



Red Hat JBoss BPM Suite 6.4

Installation Guide

Red Hat JBoss BPM Suite 6.4 Installation Guide For Red Hat JBoss Administrators

Red Hat JBoss BPM Suite 6.4 Installation Guide

Red Hat JBoss BPM Suite 6.4 Installation Guide For Red Hat JBoss Administrators

Red Hat Customer Content Services

brms-docs@redhat.com

Emily Murphy

Gemma Sheldon

Michele Haglund

Mikhail Ramendik

Stetson Robinson

Vidya Iyengar

Legal Notice

Copyright © 2020 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Abstract

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

Table of Contents

CHAPTER 1. INTRODUCTION	4
1.1. ABOUT RED HAT JBOSS BPM SUITE	4
1.2. SUPPORTED PLATFORMS	4
1.3. SUPPORTED COMPONENT VERSIONS	4
1.4. USE CASE: PROCESS-BASED SOLUTIONS IN LOAN INDUSTRY	5
CHAPTER 2. INSTALLATION OPTIONS	7
2.1. RED HAT JBOSS BPM SUITE INSTALLER INSTALLATION	7
2.1.1. Installing Red Hat JBoss BPM Suite Using Installer	8
Prerequisite	8
LDAP Security (Management Console)	10
LDAP Security (Business Central)	11
2.1.2. Installing Red Hat JBoss BPM Suite Using Installer in CLI Mode	12
Prerequisite	12
2.1.3. Troubleshooting Red Hat JBoss BPM Suite Installer	16
2.2. INSTALLING RED HAT JBOSS BPM SUITE ON RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM	17
Starting Red Hat JBoss BPM Suite in Standalone Mode	18
Configuring Domain Mode	18
Configuring Unified Execution Servers	19
Red Hat JBoss BPM Suite Settings for Red Hat JBoss EAP	19
2.3. GENERIC DEPLOYABLE BUNDLE INSTALLATION	20
2.3.1. Downloading Generic Deployable Package	21
2.3.2. Installing Generic Deployable Package	21
2.3.2.1. Setting up Transaction Manager for Red Hat JBoss Web Server version 2.1 with Tomcat 7 and version 3.0 with Tomcat 8	22
2.3.2.2. Setting up Business Central for Red Hat JBoss Web Server version 2.0 with Tomcat 7 and version 3.0 with Tomcat 8	25
2.3.2.3. Setting up Intelligent Process Server for Red Hat JBoss Web Server version 2.0 with Tomcat 7 and version 3.0 with Tomcat 8	27
2.3.2.4. Setting up Dashbuilder for Red Hat JBoss Web Server version 2.0 with Tomcat 7 and version 3.0 with Tomcat 8	28
2.4. CONFIGURING SSH TO USE RSA	30
CHAPTER 3. PERSISTENCE SETUPS	31
3.1. CONFIGURING PERSISTENCE FOR BUSINESS CENTRAL	31
3.2. CONFIGURING PERSISTENCE FOR DASHBUILDER	32
3.3. CONFIGURING PERSISTENCE FOR THE INTELLIGENT PROCESS SERVER	33
3.4. TROUBLESHOOTING	34
CHAPTER 4. ROLES AND USERS	36
4.1. DEFINING ROLES	36
4.2. CREATING USERS	36
CHAPTER 5. TESTING INSTALLATION	38
5.1. STARTING SERVER	38
5.1.1. Standalone Mode	38
5.1.2. Domain Mode	38
5.2. ENABLING THE SECURITY MANAGER	39
Enabling Security Manager in Red Hat JBoss EAP 6	39
Enabling Security Manager in Red Hat JBoss EAP 7	40
Enabling Security Manager for an embedded application	40
5.2.1. Java Security Manager and Performance Management	40

5.3. LOGGING INTO BUSINESS CENTRAL	41
Troubleshooting	41
5.4. GIT REPOSITORY CLUSTERING MECHANISM	42
Clustering Maven Repositories	43
5.5. CLUSTERING ON RED HAT JBOSS EAP	43
5.5.1. Clustering Using the JAR Installer	44
5.5.2. Starting a Cluster	47
5.5.3. Stopping a Cluster	48
5.5.4. Custom Configuration (Deployable ZIP)	48
5.5.5. Clustering the Intelligent Process Server	54
5.6. GENERIC BUNDLE CLUSTERING	57
5.6.1. Setting a Cluster	57
5.6.2. Starting and Stopping a Cluster	59
5.6.3. Setting Quartz	59
CHAPTER 6. MAVEN REPOSITORIES	62
6.1. ABOUT MAVEN	62
6.2. ABOUT PROVIDED MAVEN REPOSITORIES	62
6.3. CONFIGURING MAVEN TO USE FILE SYSTEM REPOSITORIES	62
6.3.1. Troubleshooting	65
6.3.1.1. Why do I still get errors when building or deploying my applications?	65
6.3.1.2. Why is Red Hat JBoss Developer Studio using my old Maven configuration?	65
6.4. CONFIGURING MAVEN TO USE ONLINE REPOSITORIES	66
6.5. DEPENDENCY MANAGEMENT	69
CHAPTER 7. RED HAT JBOSS DEVELOPER STUDIO	70
7.1. RED HAT JBOSS DEVELOPER STUDIO	70
7.2. INSTALLING THE RED HAT JBOSS DEVELOPER STUDIO PLUG-INS	70
7.3. SETTING THE DROOLS AND JBPM RUNTIME ENVIRONMENTS	70
7.4. CONFIGURING RED HAT JBOSS BPM SUITE SERVER	71
7.5. IMPORTING PROJECTS FROM GIT REPOSITORY INTO RED HAT JBOSS DEVELOPER STUDIO	71
CHAPTER 8. PATCHING AND UPGRADING RED HAT JBOSS BPM SUITE	73
8.1. ABOUT PATCHES AND UPGRADES	73
8.2. APPLYING PATCHES IN RED HAT JBOSS BPM SUITE 6.4	73
Backup Feature	75
Blacklist Feature	75
8.3. PATCHING OTHER PLATFORMS AND APPLICATIONS	76
8.4. UPGRADING TO LATEST MINOR RELEASE	77
APPENDIX A. VERSIONING INFORMATION	79

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 with this release.

Red Hat JBoss BPM Suite is supported for use with Red Hat Enterprise Linux 7 (RHEL7).

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 6.4.(7+) *
- Red Hat JBoss Enterprise Application Platform 7.0 *
- Apache Tomcat 6.0.(37+)
- Apache Tomcat 7.0.(59+)
- Apache Tomcat 8.0.(18+)
- Red Hat JBoss Fuse 6.2.0, 6.2.1 *
- Red Hat JBoss Web Server 2.1 (Tomcat 7) on JDK 1.7 *
- Red Hat JBoss Web Server 3.0 (Tomcat 8) *
- IBM WebSphere Application Server 8.5.(5+) *
- Oracle WebLogic Server 12.1.(3+) *



NOTE

Only Drools, Planner, and jBPM engine artifacts can be deployed on Red Hat JBoss Fuse.

Containers marked with an asterisk (*) are fully supported and tested. For more information, see a list of [Red Hat JBoss BPM Suite 6 Supported Configurations](#) at Red Hat Knowledgebase. Red Hat JBoss Enterprise Application Platform 7.0 is supported only for the Deployable ZIP installation option.

1.3. SUPPORTED COMPONENT VERSIONS

Red Hat JBoss BPM Suite and Red Hat JBoss BRMS 6.4 support the following component versions:

Table 1.1. Supported Maven Artifact Versions

Red Hat JBoss BPM Suite and Red Hat JBoss BRMS Version	Maven Artifact Version
6.4.0	6.5.0.Final-redhat-2
6.4.1	6.5.0.Final-redhat-5
6.4.2	6.5.0.Final-redhat-7
6.4.3	6.5.0.Final-redhat-9
6.4.4	6.5.0.Final-redhat-12
6.4.5	6.5.0.Final-redhat-15

Table 1.2. Supported Bill of Material Versions

Red Hat JBoss BPM Suite and Red Hat JBoss BRMS Version	BOM Version
6.4.0	6.4.0.GA-redhat-2
6.4.1	6.4.1.GA-redhat-3
6.4.2	6.4.2.GA-redhat-2
6.4.3	6.4.3.GA-redhat-2
6.4.4	6.4.4.GA-redhat-3
6.4.5	6.4.5.GA-redhat-3

Use one of the following Bill of Materials (BOM):

- **org.jboss.bom.brms:jboss-brms-bpmsuite-platform-bom:\$VERSION**
- **org.jboss.bom.brms:jboss-brms-bpmsuite-bom:\$VERSION**

For further information about BOM, see the [Dependency Management](#) chapter of *Red Hat JBoss BPM Suite Development Guide*.

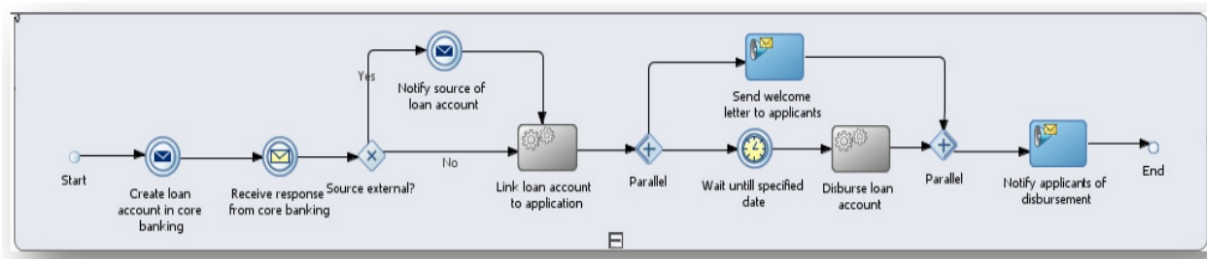
1.4. USE CASE: PROCESS-BASED SOLUTIONS IN LOAN INDUSTRY

This section describes a use case of deploying Red Hat JBoss BPM Suite to automate business processes (such as loan approval process) at a retail bank. This use case is a typical process-based specific deployment that might be the first step in a wider adoption of Red Hat JBoss BPM Suite throughout an enterprise. It leverages features of both business rules and processes of Red Hat JBoss BPM Suite.

