (**\$ZOOKEEPER_HOME**) and the **Helix** directory (**\$HELIX_HOME**) are created.

2. Now Configure Zookeeper:

- a. In the Zookeeper directory, go to **conf** directory and do the following:

```
cp zoo_sample.cfg zoo.cfg
```

- b. Open **zoo.cfg** for editing and adjust the settings including the following:

```
# the directory where the snapshot is stored.
dataDir=$ZOOKEEPER_HOME/tmp/
# the port at which the clients connects
clientPort=2181
```

Make sure the dataDir location exists and is accessible.

3. Set up Zookeeper, so you can use it when creating the cluster with Helix:

- a. Go to the **\$ZOOKEEPER_HOME/bin/** directory and start Zookeeper:

```
./zkServer.sh start
```

You can check the Zookeeper log in the **\$ZOOKEEPER_HOME/bin/zookeeper.out** file.

4. Set up the cluster with the Zookeeper server as the master of the configuration:

- a. Create the cluster:

```
$HELIX_HOME/bin/helix-admin.sh --zkSvr
ZOOKEEPER_HOST:ZOOKEEPER_PORT --addCluster CLUSTER_NAME
```

- b. Add your nodes to the cluster:

```
$HELIX_HOME/bin/helix-admin.sh --zkSvr
ZOOKEEPER_HOST:ZOOKEEPER_PORT --addNode CLUSTER_NAME
NODE_NAMEUNIQUE_ID
```

Example 6.1. Adding two cluster nodes

```
./helix-admin.sh --zkSvr localhost:2181 --addNode bpms-cluster
nodeOne:12345
./helix-admin.sh --zkSvr localhost:2181 --addNode bpms-cluster
nodeTwo:12346
```

5. Add resources to the cluster.

Example 6.2. Adding vfs-repo as resource

```
./helix-admin.sh --zkSvr localhost:2181 --addResource bpms-cluster  
vfs-repo 1 LeaderStandby AUTO_REBALANCE
```

6. Rebalance the cluster.

Example 6.3. Rebalancing the bpms-cluster

```
./helix-admin.sh --zkSvr localhost:2181 --rebalance bpms-cluster  
vfs-repo 2
```

7. Start the Helix controller.

Example 6.4. Starting the Helix controller

```
./run-helix-controller.sh --zkSvr localhost:2181 --cluster bpms-  
cluster 2>&1 > /tmp/controller.log &
```

[Report a bug](#)

6.2. SETTING UP QUARTZ

Before you can configure the database on your application server, you need to prepare the database for Quartz to create Quartz tables, which will hold the timer data, and the Quartz definition file.

Of course, if you are not using Quartz (timers) in your business processes, or if you are not using the Execution Server at all, you can skip this section completely. Please note that if you want to replicate timers in your business process, you need to use the Quartz component.

**WARNING**

Please keep in mind that due to a bug in the implementation of the Quartz API in the current version of Red Hat JBoss BPM Suite, a cluster with EAP 6 nodes and DB2 as the backend database that attempts to create and then deploy a process dealing with timers or deadlines will fail with an error. To get around this issue, remove the constraints on blob size in Quartz DDL for DB2.

To set this up, do the following:

1. Set up the database. Make sure to use one of the supported non-JTA data source. Note, that since Quartz need a non-JTA data source, you cannot use the Business Central data source. In the example code, PostgreSQL with the user **bpms** and password **bpms** is used. This database

will need to be connected to your application server, so make sure to keep a note on the database information and credentials.

2. Create Quartz tables on your database to allow timer events synchronization. To do so, use the DLL script for your database, which is available in the extracted supplementary zip archive in `QUARTZ_HOME/docs/dbTables`.
3. Create the Quartz configuration file **quartz-definition.properties** in **`$JBOSS_HOME/PROFILE/configuration/`** directory and define the Quartz properties.

Example 6.5. Quartz configuration file for a PostgreSQL database

```
#=====
=====
# Configure Main Scheduler Properties
#=====
=====

org.quartz.scheduler.instanceName = jBPMClusteredScheduler
org.quartz.scheduler.instanceId = AUTO

#=====
=====
# Configure ThreadPool
#=====
=====

org.quartz.threadPool.class = org.quartz.simpl.SimpleThreadPool
org.quartz.threadPool.threadCount = 5
org.quartz.threadPool.threadPriority = 5

#=====
=====
# Configure JobStore
#=====
=====

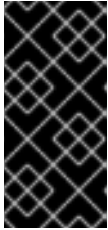
org.quartz.jobStore.misfireThreshold = 60000

org.quartz.jobStore.class=org.quartz.impl.jdbcjobstore.JobStoreCMT
org.quartz.jobStore.driverDelegateClass=org.quartz.impl.jdbcjobsto
re.PostgreSQLDelegate
org.quartz.jobStore.useProperties=false
org.quartz.jobStore.dataSource=managedDS
org.quartz.jobStore.nonManagedTXDataSource=notManagedDS
org.quartz.jobStore.tablePrefix=QRTZ_
org.quartz.jobStore.isClustered=true
org.quartz.jobStore.clusterCheckinInterval = 20000

#=====
=====
# Configure Datasources
#=====
=====
```

```
org.quartz.dataSource.managedDS.jndiURL=jboss/datasources/psbpmsDS
org.quartz.dataSource.notManagedDS.jndiURL=jboss/datasources/quartzNotManagedDS
```

Note the configured datasources that will accommodate the two Quartz schemes at the very end of the file.