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. The bank then processes the application in several steps, such as 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 has 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 Red Hat JBoss BPM Suite. Here is the process flow:

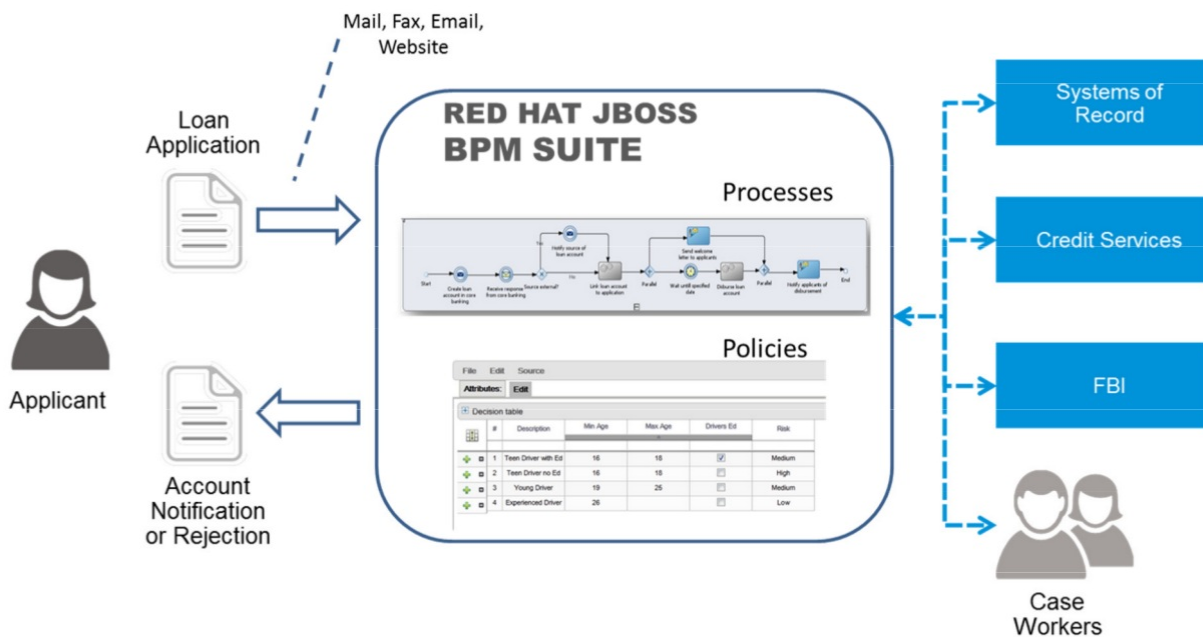
Figure 1.1. High-Level Loan Application Process Flow



Business rules are developed with the rule authoring tools in Red Hat JBoss 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 Red Hat JBoss 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.

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) 6.4.
- ZIP file install which itself comes in two versions:
 - **jboss-bpmsuite-6.4-deployable-eap6.x.zip**: version adapted for deployment on Red Hat JBoss Enterprise Application Platform (EAP 6.4).
 - **jboss-bpmsuite-6.4-deployable-generic.zip**: the deployable version with additional libraries adapted for deployment on Red Hat JBoss Web Server (EWS), 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**".



IMPORTANT

From Red Hat JBoss BPM Suite 6.1 onwards, you must have Red Hat JBoss EAP 6.4 or better already installed before attempting to install Red Hat JBoss BPM Suite.



IMPORTANT

This document describes deploying and configuring Business Central and Intelligent Process Server on the same server. Use this configuration for development environments. In production environments, deploy Business Central and Intelligent Process Server on different servers.

2.1. 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 Red Hat JBoss BPM Suite on an existing Red Hat JBoss EAP 6 installation. However, the JAR installer does not support Red Hat JBoss EAP 7.0. For Red Hat JBoss EAP 7.0, only Deployable Zip installation option is supported.



WARNING

Note that the provided Red Hat JBoss BPM Suite JAR file installer does *not* support the Red Hat JBoss EAP distribution installed by yum or RPM Package Manager. In this case, download the **Red Hat JBoss BPM Suite 6.4.0 Deployable for EAP 6** file and follow the steps described in [Section 2.2, "Installing Red Hat JBoss BPM Suite on Red Hat JBoss Enterprise Application Platform"](#).



INSTALLATION ON IBM JDK

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

2.1.1. Installing Red Hat JBoss BPM Suite Using Installer

The installer for Red Hat JBoss BPM Suite is an executable Java JAR file. You can use it to install Red Hat JBoss BPM Suite on an existing Red Hat JBoss EAP 6.4 installation.



NOTE

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

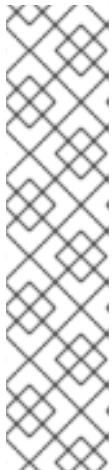
Prerequisite

Before attempting to install Red Hat JBoss BPM Suite, ensure you have already installed Red Hat JBoss EAP 6, version 6.4.7 or higher, and create a back up. Ensure that your user has sufficient rights to complete the installation.

Make sure you have the JAR binary present in **\$PATH** before running the installer. On Red Hat Enterprise Linux, it is present in the package **java-\$JAVA_VERSION-openjdk-devel**.

1. Navigate to the folder where you downloaded the installer file in a command prompt and execute the following command.

```
java -jar jboss-bpmsuite-VERSION-installer.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-bpmsuite-VERSION-installer.jar
```

Furthermore, when running the installer with a 32-bit Java Virtual Machine, you can encounter memory limitations. To solve the issue, run:

```
java -XX:MaxHeapSize=4g -jar jboss-bpmsuite-VERSION-installer.jar
```

The graphical installer will execute and display a splash screen and a license agreement page.

2. Accept the license to proceed.
3. In the next screen, provide the parent location of an existing Red Hat JBoss EAP where you want to install Red Hat JBoss BPM Suite.
4. In the next two screens, create two users: the first one for the management console of the Red Hat JBoss EAP (ManagementRealm) and the second one for managing Red Hat JBoss BPM Suite itself (ApplicationRealm).

Creation of the first user for the management console of Red Hat JBoss EAP is optional and you may skip it if it is not required.

Make a note of these user names and passwords as you will need them to access the Red Hat JBoss EAP server (if you do decide to create it) and the Red Hat JBoss BPM Suite application respectively.

Unless advanced configuration is performed, the Red Hat JBoss BPM Suite user password will be used as the default password for both client and server JMS SSL keystores, as well as password vault keystores.



IMPORTANT

Make sure that the selected user name does *not* conflict with any known title of a role or a group. See [Section 4.1, "Defining Roles"](#).

For example, if there is a role called **admin**, you should *not* create a user with the user name **admin**.



NOTE

The passwords that you create must have at least 8 characters and must contain at least one number and one non-alphanumeric character (*not* including the character **&**).



NOTE

The application role assigned to the second user that you create is the **admin** role. You can assign additional roles to this user at this stage.

5. Next, set up the security environment of your new Red Hat JBoss BPM Suite installation. Decide to enable or disable the Java Security Manager in this step by clicking on the check box. The Java Security Manager makes your system more secure but may downgrade performance. You need to make a decision based on your environment.
6. Choose whether you want to setup pure IPv6 configuration on the server that the installation is taking place. This will enable you to setup runtime IPv6 specific configurations later.
7. Configure runtime environment.
This step provides the option of using a default configuration or specifying an advanced configuration.

- a. *Default Configuration*

Choose **Perform 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.

b. *Advanced Configuration*

Choose to enable advanced configuration options. Select **Perform advanced configuration** and choose the advanced configuration options you want to enable for your environment using the check boxes.

i. *Configure Password Vault*

Vault passwords are used to obfuscate passwords in the various server descriptors using a Java secret key generated during the installation process, or manually using the keytool. This prevents passwords from being stored as plain text in the descriptors. The **Iteration count** and **Salt** are both parameters to the encryption process.

In the case of Red Hat JBoss BPM Suite, a vault is always installed, even if the user does not choose to install one with their own parameters. When this occurs, default values will be used.

For more information about vault passwords, see the *Red Hat JBoss EAP Security Guide* .

ii. *SSL Security*

The **SSL Security** screen enables you to add the **<ssl>** and **<truststore>** elements to the ManagementRealm security realm using the provided keystore.

- The **<ssl>** element causes the server to present the certificate within the keystore as its identity, which enables the user to apply their official certificate.
- The **<truststore>** element enables Client-Cert authentication. This means that, if a remote client attempts to connect to any resource managed by the ManagementRealm, the client can present a certificate, and if an entry in the truststore matches, will be authenticated without needing to provide a user name/password.

The end result is an encrypted connection that is secure between the client and the server for the ManagementRealm.

iii. *LDAP Connection*

This step in the installer enables the user to define an LDAP server, which in turn defines users which should be allowed to authenticate with the ManagementRealm. This replaces the default configuration.

The **LDAP Connection** screen enables users to define how to connect to the LDAP server.

- **Distinguished Name (DN):** the user that can connect to the LDAP server. Typically the DN will uniquely define a special user for this purpose.

LDAP Security (Management Console)

The **Management Console LDAP Configuration** screen enables you to set up a security realm. This defines the **<security-realm>** element to be added to the descriptors, and utilizes the connection defined previously.

- **Base DN:** Will typically define a 'base search' or 'root context' to begin searching for users.
- **Filter Type:** Tells Red Hat JBoss EAP how to find the LDAP attribute that defines a user; it can be a simple attribute, but can also be a complex LDAP filter.

- **Username filter:** The LDAP attribute which holds the user name values. A user name entered in this field is used for search queries as a value of the **uid** attribute. If a user chooses LDAP Syntax Query as a filter type, this query must be specified in this field.
- **Recursive directory search:** If enabled, Red Hat JBoss EAP will traverse the LDAP tree recursively, starting at Base DN. Otherwise, the search will be limited to Base DN.

LDAP Security (Business Central)

Most of the following fields are similar to the Base DN. Contexts are used to search for roles, which enables it to perform authorization in addition to authentication. Otherwise, the context fields are analogous to the Base DN from the previous, and attribute fields are analogous to user name attribute. The filters enable fine grained control over which values of the given attribute will be accepted.

In Red Hat JBoss BPM Suite, the **jbpm.usergroup.callback.properties** and **jbpm.user.info.properties** files used by **LDAPUserGroupInfo** and **LDAPUserInfo** components of Task Service, are also filled by values entered on the **Business Central LDAP Configuration** page.

Input values from **Business Central LDAP Configuration** page are used to configure a new security domain, which make use of **LdapExtended** login module. This security domain is set as default for Business Central web application. For more information about security domains and login modules, see the *Red Hat JBoss EAP Security Guide* .

iv. *Security Domain and JSSE Configuration*

The **Security Domain** screen enables you to configure all of the elements of the **<security-domain>** security subsystem for managing security information, including JSSE configuration. For more detailed information about configuring security domains, see the *Red Hat JBoss EAP Security Guide* .

v. *Configure JMS SSL Keystores*

The **Configuring JMS SSL Keystores** screen enables the encryption of JMS messages sent to Business Central. The client keystores are distributed to systems that need to communicate with the server to facilitate encrypted communications. You can use your pre-existing keystores or generate new ones.

vi. *Configure Clustering*

Selecting this option installs Red Hat JBoss BPM Suite ready for clustered operation. For more information, see [Section 5.5, "Clustering on Red Hat JBoss EAP"](#) .

vii. *Business Central Datasource Setup*

After cluster configuration, the next screen enables you to configure the Business Central data source.

viii. *Dashbuilder Datasource Setup*

The **Dashbuilder Datasource Setup** screen enables you to configure the Dashbuilder data source.

ix. *Configure Optaplanner Execution Server*

Optaplanner is enabled by default. To disable Optaplanner, select **Configure Optaplanner Execution Server** then select **Disable Optaplanner Execution Server** on the **Configure Optaplanner** screen.

x. *Configure KIE Server Management*

Check **Enable KIE server management** if you want Business Central to manage the Intelligent Process Server and use the same data source for both execution servers.

Managing the Intelligent Process Server using the Business Central requires a password vault to be configured. If you did not configure one, a vault with default values is created. See [Configuring Password Vault](#) for further information. The password to the keystore is the same as for the user **bpmsAdmin**.

8. The installer will go through the steps to install Red Hat JBoss BPM Suite and will perform post installation configuration steps when you click **Next**. The installer can also start the Red Hat JBoss BPM Suite server in the mode of your choosing (*standalone* or *domain*, more information in [Section 5.1, "Starting 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.

2.1.2. Installing Red Hat JBoss BPM Suite Using Installer in CLI Mode

The installer for Red Hat JBoss BPM Suite 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 Red Hat JBoss BPM Suite.

Prerequisite

Before attempting to install Red Hat JBoss BPM Suite, ensure you have already installed Red Hat JBoss EAP 6, version 6.4.7 or higher, and create a back up. Ensure that your user has sufficient rights to complete the installation.

Make sure you have the JAR binary present in **\$PATH** before running the installer. On Red Hat Enterprise Linux, it is present in the package **java-\$JAVA_VERSION-openjdk-devel**.

1. Navigate to the folder where you downloaded the installer file in a command prompt and execute the following command.

```
java -jar jboss-bpmsuite-VERSION-installer.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 Red Hat JBoss EAP installation.

```
The location below must specify the JBOSS_HOME of an existing EAP installation.
```

```
[/home/user/BPMSuite-VERSION/jboss-eap-6.4]
```

The installer will verify the location of the Red Hat JBoss EAP installation at the provided location. Enter **1** to confirm and continue.

4. *Optional*: create a user for the management console of Red Hat JBoss EAP (ManagementRealm):

```
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.
```


Enter **0** to skip creating a new administrative user or **1** to create one. If you do decide to create one, then follow these steps:

- a. Enter a user name:

Admin username: [admin]

- b. 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 (not including the character &).

Admin password: []

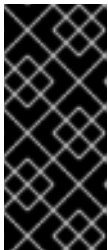
Confirm admin password: [*****]

After this user has been created successfully, continue to the next step.

5. Create a Red Hat JBoss BPM Suite administrator user.

Create a Business Process Management Suite Admin User
Create a BPM Suite admin 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 BPM Suite username cannot be any of the following: 'admin', 'analyst', 'user', 'manager' or 'developer'.

BPM Suite username: [bpmsAdmin]



IMPORTANT

Make sure that the selected user name does *not* conflict with any known title of a role or a group.

For example, if there is a role called **admin**, you should *not* create a user with the user name **admin**.

6. Enter a user name for this user and then create and confirm a password.

The password must have at least 8 characters, and contain at least one number and one non-alphanumeric symbol (not including the character &).

BPM Suite password: []

Confirm BPM Suite password: [*****]

7. After the passwords have been entered and confirmed, you will be given an optional step to define other roles for this user (besides the **admin** role). Enter these roles in a comma-separated list or just press **Enter** to skip this part.

(Optional) You can add this user to additional roles that will be used for task management. These roles are custom named and used again when building your processes with human tasks. Add your custom named roles in a comma separated list below.

Additional user roles: []

-
- 8. 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.

Enable the Java security manager

Input 1 to select, 0 to deselect:

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

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

- 10. Specify whether or not you are using IPv6.

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:

After selecting or deselecting IPv6 configuration, select one of the following options:

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

- 11. Configure the runtime environment by either choosing the default configuration or advanced options.

Configure runtime environment

Red Hat JBoss Business Process Management Suite can be further customized at this time.

0 Perform default configuration

1 Perform advanced configuration

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

- a. Install password vault
Input 1 to select, 0 to deselect:

- b. Enable SSL security
Input 1 to select, 0 to deselect:
- c. Secure EAP Management Console with LDAP
Input 1 to select, 0 to deselect:
- d. Secure Business Central and Dashbuilder with LDAP
Input 1 to select, 0 to deselect:
- e. Add a security-domain
Input 1 to select, 0 to deselect:
- f. Generate JMS Client Keystores
Input 1 to select, 0 to deselect:
- g. Install clustered configuration
Input 1 to select, 0 to deselect:
- h. Install Business-Central Datasource
Input 1 to select, 0 to deselect:
- i. Install Dashbuilder Datasource
Input 1 to select, 0 to deselect:
- j. Configure Optaplanner Execution Server
Input 1 to select, 0 to deselect:
- k. Configure KIE Server management
Input 1 to select, 0 to deselect:

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

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

13. The **.jar** file begins the unpacking and configuration.

14. 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/BPMSuite-VERSION/jboss-eap-6.4  
Would you like to generate an automatic intallation script and properties file?  
(y/n) [n]:
```

15. If you select **y**, provide a path for the automatic installation script:

```
Select path for the automatic installation script: [/home/user/BPMSuite-VERSION/jboss-eap-6.4/AUTO_SCRIPT_FILENAME]
```

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

```
java -jar jboss-bpmsuite-VERSION-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 enables 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
```

16. 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]$
```

17. Start Red Hat JBoss EAP as described in [Section 5.1, "Starting Server"](#).
18. Navigate to <http://localhost:8080/business-central> in a web browser.
19. Log in with the correct user name/password as given to the Red Hat JBoss BPM Suite user in the *Create and confirm a password for the Red Hat JBoss BPM Suite user* step.

2.1.3. Troubleshooting Red Hat JBoss BPM Suite Installer

The Red Hat JBoss BPM Suite installation failed. How do I reinstall Red Hat JBoss BPM Suite?

If the installer detects Red Hat JBoss BPM Suite applications, the installation will not continue. In case of a failed installation:

1. Change into ***EAP_HOME/standalone/deployments***.
2. Delete all Red Hat JBoss BPM Suite deployments, that is:
 - **business-central.war**
 - **dashbuilder.war**
 - **kie-server.war**
3. Start the installer again.

2.2. INSTALLING RED HAT JBOSS BPM SUITE ON RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM

To install Red Hat JBoss BPM Suite 6.4 deployable on Red Hat JBoss EAP 6.4 or later, perform the following tasks:

1. Download the **Red Hat JBoss Enterprise Application Platform 6.4.0** or **Red Hat JBoss Enterprise Application Platform 7.0** ZIP file from the [Customer Portal](#).
2. Extract the ZIP file. This location is your **EAP_HOME**.
3. Patch the Red Hat JBoss EAP to the supported version for your Red Hat JBoss BPM Suite version.
 - See [Red Hat JBoss BPM Suite 6 Supported Configurations](#) to verify which patch is applicable for your Red Hat JBoss BPM Suite version.
 - See [Patching a Zip/Installer Installation](#) from the *Red Hat JBoss EAP Installation Guide* for further information about applying patches.
4. Download the **Red Hat JBoss BPM Suite 6.4.0 Deployable for EAP 6** ZIP file.
5. Extract the file and copy **jboss-eap-6.4/bin/*** into **EAP_HOME/bin/***. When asked, merge the directories.
 - a. If you want to run Red Hat JBoss BPM Suite in the standalone mode:
 - Copy **jboss-eap-6.4/standalone/configuration/*** into **EAP_HOME/standalone/configuration/**.
 - Copy **jboss-eap-6.4/standalone/deployments/*** into **EAP_HOME/standalone/deployments/**.



NOTE

If you already have deployments on your Red Hat JBoss EAP, ensure that your current deployments do not have colliding names with Red Hat JBoss BPM Suite deployments.

- b. If you want to run Red Hat JBoss BPM Suite in the domain mode:
 - Copy **jboss-eap-6.4/domain/configuration/*** into **EAP_HOME/domain/configuration/**.



WARNING

Make sure this step is performed by the same user account that was used to install Red Hat JBoss EAP. This account must not be a superuser account.

6. Add an application user:

```
./EAP_HOME/bin/add-user.sh -a --user bpmsAdmin --password password@1 --role kie-  
server,admin,rest-all,analyst
```

Starting Red Hat JBoss BPM Suite in Standalone Mode

1. Change into **EAP_HOME/bin**.

2. Execute:

In a Unix environment:

```
./standalone.sh
```

In a Windows environment:

```
standalone.bat
```

You can now log into Business Central in your web browser: **localhost:8080/business-central**.

Configuring Domain Mode

If you installed Red Hat JBoss BPM Suite as described in [Section 2.2, "Installing Red Hat JBoss BPM Suite on Red Hat JBoss Enterprise Application Platform"](#), deploy Red Hat JBoss BPM Suite web applications manually.

The **business-central.war**, **dashbuilder.war**, and **kie-server.war** applications are supplied in the **Red Hat JBoss BPM Suite 6.4.0 Deployable for EAP 6** ZIP file as directories. To deploy the applications into domain mode:

1. Package the application directories into archives:
 - a. Extract the following files from the **Red Hat JBoss BPM Suite 6.4.0 Deployable for EAP 6** ZIP file:
 - **jboss-eap-6.4/standalone/deployments/business-central.war**
 - **jboss-eap-6.4/standalone/deployments/kie-server.war**
 - **jboss-eap-6.4/standalone/deployments/dashbuilder.war**
 - b. Create a ZIP file with the content of the **business-central.war**, **kie-server.war**, and **kie-server.war** directories, for example:
 - i. Change into the directory:

```
cd business-central.war
```
 - ii. Execute **zip -r business-central.war .** to create a ZIP file of the content of the **business-central.war** directory.
 - iii. Repeat this procedure for all the web applications you want to deploy.
This ensures that **business-central.war**, **kie-server.war**, and **dashbuilder.war** are archives, not directories.
2. Deploy the archives:
 - a. Add a management user:
-