IMPORTANT

The recommended interval for cluster discovery is 20 seconds and is set in the `org.quartz.jobStore.clusterCheckinInterval` of the **quartz-definition.properties** file. Depending on your set up consider the performance impact and modify the setting as necessary.

Also note the `org.quartz.jobStore.driverDelegateClass` property that defines the DB dialect to be used when communicating with the set database (in this example, **`org.quartz.impl.jdbcjobstore.PostgreSQLDelegate`**).

[Report a bug](#)

6.3. CONFIGURING CLUSTERING ON RED HAT JBOSS EAP

The information provided in this section is a simple clustering recipe. For additional information on clustering refer to *Red Hat JBoss EAP documentation*.

To configure clustering on Red Hat JBoss EAP 6, do the following:

1. Install your JDBC driver as a core module: copy the driver jar to **`$EAP_HOME/modules/system/layers/base/`** and create a **`module.xml`** file in the directory.
2. Edit the **`module.xml`** file as of the respective module XSD.

Example 6.6. The module.xml file content for a PostgreSQL datasource

```
<module xmlns="urn:jboss:module:1.0" name="org.postgresql">
  <resources>
    <resource-root path="postgresql-jdbc.jar"/>
  </resources>

  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

3. Configure individual server nodes in the **`main-server-group`** element in the **`$EAP_HOME/domain/configuration/host.xml`** file with properties defined in [Table 6.1](#), “Cluster node properties”:

Table 6.1. Cluster node properties

Property name	Value	Description
org.uberfire.nio.git.dir	/home/jbpm/node[N]/repo	GIT (VFS) repository location on node[N]
org.quartz.properties	/bpms/quartz-definition.properties	absolute path to the quartz configuration file
jboss.node.name	nodeOne	node name unique within the cluster
org.uberfire.cluster.id	bpms-cluster	Helix cluster name
org.uberfire.cluster.zk	localhost:2181	Zookeeper location
org.uberfire.cluster.local.id	nodeOne_12345	unique ID of the Helix cluster node Note that : is replaced with __.
org.uberfire.cluster.vfs.lock	vfs-repo	name of the resource defined on the Helix cluster
org.uberfire.nio.git.daemon.port	9418	port used by the VFS repo to accept client connections The port must be unique for each cluster member.
org.uberfire.metadata.index.dir	/home/jbpm/node[N]/index	location where the index for search is to be created (maintained by Apache Lucene)
org.uberfire.cluster.autostart	false	This value delays VFS clustering until the application is fully initialized to avoid conflicts when all cluster members create local clones.
org.uberfire.nio.git.ssh.port	8003	the unique port number for ssh access to the GIT repo for a cluster running on physical machines.
org.uberfire.nio.git.daemon.host	nodeOne	the name of the daemon host machine in a physical cluster.
org.uberfire.nio.git.ssh.host	nodeOne	the name of the SSH host machine in a physical cluster.

Example 6.7. Cluster nodeOne configuration

```

<system-properties>

  <property name="org.uberfire.nio.git.dir"
value="/tmp/bpms/nodeone" boot-time="false"/>
  <property name="jboss.node.name" value="nodeOne" boot-
time="false"/>
  <property name="org.uberfire.cluster.id" value="bpms-cluster"
boot-time="false"/>
  <property name="org.uberfire.cluster.zk" value="localhost:2181"
boot-time="false"/>
  <property name="org.uberfire.cluster.local.id"
value="nodeOne_12345" boot-time="false"/>
  <property name="org.uberfire.cluster.vfs.lock" value="vfs-repo"
boot-time="false"/>
  <property name="org.uberfire.nio.git.daemon.port" value="9418"
boot-time="false"/>
  <property name="org.uberfire.metadata.index.dir"
value="/tmp/jbpm/nodeone" boot-time="false"/>
  <property name="org.uberfire.cluster.autostart" value="false"
boot-time="false"/>
  <property name="org.uberfire.nio.git.ssh.port" value="8003"
boot-time="false"/>
  <property name="org.uberfire.nio.git.daemon.host"
value="nodeOne" />
  <property name="org.uberfire.nio.git.ssh.host" value="nodeOne"
/>
  <property name="org.quartz.properties"
value="/tmp/jbpm/quartz/quartz-db-postgres.properties" boot-
time="false"/>