```
./EAP_HOME/bin/add-user.sh -b --user mgmtAdmin --password password@1 --role
admin
```

- b. Execute `./EAP_HOME/bin/domain.sh`.
- c. Log into `http://localhost:9990/` using your management user.
- d. Click **Deployments** → **Content Repository** → **Add**.
- e. Select and upload the web archive from the file system.
- f. Select the deployment and click **Assign**.



NOTE

If you want to deploy multiple Red Hat JBoss BPM Suite nodes on a single machine, set ports and other properties before assigning the deployment to a server. See [the section called "Red Hat JBoss BPM Suite Settings for Red Hat JBoss EAP"](#) for more information.

- g. Select the server group.

You can now log into Business Central at `localhost:8080/business-central`.



NOTE

To log into 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 [Section 4.2, "Creating Users"](#).

Configuring Unified Execution Servers

To configure Business Central to manage the Intelligent Process Server and use the same data source, follow the instructions in the [Unified Execution Servers](#) section of the *Red Hat JBoss BPM Suite Administration and Configuration Guide*.

The JVM properties in the `*.xml` files referenced in the procedure are already present but commented out. It is sufficient to uncomment them.

Red Hat JBoss BPM Suite Settings for Red Hat JBoss EAP

If you want to run multiple instances of Red Hat JBoss EAP with Red Hat JBoss BPM Suite, the best practice is to set the following properties:

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

When multiple Red Hat JBoss BPM Suite nodes are used on a single machine, the below properties need to be specified:

- `org.uberfire.nio.git.daemon.host`: can be `localhost`.
- `org.uberfire.nio.git.daemon.port`

- **org.uberfire.nio.git.ssh.host**: can be **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 to avoid port conflicts.

Set the properties in the **EAP_HOME/domain/configuration/host.xml** file:

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 **.niogit** files. These files, which should be used by a respective node, would then be grouped in a particular domain.

2.3. GENERIC DEPLOYABLE BUNDLE INSTALLATION

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

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

Note that to install the generic deployable package, you need the following ZIP files:

- **jboss-bpmsuite-*VERSION*-deployable-generic.zip**: contains the **business-central.war**, **dashbuilder.war**, and **kie-server.war** web applications.
- **jboss-bpmsuite-*VERSION*-engine.zip**: supported execution engine libraries for embedding the engine into your application and other libraries needed for generic deployment.

2.3.1. Downloading Generic Deployable Package

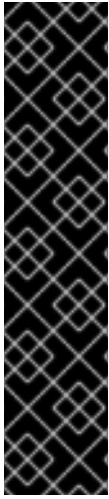
To download the generic deployable Red Hat JBoss BPM Suite package for JBoss Web Server, do the following:

1. Go to the [Red Hat Customer Portal](#) and log in.
2. Click **DOWNLOADS** at the top of the page.
3. From the list of products, choose **Red Hat JBoss BPM Suite**.
4. From the **Version** drop-down menu, select version **6.4** (if not already selected).
5. In the **Software Downloads** section that comes up, navigate to the **Red Hat JBoss BPM Suite 6.4 Deployable for All Supported Containers** row and then click **Download**.
6. Also navigate to the **Red Hat JBoss BPM Suite 6.4 Core Engine** row and click **Download** to download the Red Hat JBoss BPM Suite Core Engine files.

2.3.2. Installing 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 (see [Section 2.3.2.1, “Setting up Transaction Manager for Red Hat JBoss Web Server version 2.1 with Tomcat 7 and version 3.0 with Tomcat 8”](#)).
- Set up the Business Central application: set up users and roles and set up persistence (see [Section 2.3.2.2, “Setting up Business Central for Red Hat JBoss Web Server version 2.0 with Tomcat 7 and version 3.0 with Tomcat 8”](#)).
- Set up the Intelligent Process server (see [Section 2.3.2.3, “Setting up Intelligent Process Server for Red Hat JBoss Web Server version 2.0 with Tomcat 7 and version 3.0 with Tomcat 8”](#)).
- Set up the Dashbuilder application: set up users and roles and set up persistence (see [Section 2.3.2.4, “Setting up Dashbuilder for Red Hat JBoss Web Server version 2.0 with Tomcat 7 and version 3.0 with Tomcat 8”](#)).



IMPORTANT

In a fresh Red Hat JBoss BPM Suite installation, you can encounter exceptions in the log similar to the following:

WARNING: Unable to instantiate EJB Asynchronous Bean. Falling back to Executors' CachedThreadPool.

The exceptions are caused by the code that failed to look up an Enterprise Java Beans object registered under a JNDI name that does not exist in Apache Tomcat. Therefore, a default implementation is used instead.

These messages are only warnings and do not have any impact on the overall functionality of the system.

2.3.2.1. Setting up Transaction Manager for Red Hat JBoss Web Server version 2.1 with Tomcat 7 and version 3.0 with Tomcat 8

1. Extract the generic deployable ZIP package you downloaded from the [Red Hat Customer Portal](#) to a temporary location. This ZIP package contains the following three web application archives: **business-central.war**, **dashbuilder.war**, and **kie-server.war** in an exploded format. Rename these folders to remove the **war** extension.
2. Copy these folders directly under the **\$TOMCAT_DIR/webapps** folder. You should end up with three folders in an exploded format: **\$TOMCAT_DIR/webapps/business-central**, **\$TOMCAT_DIR/webapps/dashbuilder**, and **\$TOMCAT_DIR/webapps/kie-server**.



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-2.1/tomcat7/**.

3. Extract the contents of the Red Hat JBoss BPM Suite Core Engine files archive to a temporary location from where you can copy the required libraries. This folder now contains all the core Red Hat JBoss BPM Suite 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 to **\$TOMCAT_DIR/lib/** directory. These JARs are available in **jboss-bpmsuite-VERSION-deployable-generic.zip** and **jboss-bpmsuite-VERSION-engine.zip**.

- **btm-VERSION.jar**

- **btm-tomcat55-lifecycle-*VERSION*.jar**
- **h2-*VERSION*.jar**
- **jta-*VERSION*.jar**
- **slf4j-api-*VERSION*.jar**
- **slf4j-jdk14-*VERSION*.jar**

Additionally, download the following library and copy it into the **\$TOMCAT_DIR/lib/** folder: [javax.security.jacc-api.jar](#). Add **Valve** configuration into **TOMCAT_HOME/conf/server.xml** inside the **<host>** element as last **Valve** definition:

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

5. Install the driver to your database.
Copy the JAR file with the relevant database driver to **\$TOMCAT_DIR/lib/**.



DRIVER TO THE EMBEDDED H2 DATABASE

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=${btm.root}/work/btm1.tlog
bitronix.tm.journal.disk.logPart2Filename=${btm.root}/work/btm2.tlog
bitronix.tm.resource.configuration=${btm.root}/conf/resources.properties
```

- **resources.properties** (the **resource.ds1.uniqueName** defines the data source 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 2.1. H2 Data Source 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 2.2. MySQL 5.5 Data Source 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/sampledb
resource.ds1.driverProperties.user=dbuser
resource.ds1.driverProperties.password=dbpassword
resource.ds1.allowLocalTransactions=true
```

Example 2.3. DB2 Type 4 Data Source 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/sampledb
resource.ds1.driverProperties.user=dbuser
resource.ds1.driverProperties.password=dbpassword
resource.ds1.allowLocalTransactions=true
```

Example 2.4. Oracle Data Source 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 2.5. Microsoft SQL Server Data Source Definition

```
resource.ds1.className=com.microsoft.sqlserver.jdbc.SQLServerDriver
resource.ds1.uniqueName=jdbc/jbpm
resource.ds1.minPoolSize=0
resource.ds1.maxPoolSize=10
resource.ds1.driverProperties.URL=jdbc:sqlserver://localhost:1433;databaseName=bpm
S;
resource.ds1.driverProperties.user=dbuser
resource.ds1.driverProperties.password=dbpassword
resource.ds1.allowLocalTransactions=true
```

Example 2.6. PostgreSQL Data Source Definition

```
resource.ds1.className=org.postgresql.xa.PGXDataSource
resource.ds1.uniqueName=jdbc/jbpm
resource.ds1.minPoolSize=0
resource.ds1.maxPoolSize=10
```

```
resource.ds1.driverProperties.serverName=localhost
resource.ds1.driverProperties.databaseName=bpm641tomcat
resource.ds1.driverProperties.portNumber=5432
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 configuration file is placed: In `$TOMCAT_DIR/bin/`, create a readable `setenv.sh` file with the following content:

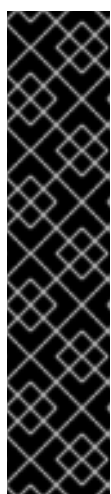
```
CATALINA_OPTS="-Xmx512M -XX:MaxPermSize=512m -
Djava.security.auth.login.config=$CATALINA_HOME/webapps/business-central/WEB-
INF/classes/login.config -Dbtm.root=$CATALINA_HOME -
Dbitronix.tm.configuration=$CATALINA_HOME/conf/btm-config.properties -
Dorg.jbpm.designer.perspective=RuleFlow -
Djbpm.tsr.jndi.lookup=java:comp/env/TransactionSynchronizationRegistry -
Dorg.jboss.logging.provider=jdk -Ddesignerdataobjects=false"
```

The `java.security.auth.login.config` property must be set in order for the `ssh clone` of the Git repository to work.



NOTE

The `-XX:MaxPermSize=512m` JVM property is valid only for JDK 6 and 7. It is *not* valid for JDK 8+.



TOMCAT ON MICROSOFT WINDOWS SYSTEMS

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 -
Djava.security.auth.login.config=C:\apache-tomcat\webapps\business-
central\WEB-INF\classes\login.config -Dbtm.root=C:\apache-tomcat -
Dbitronix.tm.configuration=C:\apache-tomcat\conf\btm-config.properties -
Djbpm.tsr.jndi.lookup=java:comp/env/TransactionSynchronizationRegistry -
Ddesignerdataobjects=false"
```

2.3.2.2. Setting up Business Central for Red Hat JBoss Web Server version 2.0 with Tomcat 7 and version 3.0 with Tomcat 8

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, see the Tomcat 7 documentation).



IMPORTANT

Make sure that the selected user name does *not* conflict with any known title of a role or a group.

For example, if there is a role called **admin**, you should *not* create a user with the user name **admin**.

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. Move (not copy) **kie-tomcat-integration-*VERSION*.jar** from the extracted **jboss-bpmsuite-*VERSION*-engine.zip** to **`$TOMCAT_DIR/lib/`**.
 - c. Copy **jboss-jaxb-api-*VERSION*.jar** from **`$TOMCAT_DIR/webapps/business-central/WEB-INF/lib/`** to **`$TOMCAT_DIR/lib/`**.
2. If you are using a data source other than the default provided by the underlying H2 database, you will need to set up 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 data source with the JNDI name **jdbc/myDatasource** as defined in **uniqueName=jdbc/jbpm** property in the Bitronix **resources.properties** file earlier (for the MySQL option).

- a. In **business-central/META-INF/context.xml**, replace the data source 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 data source JNDI name in the **<res-ref-name>** element with your data source 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 Red Hat JBoss BPM Suite process engine can use the new database.

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

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

Change this value to the data source defined earlier:

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

3. You can now start the JBoss Web Server to log into 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 user name/password you defined earlier in **tomcat-users.xml** file.

2.3.2.3. Setting up Intelligent Process Server for Red Hat JBoss Web Server version 2.0 with Tomcat 7 and version 3.0 with Tomcat 8

After setting up Business Central, it is necessary to configure data sources for the Intelligent Process Server (**kie-server**) as well. Otherwise, Intelligent Process Server tries to find a data source under the JNDI **jboss/datasources/ExampleDS**, which is by default registered only in Red Hat JBoss EAP and *not* in Apache Tomcat.



NOTE

Intelligent Process Server requires a data source *only* if the jBPM extension is enabled. This extension is enabled by default.

Intelligent Process Server needs a dedicated database, which is why it is not possible to reuse the data source registered for Business Central. To add a dedicated data source for **kie-server**, do the following:

1. Copy the JAR file with the relevant database driver to **\$TOMCAT_DIR/lib/**. If you are using an H2 database, this step has already been done during the Business Central installation.
2. Add the data source into the **resources.properties** file. Note that the actual values may differ, based on the underlying database.

```
resource.ds2.className=bitronix.tm.resource.jdbc.lrc.LrcXADataSource
resource.ds2.uniqueName=jdbc/kieserver
resource.ds2.minPoolSize=10
resource.ds2.maxPoolSize=20
resource.ds2.driverProperties.driverClassName=org.h2.Driver
resource.ds2.driverProperties.url=jdbc:h2:file:~/bpm630tomcat7kieserver
resource.ds2.driverProperties.user=sa
resource.ds2.driverProperties.password=
resource.ds2.allowLocalTransactions=true
```

3. Register a new resource in the **kie-server/META-INF/context.xml**:

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

4. Update the **\$TOMCAT_DIR/bin/setenv.sh** file. Add the following data source-related properties for **kie-server**:

```
-Dorg.kie.server.persistence.ds=java:comp/env/jdbc/kieserver
-Dorg.kie.server.persistence.tm=org.hibernate.service.jta.platform.internal.BitronixJtaPlatform
```

2.3.2.4. Setting up Dashbuilder for Red Hat JBoss Web Server version 2.0 with Tomcat 7 and version 3.0 with Tomcat 8



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 version 2.0 with Tomcat 7 and version 3.0 with Tomcat 8”](#). 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 do not 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 set up persistence. In this procedure, you configure a data source 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 data source 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 data source JNDI name in the **<res-ref-name>** element with your data source 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 data source JNDI name in **dashbuilder/WEB-INF/etc/hibernate.cfg.xml** in the **<session-factory>** element:

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

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

2.4. CONFIGURING SSH TO USE RSA

SSH is used to clone Git repositories. By default, the DSA encryption algorithm is provided by Business Central. However, some SSH clients, for example SSH clients in the Fedora 23 environment, use the RSA algorithm instead of the DSA algorithm. Business Central contains a system property that you can use to switch from DSA to RSA if required. Complete the following steps to enable this system property.



NOTE

SSH clients on supported systems, for example Red Hat Enterprise Linux 7, are not affected by this issue.

Procedure

1. Install the Bouncy Castle cryptography extension, as described in [Installing / registering the Bouncy Castle JCE library as a JBoss Module](#).
2. Complete the steps listed in [Issue while cloning git repository from business-central console using ssh protocol](#).

CHAPTER 3. PERSISTENCE SETUPS

3.1. CONFIGURING PERSISTENCE FOR BUSINESS CENTRAL

Red Hat JBoss BPM Suite is configured to use an example data source with Java Naming and Directory Interface (JNDI) name **java:jboss/datasources/ExampleDS**. The example data source is not suitable for production.

To change the data source:

1. Prepare your database:
 - a. Go to the [Product Downloads](#) on the Customer Portal and select Red Hat JBoss BPM Suite.
 - b. Download **Red Hat JBoss BPM Suite 6.4.0 Supplementary Tools**.
 - c. Unzip **jboss-brms-bpmsuite-6.4-supplementary-tools/ddl-scripts**, for example into **/tmp/ddl**.
 - d. Import the DDL script for your database into the database you want to use, for example:

```
psql jbpm < /tmp/ddl/postgresql/postgresql-jbpm-schema.sql
```

2. Issue the following command to install the Java Database Connectivity (JDBC) driver onto your application platform:

```
./jboss-cli.sh (no need to actually connect to the server)
module add --name=org.postgresql --resources=/path/to/postgresql-jdbc-driver.jar --
dependencies=javax.api,javax.transaction.api
```

For further information about deploying JDBC drivers, see [Install a JDBC Driver as a Core Module](#) of the *Red Hat JBoss Enterprise Application Platform Administration and Configuration Guide*.

3. Connect to the running server and create the driver and data source, for example:

```
./jboss-cli.sh --connect --controller=HOST:PORT
/subsystem=datasources/jdbc-driver=postgresql:add(driver-name=postgresql,driver-module-
name=org.postgresql,driver-xa-datasource-class-name=org.postgresql.xa.PGXADatasource)
xa-datasource add --driver-name=postgresql --password=SOME_PASSWORD --user-
name=SOME_USERNAME --xa-datasource-
properties=url=jdbc:postgresql://localhost:5432/jbpm --name=PostgresqlDS --jndi-
name=java:jboss/datasources/PostgresqlDS
```



NOTE

Always use a distributed (XA) data source with the JBoss BPM Suite persistence service.

In general, you should use an XA data source when multiple resources are involved in a single transaction. For example, if you are using a Java Message Service (JMS) executor (which is used by default when asynchronous tasks are included) or timers based on Quartz, you should use an XA data source.

4. Register the data source in Business Central:

a. Open ***EAP_HOME/standalone/deployments/business-central.war/WEB-INF/classes/META-INF/persistence.xml***.

b. Locate the **<jta-data-source>** tag and change it to the JNDI name of your data source, for example:

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

c. Locate the **<properties>** tag and change the **hibernate.dialect** property, for example:

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

3.2. CONFIGURING PERSISTENCE FOR DASHBUILDER

Dashbuilder depends on the configuration of Business Central. Ensure that Business Central is configured according to [Section 3.1, "Configuring Persistence for Business Central"](#).



IMPORTANT

On Unix-like systems override the default value of MySQL **lower_case_table_names** from **0** (case sensitive) to **1** (case insensitive). The Red Hat JBoss BPM Suite KPI queries are written in lowercase, but the table names are in CamelCase. By changing the **lower_case_table_names** property you prevent exceptions from occurring later on.

To change the database for Dashbuilder:

1. Prepare your database:

a. Go to the [Product Downloads](#) on the Customer Portal and select Red Hat JBoss BPM Suite.

b. Download **Red Hat JBoss BPM Suite 6.4.0 Supplementary Tools**.

c. Unzip **jboss-brms-bpmsuite-6.4-supplementary-tools/ddl-scripts**, for example into **/tmp/ddl**.

d. Import the DDL script for your database into the database you want to use, for example:

```
psql jbpms < /tmp/ddl/postgresql/postgres-dashbuilder-schema.sql
```

2. Install the Java Database Connectivity (JDBC) driver onto your application platform. For more information, see [Section 3.1, "Configuring Persistence for Business Central"](#).

3. Create a new data source in ***EAP_HOME/standalone/configuration/standalone.xml***. For more information, see [Section 3.1, "Configuring Persistence for Business Central"](#).

4. Register the data source in Dashbuilder:

a. Open ***EAP_HOME/standalone/deployments/dashbuilder.war/WEB-INF/jboss-web.xml***.

b. Change the **<jndi-name>** to the JNDI name of your data source, for example:

```
<jndi-name>java:jboss/datasources/PostgresqlDS</jndi-name>
```

5. Add the module dependency for the driver:
 - a. Open the ***EAP_HOME/standalone/deployments/dashbuilder.war/WEB-INF/jboss-deployment-structure.xml*** file and locate the **`<dependencies>`** tag.
 - b. Add the JDBC driver module, for example:


```
<module name="org.postgres"/>
```

3.3. CONFIGURING PERSISTENCE FOR THE INTELLIGENT PROCESS SERVER

It is best practice to use a separate server for Intelligent Process Server, not the same server as for Business Central. Also, if you want to use Business Central and Intelligent Process Server as separate execution engines, use different databases for them. If you want to use Business Central and Intelligent Process Server as unified execution engines for shared data, ensure that the configuration for them is exactly the same, including the database, scheduler, executor, and KJAR deployments. For more information, see [Unified Execution Servers](#) in the Red Hat JBoss BPM Suite Administration and Configuration Guide.

To change the database for the Intelligent Process Server:

1. Open ***EAP_HOME/standalone/configuration/standalone.xml*** and locate the **`<system-properties>`** tag.
2. Add the following properties:
 - **`org.kie.server.persistence.ds`**: The JNDI name of your data source.
 - **`org.kie.server.persistence.dialect`**: The hibernate dialect for your database.
 For example:

```
<system-properties>
  <property name="org.kie.server.repo" value="${jboss.server.data.dir}"/>
  <property name="org.kie.example" value="true"/>
  <property name="org.jbpm.designer.perspective" value="full"/>
  <property name="designerdataobjects" value="false"/>
  <property name="org.kie.server.user" value="bpmsUser"/>
  <property name="org.kie.server.pwd" value="bpms123!"/>
  <property name="org.kie.server.location" value="http://localhost:8080/kie-
server/services/rest/server"/>
  <property name="org.kie.server.controller" value="http://localhost:8080/business-
central/rest/controller"/>
  <property name="org.kie.server.controller.user" value="kieserver"/>
  <property name="org.kie.server.controller.pwd" value="kieserver1!"/>
  <property name="org.kie.server.id" value="local-server-123"/>

  <!-- Data source properties. -->
  <property name="org.kie.server.persistence.ds"
value="java:jboss/datasources/KieServerDS"/>
  <property name="org.kie.server.persistence.dialect"
value="org.hibernate.dialect.PostgreSQLDialect"/>
</system-properties>
```

3.4. TROUBLESHOOTING

IBM DB2 database has problems with Dashbuilder

If you want to use an IBM DB2 database as the underlying data source for Business Central, 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 Red Hat JBoss BPM Suite has been run for the first time.

Non-English characters are not displayed in Dashbuilder

If you want to use UTF 8 to display non-English characters, set the encoding at the level of database for Dashbuilder to work correctly. For example, in MySQL, add the following to the server configuration file:

```
[mysqld]
character-set-server=utf8
collation-server=utf8_general_ci
```

Deadlocks occur with Microsoft SQL Server

If you are using Microsoft SQL Server, make sure you have configured proper transaction isolation for your database. If you do not, you may experience deadlocks. The recommended configuration is to turn on `ALLOW_SNAPSHOT_ISOLATION` and `READ_COMMITTED_SNAPSHOT` by entering the following statements:

```
ALTER DATABASE <DBNAME> SET ALLOW_SNAPSHOT_ISOLATION ON
ALTER DATABASE <DBNAME> SET READ_COMMITTED_SNAPSHOT ON
```

Oracle 11 produces multiple warnings without any cause

When you use Oracle 11 as the data source, multiple warning (**WARN**) messages are produced in the logs without any corresponding Business Central activity being performed. This is expected behavior. To turn off these messages, set the level of the **org.hibernate.loader** category of the logger to **ERROR** in the **standalone.xml** file:

```
<logger category="org.hibernate.loader">
  <level name="ERROR"/>
</logger>
```

Vacuumlo deletes active large objects of Red Hat JBoss BPM Suite CLOB data

The vacuumlo utility program removes large objects, whose OIDs are not available in the **oid** or **lo** data columns, from a PostgreSQL database. In Red Hat JBoss BPM Suite, the **text** columns hold large object as well. As vacuumlo does not analyze any other columns than **oid** or **lo**, active objects may be deleted.

To prevent vacuumlo from deleting active large objects, run the **postgresql-jbpm-lo-trigger-clob.sql** script:

1. Download **Red Hat JBoss BPM Suite 6.4.2 Supplementary Tools** from the [Red Hat Customer Portal](#). The script is located in the **ddl-scripts/postgresql/** directory.
2. Make sure that the user which runs the script has the **TRIGGER** privilege on the Red Hat JBoss BPM Suite tables and the **USAGE** privilege to allow the use of the PL/pgSQL procedural language.
3. Run the script to create triggers and the **jbpm_active_clob** table:

```
❯ \i postgresql-jbpm-lo-trigger-clob.sql
```

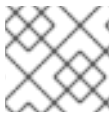
After performing these steps, **jbpm_active_clob** is maintained by the trigger and CLOB references cannot be deleted by vacuumlo.