</system-properties>

```

Example 6.8. Cluster nodeTwo configuration

```

<system-properties>

  <property name="org.uberfire.nio.git.dir"
value="/tmp/bpms/nodetwo" boot-time="false"/>
  <property name="jboss.node.name" value="nodeTwo" boot-
time="false"/>
  <property name="org.uberfire.cluster.id" value="bpms-cluster"
boot-time="false"/>
  <property name="org.uberfire.cluster.zk" value="localhost:2181"
boot-time="false"/>
  <property name="org.uberfire.cluster.local.id"
value="nodeTwo_12346" boot-time="false"/>
  <property name="org.uberfire.cluster.vfs.lock" value="vfs-repo"
boot-time="false"/>
  <property name="org.uberfire.nio.git.daemon.port" value="9418"
boot-time="false"/>
  <property name="org.uberfire.metadata.index.dir"
value="/tmp/jbpm/nodetwo" boot-time="false"/>

```



```

    <property name="org.uberfire.cluster.autostart" value="false"
boot-time="false"/>
    <property name="org.uberfire.nio.git.ssh.port" value="8003"
boot-time="false"/>
    <property name="org.uberfire.nio.git.daemon.host"
value="nodeTwo" />
    <property name="org.uberfire.nio.git.ssh.host" value="nodeTwo"
/>
    <property name="org.quartz.properties"
value="/tmp/jbpm/quartz/quartz-db-postgres.properties" boot-
time="false"/>

</system-properties>

```

4. Add management users as instructed in the *Administration and Configuration Guide* for Red Hat JBoss EAP and application users as instructed in *Red Hat JBoss BPM Suite Administration and Configuration Guide*.
5. Start the application server:

```
]$ $JBOSS_HOME/bin/domain.sh
```

6. Check that the nodes are available.

Deploy the Business Central application to your servers:

1. Change the predefined persistence of the application to the required data base (PostgreSQL): in **persistence.xml** change the following:
 - a. jta-data-source name to the source defined on the application server
(**java:jboss/datasources/psbpmsDS**)
 - b. hibernate dialect to be match the data source dialect
(**org.hibernate.dialect.PostgreSQLDialect**)
2. Log on as the management user to the server Administration console of your domain and add the new deployments using the Runtime view of the console. Once the deployment is added to the domain, assign it to the correct server group (**main-server-group**).

[Report a bug](#)

CHAPTER 7. MAVEN REPOSITORIES

7.1. 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](#).

[Report a bug](#)

7.2. ABOUT THE PROVIDED MAVEN REPOSITORIES

A set of repositories containing artifacts required to build applications based on Red Hat JBoss BPM Suite 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. You can configure Maven to use the repositories following the procedure in [Section 7.4, “Configuring Maven to Use the Online Repositories”](#) or [Section 7.3, “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](#)

7.3. 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 7.1.

1. Download the following ZIP archives containing the required repositories:
 - <http://maven.repository.redhat.com/techpreview/brms6/6.0.3/jboss-bpms-brms-6.0.3.GA-redhat-1-maven-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, configuration of authentication for access to 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 BRMS/BPMS -->
    <profile>
      <id>brms-bpms-local-profile</id>
      <repositories>
        <repository>
          <id>jboss-bpms-brms-6.0.3.GA-redhat-1-maven-
repository</id>
          <name>BRMS/BPMS 6.0.3 GA Repository</name>
          <url>file://<!-- path to the repository -->/jboss-bpms-
brms-6.0.3.GA-redhat-1-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-6.1.1.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-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-bpms-brms-6.0.3.GA-redhat-1-maven-
repository</id>
        <name>BRMS/BPMS 6.0.3 GA Repository</name>
        <url>file://<!-- path to the repository -->/jboss-bpms-
brms-6.0.3.GA-redhat-1-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-6.1.1.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-public-plugin-repository</id>

<url>http://repository.jboss.org/nexus/content/repositories/public/<
/ur>

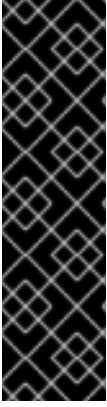
            <releases>
                <enabled>true</enabled>
            </releases>
            <snapshots>
                <enabled>>false</enabled>
            </snapshots>
        </pluginRepository>
    </pluginRepositories>
</profile>
</profiles>

    <!-- Configuring pre-emptive authentication for the repository
server -->
    <server>
        <id>brms-bpms-m2-repo</id>
        <username>admin</username>
        <password>admin</password>
        <configuration>
            <wagonProvider>httpClient</wagonProvider>
            <httpConfiguration>
                <all>
                    <usePreemptive>true</usePreemptive>
                </all>
            </httpConfiguration>
        </configuration>
    </server>

    <!-- Alternative to enabling pre-emptive authentication -
configuring
        the Authorization HTTP header with Base64-encoded credentials
    <server>
        <id>brms-bpms-m2-repo</id>
        <configuration>
            <httpHeaders>
                <property>
                    <name>Authorization</name>
                    <value>Basic YWRtaW46YWRtaW4=</value>
                </property>
            </httpHeaders>
        </configuration>
    </server>
    -->

    <activeProfiles>
        <!-- Activation of the BRMS/BPMS profile -->