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.

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.

Procedure: Creating New Users

1. Go to the EAP **bin** directory.
2. On a Unix system, run the following command:

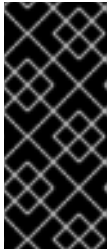
```
./add-user.sh
```

On a Windows system, run:

```
./add-user.bat
```

3. Enter **b** to select the application user and press **Enter**.

4. Accept the default realm (ApplicationRealm) by pressing **Enter**.
5. At the user name prompt, enter the user name and confirm. For example: **helloworlduser**.



IMPORTANT

Make sure that the selected user name does *not* conflict with any known title of a role or a group.

For example, if there is a role called **admin**, you should *not* create a user with the user name **admin**.

6. Create the user 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 (A-Z, a-z), at least one numerical character (0-9) and at least one special character (for example ~ ! @ # \$ % ^ * () - _ + =).

7. Enter a comma-separated list of roles the user will need at the roles prompt (see [Section 4.1, "Defining Roles"](#)).
Note that Business Central users need to have at least the **analyst** role, while the Dashboard Builder users need to have the **admin** role assigned. Roles should be entered as a comma-separated list.
8. Confirm that you want to add the user.
9. Enter **yes** at the next prompt to enable clustering in the future.

CHAPTER 5. TESTING INSTALLATION

5.1. STARTING SERVER



NOTE

If you installed Red Hat JBoss BPM Suite using the generic deployable package on Red Hat Java Web Server, [Section 2.3, “Generic Deployable Bundle Installation”](#) contains the instructions for starting the server. You can ignore the following discussion.

Once the Red Hat JBoss BPM Suite server is installed on Red Hat JBoss EAP, you can run it either in the *standalone* or the *domain* mode.

5.1.1. Standalone Mode



NOTE

If you chose the deployable ZIP package for Red Hat JBoss EAP, the configuration steps are described in [Section 2.2, “Installing Red Hat JBoss BPM Suite on Red Hat JBoss Enterprise Application Platform”](#).

The default startup script, **standalone.sh**, is optimized for performance. To run your server in the performance mode, do the following:

1. On the command line, change to the **EAP_HOME/bin/** directory.
2. In a Unix environment, run:

```
./standalone.sh
```

In a Windows environment, run:

```
./standalone.bat
```

5.1.2. Domain Mode

If you used the JAR installer, referenced in [Section 2.1, “Red Hat JBoss BPM Suite Installer Installation”](#), Red Hat JBoss BPM Suite is already configured for running in the domain mode.



NOTE

If you chose the deployable ZIP package for Red Hat JBoss EAP, the configuration steps for domain mode are described in [Section 2.2, “Installing Red Hat JBoss BPM Suite on Red Hat JBoss Enterprise Application Platform”](#).

To start Red Hat JBoss BPM Suite in the domain mode, perform the following steps:

1. On the command line, change to the **EAP_HOME/bin/** directory.
2. In a Unix environment, run:

```
./domain.sh
```

In a Windows environment, run:

```
./domain.bat
```

5.2. ENABLING THE SECURITY MANAGER

Red Hat JBoss BPM Suite ships with a standard security policy, located in the **kie.policy** file. The location of this file varies depending on your distribution. In order to use the Kie Policy for Java Security Manager, the application server must have its security manager activated. For Red Hat JBoss EAP 6.x or better, it is started using a valid **security.policy** file specified at **java.security.policy** and a valid **kie.policy** file specified at **kie.security.policy**.

This applies to all containers, even when using the rule and process engine in embedded mode.



NOTE

If you installed Red Hat JBoss BPM Suite using the installer, an option to apply the security policy is given to you at the time of installation. Applying the security policy using the installer will modify the **standalone.conf** file to include the **security.policy** and **kie.policy** security policies in the **JBOSS_HOME/bin** folder. These policies will be enabled at runtime using **standalone.sh**.

Enabling Security Manager in Red Hat JBoss EAP 6

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

The **standalone-secure.sh** script is only available when using the Red Hat JBoss EAP Deployable package.



IMPORTANT

It is recommended to use the **standalone-secure.sh** script in production environments.

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 5.2.1, “Java Security Manager and Performance Management”](#).

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

1. On the command line, change to the **EAP_HOME/bin/** directory.
2. In a Unix environment, run:

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

In a Windows environment, run:

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

Enabling Security Manager in Red Hat JBoss EAP 7

If you are using Red Hat JBoss EAP in version 7, the **standalone-secure.sh** script is no longer available. To enable the security manager, start the server with the **-secmgr** and **-Dkie.security.policy=./kie.policy** flags. For example:

```
./standalone.sh -secmgr -Dkie.security.policy=./kie.policy
```

For further information about Java Security Manager in Red Hat JBoss EAP 7, see chapter [Java Security Manager](#) of *Red Hat JBoss Enterprise Application Platform: How to Configure Server Security*.

Enabling Security Manager for an embedded application

Complete the following procedure to enable Security Manager for an embedded application running on Red Hat JBoss EAP 7.

1. Navigate to the [Software Downloads](#) page in the Red Hat Customer Portal (login required), and select the product and version from the drop-down options:
 - **Product:** Process Automation Manager
 - **Version:** 6.4
2. Download the **Red Hat JBoss BPM Suite 6.4.0 Deployable for EAP 7(jboss-bpmsuite-6.4.0.GA-deployable-eap7.x.zip)** file.
3. Extract the **jboss-bpmsuite-6.4.0.GA-deployable-eap7.x.zip** file to a temporary directory. In the following commands this directory is called **TEMP_DIR**.
4. Back up your EAP 7 installation, referred to as **EAP_HOME** in the following commands.
5. Copy the contents of **TEMP_DIR/bin** to **EAP_HOME/bin** and merge the files if prompted.
6. Edit the **kie.policy** file as required for your application. In the following example, replace **AllPermission** with the security policy that you require:

```
grant {
  permission java.security.AllPermission;
};
```

5.2.1. Java Security Manager and Performance Management

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

5.3. LOGGING INTO 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**.

Troubleshooting

Loading... screen does not disappear

When you log into Business Central, it is possible that the *Loading...* screen does not disappear. This can be caused by your firewall interfering with Server Sent Events (SSE) used by Business Central. To work around the problem, disable SSE usage by the Business Central:

1. Create an **ErraiService.properties** file, which contains:
errai.bus.enable_sse_support=false.
2. Copy the file to **INSTALL_PATH/standalone/deployments/business-central.war/WEB-INF/classes/**.
3. Redeploy **business-central.war**.

You can create two types of Red Hat JBoss BPM Suite clusters:

Design-Time Clustering

Allows you to share assets in the Git repository, such as processes, rules, data objects, and others, with all the Red Hat JBoss BPM Suite nodes in your cluster. It is suitable in case of concerns about a single point of failure and high availability during the development process. Design-time clustering makes use of *Apache Helix* and *Apache ZooKeeper*.

Design-time clustering is not required for runtime execution.

Runtime Clustering

Allows you to use the clustering capabilities of your container, such as Red Hat JBoss EAP. Runtime clustering does not require you to manage any Apache Helix or Apache ZooKeeper nodes. *Quartz Enterprise Job Scheduler* is supported if you use timers in your application.



NOTE

If you use the Websphere Application Server, Quartz setup is not necessary. Instead, use clustered EJB Timers. For more information, see the [How to setup BPM Suite Timers to work in Websphere Application Server clustering](#) support article.

You can cluster the following components of Red Hat JBoss BPM Suite:

- Design-time cluster
 - *Git repository*: virtual-file-system (VFS) repository that holds the business assets.

- Runtime cluster
 - *Intelligent Process Server, or web applications* : the web application nodes must share runtime data.
For instructions on clustering the Intelligent Process Server, see [Section 5.5.5, “Clustering the Intelligent Process Server”](#), or the clustering documentation of your container.
 - *Back-end database*: database with the state data, such as process instances, KIE sessions, history log, and similar.

5.4. GIT REPOSITORY CLUSTERING MECHANISM

To cluster the Git repository, Red Hat JBoss BPM Suite uses:

Apache Helix

Provides cluster management functionality that allows you to synchronize and replicate data among the nodes in your cluster. Apache Helix cluster is managed by Apache ZooKeeper. With Apache Helix, you can define a cluster, add nodes to the cluster, remove nodes from the cluster, and perform other cluster-management tasks.

Additional information:

- Apache Helix needs to be configured on a single node only. The configuration is then stored and distributed by ZooKeeper.
- Apache Helix cluster is administered by the **helix-admin.sh** script. See [Apache Helix documentation](#) for the list of commands as well as alternative ways of managing Apache Helix cluster.
- Apache Helix cluster needs exactly one controller, which must be aware of all the nodes. See [Apache Helix controller](#) documentation and [Apache Helix architecture](#) documentation.

Apache ZooKeeper

Allows you to synchronize and replicate data from the Apache Helix cluster. An Apache ZooKeeper cluster is known as an *ensemble* and requires a majority of the servers to be functional for the service to be available.