        <activeProfile>brms-bpms-local-profile</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 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.

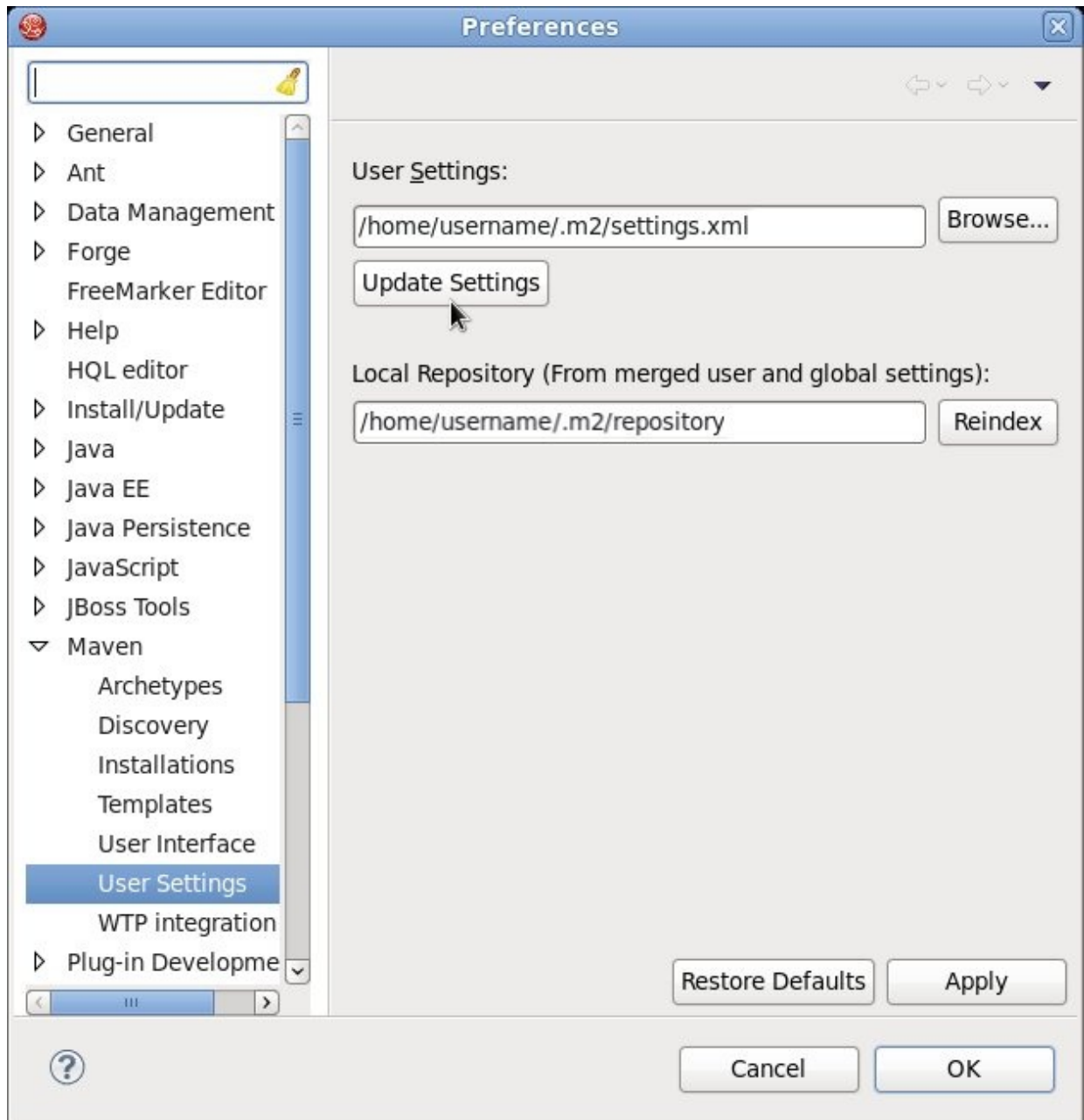


Figure 7.1. Update Maven User Settings

[Report a bug](#)

7.4. CONFIGURING MAVEN TO USE THE ONLINE REPOSITORIES

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

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 7.2. Configuring Maven to Use the Online Repositories

1. Add entries for the online repositories and configuration of authentication for accessing them 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 BRMS/BPMS -->
    <profile>
      <id>brms-bpms-online-profile</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/<
/ur1>
          <releases>
            <enabled>true</enabled>
```



```

        </releases>
        <snapshots>
            <enabled>>false</enabled>
        </snapshots>
    </pluginRepository>
</pluginRepositories>
</profile>
</profiles>

    <!-- Configuring pre-emptive authentication for the repository
server -->
    <server>
        <id>brms-bpms-m2-repo</id>
        <username>admin</username>
        <password>admin</password>
        <configuration>
            <wagonProvider>httpClient</wagonProvider>
            <httpConfiguration>
                <all>
                    <usePreemptive>true</usePreemptive>
                </all>
            </httpConfiguration>
        </configuration>
    </server>

    <!-- Alternative to enabling pre-emptive authentication -
configuring
        the Authorization HTTP header with Base64-encoded credentials
    <server>
        <id>brms-bpms-m2-repo</id>
        <configuration>
            <httpHeaders>
                <property>
                    <name>Authorization</name>
                    <value>Basic YWRtaW46YWRtaW4= </value>
                </property>
            </httpHeaders>
        </configuration>
    </server>
    -->

    <activeProfiles>
        <!-- Activation of the BRMS/BPMS profile -->
        <activeProfile>brms-bpms-online-profile</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.

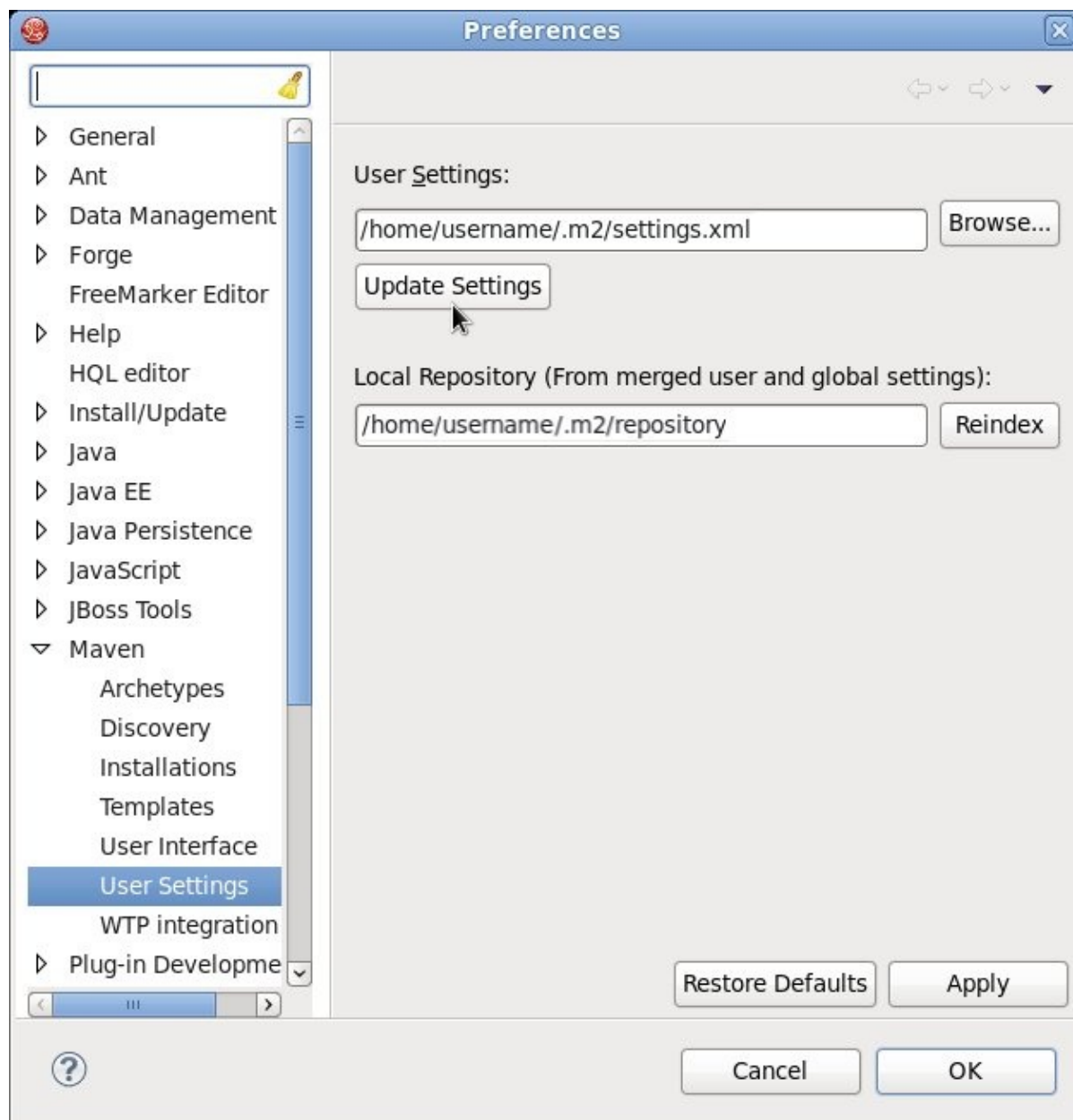


Figure 7.2. Update Maven User Settings

Result

Maven has been configured to use the online repositories provided for Red Hat JBoss BPM Suite.



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](#)

7.5. DEPENDENCY MANAGEMENT

In order to use correct Maven dependencies in your Red Hat JBoss BPM Suite 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 plugins 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:1.2.1-redhat-2`

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

- `org.jboss.ip.component.management:ip-dependency-management-all:1.2.1-redhat-2`
- `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>1.2.1-redhat-2</version>
</parent>

...

<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.jboss.ip.component.management</groupId>
      <artifactId>ip-dependency-management-all</artifactId>
      <type>pom</type>
      <version>1.2.1-redhat-2</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 8. RED HAT JBOSS DEVELOPER STUDIO

8.1. RED HAT JBOSS DEVELOPER STUDIO

Red Hat JBoss Developer Studio is the JBoss integrated development environment (IDE) based on Eclipse and available from the Red Hat customer support portal at <https://access.redhat.com>. JBoss Developer Studio provides plug-ins with tools and interfaces for Red Hat JBoss BRMS and Red Hat JBoss BPM Suite. These plugins are based on the community version of these products. So, the BRMS plugin is called the Drools plugin and the BPM Suite plugin is called the jBPM plugin.

Refer to the *Red Hat JBoss Developer Studio* documentation for installation and set-up instructions.



WARNING

Due to an issue in the way multi-byte rule names are handled, you need to ensure that the instance of JBDS is started with the file encoding set to **UTF-8**. You can do this by editing the `$JBDS_HOME/studio/jbdevstudio.ini` file and adding the following property: `"-Dfile.encoding=UTF-8"`

[Report a bug](#)

8.2. INSTALLING THE JBOSS DEVELOPER STUDIO PLUG-INS

The Drools and jBPM plug-ins for JBoss Developer Studio are available via the update site.

Procedure 8.1. Install the Drools and jBPM JBoss Developer Studio Plug-in

1. Start JBoss Developer Studio.
2. Select **Help** → **Install New Software**.
3. Click **Add** to enter the **Add Repository** menu.
4. Give the software site a name next to **Name** field and add the following url in the **Location** field:
<https://devstudio.jboss.com/updates/7.0/integration-stack/>
5. Click **OK**
6. Select the **JBoss Business Process and Rule Development** feature from the available options and click **Next** and then **Next** again.
7. Read the license and accept it by selecting the appropriate radio button, and click **Finish**.
8. After installation of the plug-ins has completed, restart JBoss Developer Studio.

[Report a bug](#)

8.3. SETTING THE DROOLS RUNTIME

In order to use the JBoss BRMS plug-in with Red Hat JBoss Developer Studio, it is necessary to set up the runtime.

A runtime is a collection of jar files that represent a specific release of the software and provides libraries needed for compilation and running of your business assets.

Procedure 8.2. Configure BRMS Runtime

1. Extract the runtime jar files located in the **jboss-brms-engine.zip** archive of the JBoss BRMS Generic Deployable zip archive (not the EAP6 deployable zip archive) available from [Red Hat Customer Portal](#).
2. From the JBoss Developer Studio menu, select **Window** and click **Preferences**.
3. Select **Drools** → **Installed Drools Runtimes**.
4. Click **Add . . .**; provide a name for the new runtime, and click **Browse** to navigate to the directory where you extracted the runtime files in step 1. Click **OK** to register the selected runtime in JBDS.
5. Mark the runtime you have created as the default Drools runtime by clicking on the check box next to it.
6. Click **OK**. If you have existing projects, a dialog box will indicate that you have to restart JBoss Developer Studio to update the Runtime.

[Report a bug](#)

8.4. CONFIGURING THE JBOSS BPM SUITE SERVER

JBoss Developer Studio can be configured to run the Red Hat JBoss BPM Suite Server.

Procedure 8.3. Configure the Server

1. Open the jBPM view by selecting **Window** → **Open Perspective** → **Other** and select **jBPM** and click **OK**.
2. Add the server view by selecting **Window** → **Show View** → **Other...** and select **Server** → **Servers**.
3. Open the server menu by right clicking the Servers panel and select **New** → **Server**.
4. Define the server by selecting **JBoss Enterprise Middleware** → **JBoss Enterprise Application Platform 6.1+** and clicking **Next**.
5. Set the home directory by clicking the **Browse** button. Navigate to and select the installation directory for JBoss EAP 6.1.1 which has JBoss BPM Suite installed.
6. Provide a name for the server in the **Name** field, make sure that the configuration file is set, and click **Finish**.

[Report a bug](#)

8.5. IMPORTING PROJECTS FROM A GIT REPOSITORY INTO JBOSS DEVELOPER STUDIO

JBoss Developer Studio can be configured to connect to a central Git asset repository. The repository is the space where versions of rules, models, functions and processes are stored. This Git repository must already be defined by the KIE Workbench.

Users can either clone a remote Git repository or import a local Git repository.

Procedure 8.4. Cloning a Remote Git Repository

1. Start the Red Hat JBoss BPM Suite server (if not already running) by selecting the server from the server tab and click the start icon.
2. Simultaneously, start the Secure Shell server, if not running already, by using the following command. The command is Linux and Mac specific only. On these platforms, if sshd has already been started, this command will fail and you may safely ignore this step.