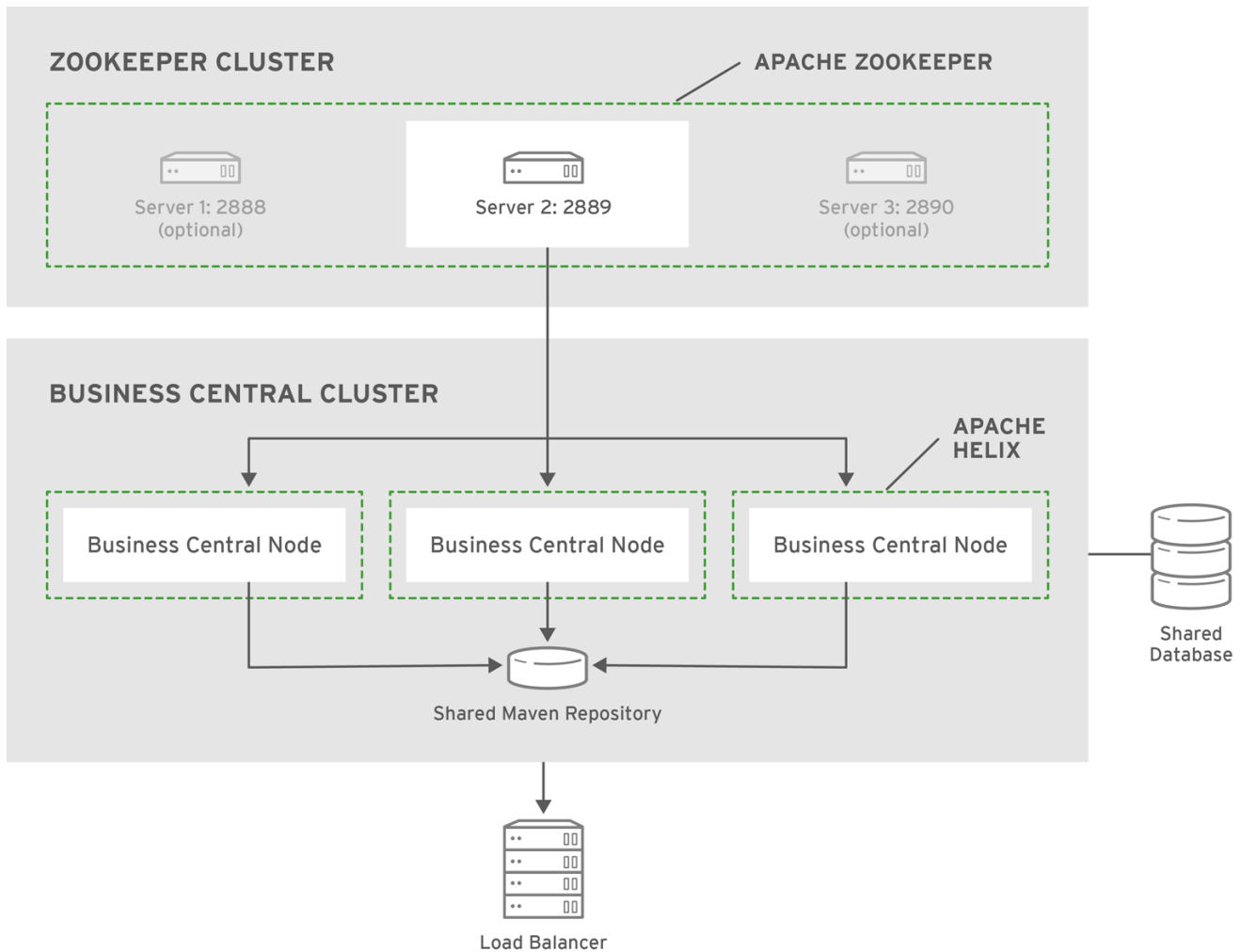
However, an ensemble is not required for any type of clustering. Only a single instance of ZooKeeper is required to allow Red Hat JBoss BPM Suite to replicate its data; the ZooKeeper ensemble serves to provide redundancy and protect against the failure of ZooKeeper itself.

Additional information:

- For more information about failure recovery, see [Apache ZooKeeper Data File Management](#).
- For a list of commands, see [Apache ZooKeeper ZooKeeper Commands: The Four Letter Words](#).

The relationship between Apache Helix and Apache ZooKeeper:

Figure 5.1. Schema of Red Hat JBoss BPM Suite Cluster



A typical clustering setup involves the following:

1. Configuring the cluster using Apache ZooKeeper and Apache Helix. This is required only for design-time clustering.
2. Configuring the back-end database with Quartz tables. This is required only for processes with timers.
3. Configuring clustering on your container. *Red Hat JBoss BPM Suite Installation Guide* provides only clustering instructions for Red Hat JBoss EAP 6.

Clustering Maven Repositories

Various Business Central operations publish JAR files to the Business Central's internal Maven Repository.

This repository exists on the application server 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 alternative 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 such case, clustering of the Maven repository folder is not needed.

5.5. CLUSTERING ON RED HAT JBOSS EAP

To install Red Hat JBoss BPM Suite in the clustered mode, the JAR installer provides a sample setup. You can configure clustering with the deployable ZIP for EAP as well.

5.5.1. Clustering Using the JAR Installer



NOTE

The JAR installer provides sample setup only. Adjusting the configuration is necessary for it to suit your project's needs.

Using the JAR installer, described in [Section 2.1, "Red Hat JBoss BPM Suite Installer Installation"](#) , you can set up a basic clustering configuration of Red Hat JBoss BPM Suite.

The automatic configuration creates:

- ZooKeeper ensemble with three ZooKeeper nodes
- A Helix cluster
- Two Quartz datastores (one managed, one unmanaged)

This Red Hat JBoss BPM Suite setup consists of two EAP nodes that share a Maven repository, use Quartz for coordinating timed tasks, and have **business-central.war**, **dashbuilder.war**, and **kie-server.war** deployed. To customize the setup to fit your scenario, or to use clustering with the deployable ZIP, see [Section 5.5.4, "Custom Configuration \(Deployable ZIP\)"](#) and the clustering documentation of your container.

Follow the installation process described in [Section 2.1.1, "Installing Red Hat JBoss BPM Suite Using Installer"](#).

1. In **Configure runtime environment**, select **Install clustered configuration** and click **Next**.
2. Select the JDBC vendor for your database.
3. Provide the corresponding driver JAR(s):
 - Select one or more files on the filesystem.
 - Provide one or more URLs. The installer downloads the files automatically.

The installer copies the JAR(s) into **EAP_HOME/modules** and creates corresponding **module.xml** file.

Figure 5.2. JDBC Driver Setup

Red Hat JBoss BPM Suite Installer

RED HAT JBOSS[®] BPM SUITE

JDBC Driver Setup

Red Hat JBoss Enterprise Application Platform supports a number of JDBC drivers, which convert application code to the relevant database language.
If your driver is not supported, [click here](#).

Driver vendor:

Driver JAR locations:

JDBC Name:

Module Name:

XA Classname:

Directory Structure:

Red Hat JBoss Middleware

4. Enter the url, username, and password for accessing the database used by Quartz.
The installer creates:

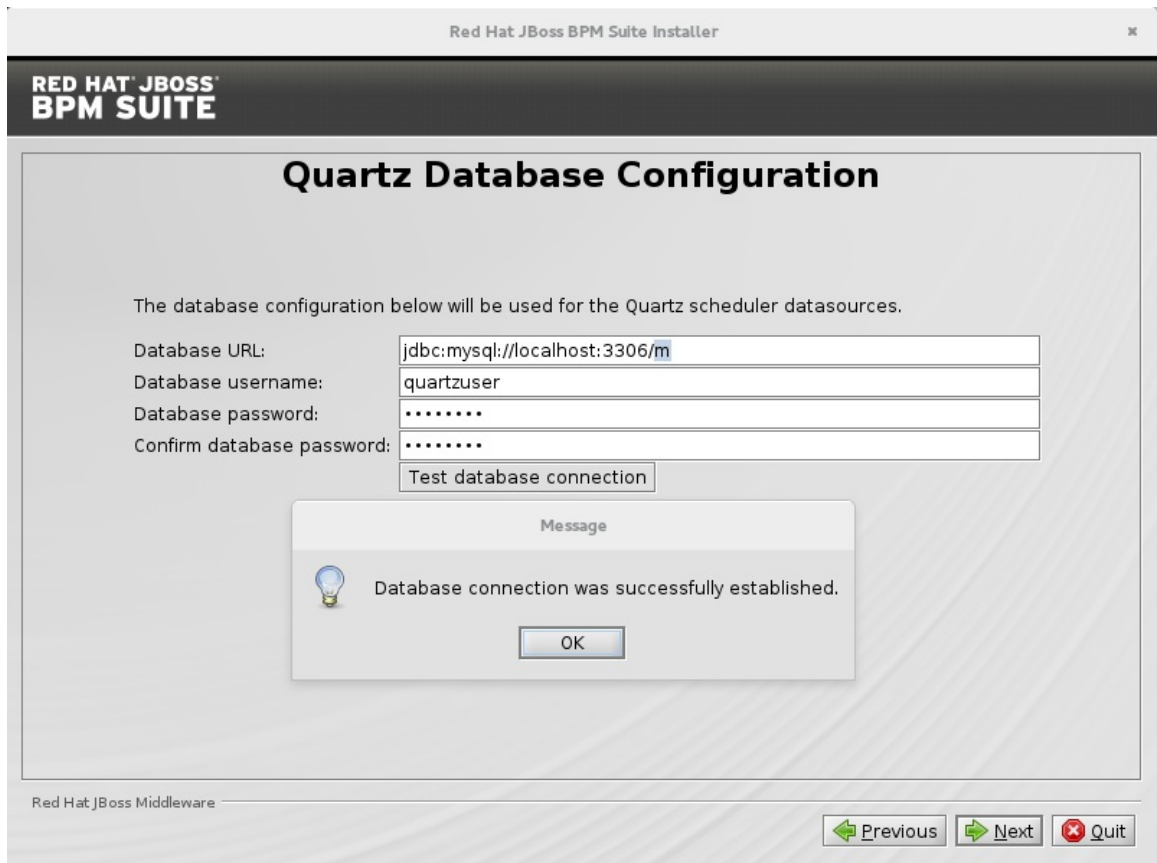
- The Quartz definition file in ***EAP_HOME/domain/configuration/quartz-definition.properties***
- Two Quartz data sources in ***EAP_HOME/domain/domain.xml***
Edit the ***domain.xml*** file to customize the setup.



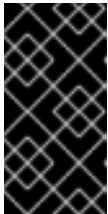
NOTE

During the installation, Quartz DDL scripts will be run on the database selected in this step. The scripts make changes needed for Quartz to operate, such as adding tables. You can view the scripts in ***EAP_HOME/jboss-brms-bpmsuite-6.4-supplementary-tools/ddl-scripts***. No modifications should be necessary.

Figure 5.3. Quartz Database Configuration



5. Click **Next** to initiate the installation.



IMPORTANT

When using the JAR installer, the **war** archives are automatically created from the applications in ***EAP_HOME/standalone/deployments/***. That means additional space is necessary as the applications exist both in uncompressed and compressed state in the storage during the installation.

Three ZooKeeper instances are created in ***EAP_HOME/jboss-brms-bpmsuite-6.4-supplementary-tools/*** in directories ***zookeeper-one***, ***zookeeper-two***, and ***zookeeper-three***.