```
/sbin/service sshd start
```

3. In JBoss Developer Studio, select **File** → **Import...** and navigate to the Git folder. Open the Git folder to select **Projects from Git** and click **Next**.
4. Select the repository source as **Clone URI** and click **Next**.
5. Enter the details of the Git repository in the next window and click **Next**.

Clone Git Repository

Source Git Repository

Enter the location of the source repository.

Location

URI:

Host:

Repository path:

Connection

Protocol:

Port:

Authentication

User:

Password:

Store in Secure Store ☐

Figure 8.1. Git Repository Details

6. Select which branch you want to import in the following window and click **Next**.
7. You will be presented with the option to define the local storage for this project. Enter (or select) a non-empty directory, make any configuration changes and click **Next**.
8. Import the project as a general project in the following window and click **Next**. Name the project and click **Finish**.

Procedure 8.5. Importing a Local Git Repository

1. Start the Red Hat JBoss BPM Suite server (if not already running) by selecting the server from the server tab and click the start icon.
2. In JBoss Developer Studio, select **File** → **Import...** and navigate to the Git folder. Open the Git folder to select **Projects from Git** and click **Next**.
3. Select the repository source as **Existing local repository** and click **Next**.

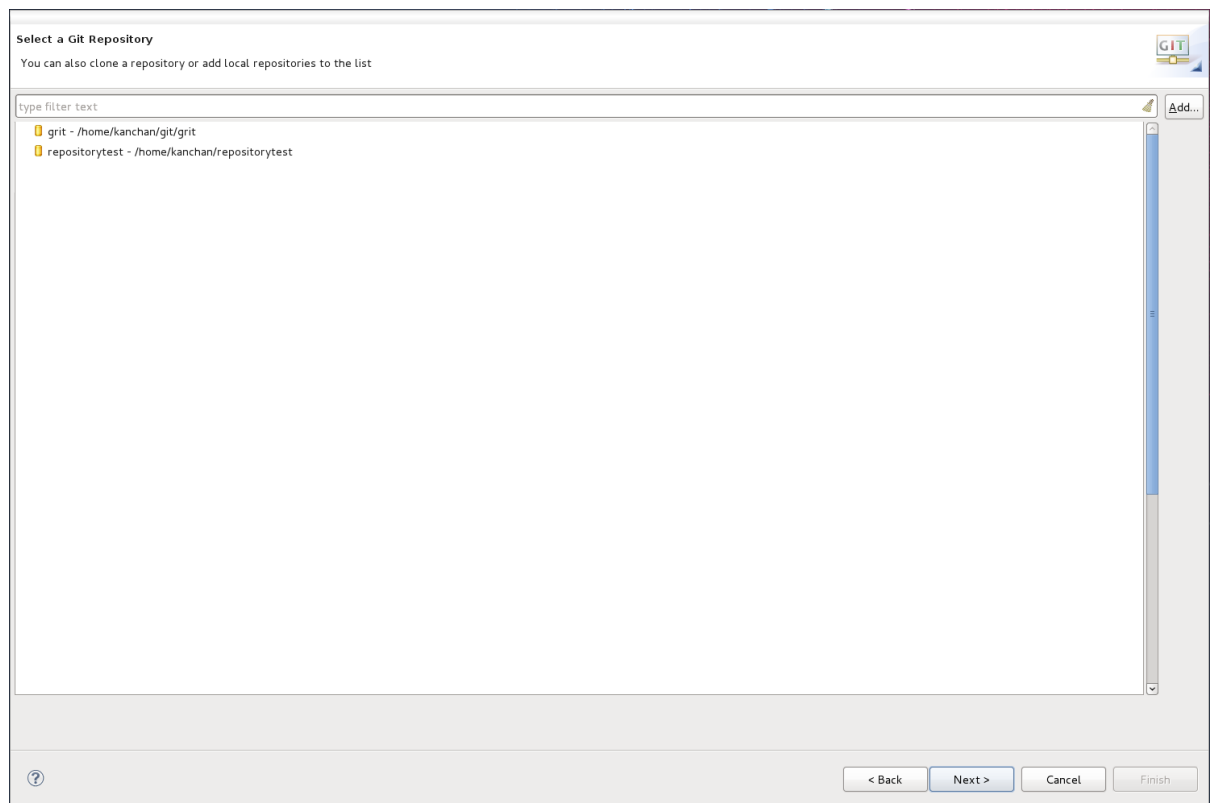


Figure 8.2. Git Repository Details

4. Select the repository that is to be configured from the list of available repositories and click **Next**.
5. In the dialog that opens, select the radio button **Import as general project** from the **Wizard for project import group** and click **Next**. Name the project and click **Finish**.

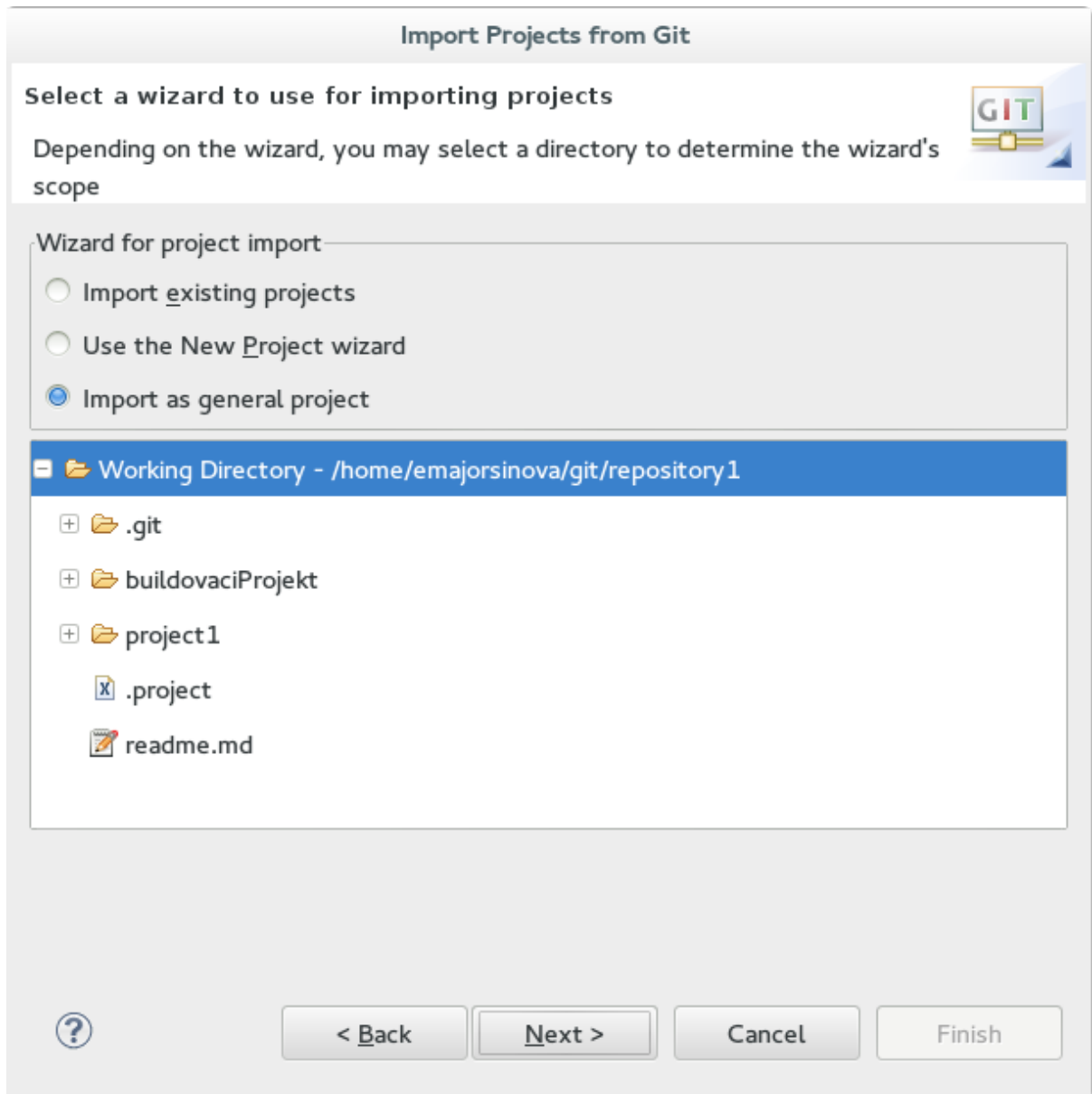


Figure 8.3. Wizard for Project Import

[Report a bug](#)

APPENDIX A. REVISION HISTORY

Revision 1.0.0-60	Thu Mar 12 2015	Vikram Goyal
Built from Content Specification: 22689, Revision: 746986 by vigoyal		