After the installation finishes, do not start the server from the installer. To make Apache Helix aware of the cluster nodes, Apache ZooKeeper instances, and start the cluster:

1. Change into ***EAP_HOME/jboss-brms-bpmsuite-6.4-supplementary-tools/helix-core***.
2. Execute the launch script:
On UNIX systems:

```
./startCluster.sh
```

On Windows:

```
./startCluster.bat
```

3. Change into ***EAP_HOME/bin***.

- Execute the following script to start Red Hat JBoss EAP:

On UNIX systems:

```
./domain.sh
```

On Windows:

```
./domain.bat
```

5.5.2. Starting a Cluster

The `startCluster.sh` script in `EAP_HOME/jboss-brms-bpmsuite-6.4-supplementary-tools/helix-core` initializes and starts the cluster. Once initialized, further usage of `startCluster.sh` results in errors. If you installed Red Hat JBoss BPM Suite cluster with the installer:

- ZOOKEEPER_HOME** is located in `EAP_HOME/jboss-brms-bpmsuite-6.4-supplementary-tools/zookeeper-NUMBER`
- HELIX_HOME** is located in `EAP_HOME/jboss-brms-bpmsuite-6.4-supplementary-tools/helix-core`

To start a cluster:

- Start all your ZooKeeper servers, for example:

On UNIX systems:

```
./ZOOKEEPER_HOME_ONE/bin/zkServer.sh start &
./ZOOKEEPER_HOME_TWO/bin/zkServer.sh start &
./ZOOKEEPER_HOME_THREE/bin/zkServer.sh start &
```

On Windows:

```
ZOOKEEPER_HOME_ONE/bin/zkServer.cmd start
ZOOKEEPER_HOME_TWO/bin/zkServer.cmd start
ZOOKEEPER_HOME_THREE/bin/zkServer.cmd start
```

- Make the Helix Controller aware of the ZooKeeper instance(s). For example:

```
./HELIX_HOME/bin/run-helix-controller.sh --zkSvr
localhost:2181,localhost:2182,localhost:2183 --cluster bpms-cluster 2>&1 >
/tmp/controller.log &
```

- Change into `EAP_HOME/bin` and start Red Hat JBoss EAP:

On UNIX systems:

```
./domain.sh
```

On Windows:

```
./domain.bat
```

4. You can access your Red Hat JBoss BPM Suite nodes. For example, if you created Red Hat JBoss BPM Suite cluster by using the installer, you can access your nodes at:

```
localhost:8080/business-central  
localhost:8230/business-central
```

5.5.3. Stopping a Cluster

To stop your cluster, stop the components in the reversed order from starting it:

1. Stop the instance of Red Hat JBoss EAP, or the container you are using.
2. Stop the Helix Controller process.
On UNIX systems, find the PID of the process:

```
ps aux|grep HelixControllerMain
```

Once you have the PID, terminate the process:

```
kill -15 <pid of HelixControllerMain>
```

On Windows, use the Task Manager to stop the process.

3. Stop the ZooKeeper server(s). For each server instance, execute:
On UNIX systems:

```
./ZOOKEEPER_HOME_ONE/bin/zkServer.sh stop  
./ZOOKEEPER_HOME_TWO/bin/zkServer.sh stop  
./ZOOKEEPER_HOME_THREE/bin/zkServer.sh stop
```

On Windows:

```
ZOOKEEPER_HOME_ONE/bin/zkServer.cmd stop  
ZOOKEEPER_HOME_TWO/bin/zkServer.cmd stop  
ZOOKEEPER_HOME_THREE/bin/zkServer.cmd stop
```

5.5.4. Custom Configuration (Deployable ZIP)

When using Red Hat JBoss EAP clustering, a single Red Hat JBoss EAP domain controller exists with other Red Hat JBoss EAP slaves connecting to it as management users. You can deploy Business Central and dashbuilder as a management user on a domain controller, and the WAR deployments will be distributed to other members of the Red Hat JBoss EAP cluster.

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

1. Configure ZooKeeper and Helix according to [Section 5.6.1, "Setting a Cluster"](#).
2. Configure Quartz according to [Section 5.6.3, "Setting Quartz"](#).
3. Install the JDBC driver. See the [Install a JDBC Driver with the Management Console](#) chapter of the *Red Hat JBoss EAP Administration and Configuration Guide*.

4. Configure the data source for the server. Based on the mode you use, open **domain.xml** or **standalone.xml**, located at **EAP_HOME/MODE/configuration**.
5. Locate the **full** profile, and do the following:

- a. Add the definition of the main data source used by Red Hat JBoss BPM Suite.

Example 5.1. PostgreSQL Data Source Defined as Main Red Hat JBoss BPM Suite Data Source

```
<datasource jndi-name="java:jboss/datasources/psbpmsDS"
  pool-name="postgresDS" enabled="true" use-java-context="true">
  <connection-url>jdbc:postgresql://localhost:5432/jbpm</connection-url>
  <driver>postgres</driver>
  <security>
    <user-name>bpms</user-name>
    <password>bpms</password>
  </security>
</datasource>
```

- b. Add the definition of the data source for the Quartz service.

Example 5.2. PostgreSQL Data Source Defined as Quartz Data Source

```
<datasource jta="false" jndi-name="java:jboss/datasources/quartzNotManagedDS"
  pool-name="quartzNotManagedDS" enabled="true" use-java-context="true">
  <connection-url>jdbc:postgresql://localhost:5432/jbpm</connection-url>
  <driver>postgres</driver>
  <security>
    <user-name>bpms</user-name>
    <password>bpms</password>
  </security>
</datasource>
```

- c. Define the data source driver.

Example 5.3. PostgreSQL Driver Definition

```
<driver name="postgres" module="org.postgresql">
  <xa-datasource-class>org.postgresql.xa.PGXADatasource</xa-datasource-class>
</driver>
```

- d. If you are deploying Red Hat JBoss BPM Suite on Red Hat JBoss EAP 7.0, ensure that the data sources contain schemas. To create the data source schemas, you can use the DDL scripts located in **jboss-bpmsuite-brms-6.4-supplementary-tools.zip**. If your data source does not contain schemas, ensure your nodes start one at a time. Additionally, when deploying on Red Hat JBoss EAP 7.0, open **EAP_HOME/domain/business-central.war/WEB-INF/classes/META-INF/persistence.xml** and change the property **hibernate.hbm2ddl.auto="update"** to **hibernate.hbm2ddl.auto="none"**.

6. Configure individual server nodes that belong to the **main-server-group** in the **EAP_HOME/domain/configuration/host.xml** file with properties defined in [Cluster Node Properties](#).

When configuring a Red Hat JBoss EAP cluster with Apache ZooKeeper, a different number of Red Hat JBoss EAP nodes than Apache ZooKeeper nodes is possible. However, having the same node count for both ZooKeeper and Red Hat JBoss EAP is considered best practice.

Cluster Node Properties

jboss.node.name

A node name unique in a Red Hat JBoss BPM Suite cluster.

Values	Default
String	N/A

org.uberfire.cluster.id

The name of the Helix cluster, for example: **kie-cluster**. You must set this property to the same value as defined in the Helix Controller.

Values	Default
String	N/A

org.uberfire.cluster.local.id

The unique ID of the Helix cluster node. Note that ':' is replaced with '_', for example **node1_12345**.

Values	Default
String	N/A

org.uberfire.cluster.vfs.lock

The name of the resource defined on the Helix cluster, for example: **kie-vfs**.

Values	Default
String	N/A

org.uberfire.cluster.zk

The location of the Zookeeper servers.

Values	Default
String of the form host1:port1,host2:port2,host3:port3,...	N/A

org.uberfire.metadata.index.dir

The location of the **.index** directory, which Apache Lucene uses when indexing and searching.

Values	Default
Path	Current working directory

org.uberfire.nio.git.daemon.host

If the Git daemon is enabled, it uses this property as the localhost identifier.

Values	Default
URL	localhost

org.uberfire.nio.git.daemon.hostport

When running in a virtualized environment, the host's outside port number for the Git daemon.

Values	Default
Port number	9418

org.uberfire.nio.git.daemon.port

If the Git daemon is enabled, it uses this property as the port number.

Values	Default
Port number	9418

org.uberfire.nio.git.dir

The location of the directory **.niogit**. Change the value for example for backup purposes.

Values	Default
Path	Current working directory

org.uberfire.nio.git.ssh.host

If the SSH daemon is enabled, it uses this property as the localhost identifier.

Values	Default
URL	localhost

org.uberfire.nio.git.ssh.hostport

When running in a virtualized environment, the host's outside port number for the SSH daemon.

Values	Default
Port number	8003

org.uberfire.nio.git.ssh.port

If the SSH daemon is enabled, it uses this property as the port number.

Values	Default
Port number	8001

Example 5.4. 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="server1:2181,server2:2182,server3:2183" 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.host" value="nodeOne"/>
  <property name="org.uberfire.nio.git.daemon.port" value="9418" boot-time="false"/>
  <property name="org.uberfire.nio.git.daemon.hostport" value="9418"
    boot-time="false"/>
  <property name="org.uberfire.nio.git.ssh.port" value="8003" boot-time="false"/>
  <property name="org.uberfire.nio.git.ssh.hostport" value="8003" boot-time="false"/>
  <property name="org.uberfire.nio.git.ssh.host" value="nodeOne"/>
  <property name="org.uberfire.metadata.index.dir" value="/tmp/jbpm/nodeone"
    boot-time="false"/>
  <property name="org.uberfire.nio.git.ssh.cert.dir" value="/tmp/jbpm/nodeone"
    boot-time="false"/>
  <property name="org.quartz.properties"
    value="/tmp/jbpm/quartz/quartz-db-postgres.properties" boot-time="false"/>
</system-properties>
```

Example 5.5. 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="server1:2181,server2:2182,server3:2183" 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.host" value="nodeTwo" />
<property name="org.uberfire.nio.git.daemon.port" value="9419" boot-time="false"/>
<property name="org.uberfire.nio.git.daemon.hostport" value="9419"
  boot-time="false"/>
<property name="org.uberfire.nio.git.ssh.port" value="8004" boot-time="false"/>
<property name="org.uberfire.nio.git.ssh.hostport" value="8004" boot-time="false"/>
<property name="org.uberfire.nio.git.ssh.host" value="nodeTwo" />
<property name="org.uberfire.metadata.index.dir" value="/tmp/jbpm/nodetwo"
  boot-time="false"/>
<property name="org.uberfire.nio.git.ssh.cert.dir" value="/tmp/jbpm/nodetwo"
  boot-time="false"/>
<property name="org.quartz.properties"
  value="/tmp/jbpm/quartz/quartz-db-postgres.properties" boot-time="false"/>
</system-properties>

```

Example 5.6. Cluster nodeThree Configuration

```

<system-properties>
  <property name="org.uberfire.nio.git.dir" value="/tmp/bpms/nodethree"
    boot-time="false"/>
  <property name="jboss.node.name" value="nodeThree" boot-time="false"/>
  <property name="org.uberfire.cluster.id" value="bpms-cluster" boot-time="false"/>
  <property name="org.uberfire.cluster.zk"
    value="server1:2181,server2:2182,server3:2183" boot-time="false"/>
  <property name="org.uberfire.cluster.local.id" value="nodeThree_12347"
    boot-time="false"/>
  <property name="org.uberfire.cluster.vfs.lock" value="vfs-repo" boot-time="false"/>
  <property name="org.uberfire.nio.git.daemon.host" value="nodeThree" />
  <property name="org.uberfire.nio.git.daemon.port" value="9420" boot-time="false"/>
  <property name="org.uberfire.nio.git.daemon.hostport" value="9420"
    boot-time="false"/>
  <property name="org.uberfire.nio.git.ssh.port" value="8005" boot-time="false"/>
  <property name="org.uberfire.nio.git.ssh.hostport" value="8005" boot-time="false"/>
  <property name="org.uberfire.nio.git.ssh.host" value="nodeThree" />
  <property name="org.uberfire.metadata.index.dir" value="/tmp/jbpm/nodethree"
    boot-time="false"/>
  <property name="org.uberfire.nio.git.ssh.cert.dir" value="/tmp/jbpm/nodethree"
    boot-time="false"/>
  <property name="org.quartz.properties"
    value="/tmp/jbpm/quartz/quartz-db-postgres.properties" boot-time="false"/>
</system-properties>

```

7. 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*.
8. Change to **EAP_HOME/bin** and start the application server in domain mode:
On UNIX systems:

```
./domain.sh
```

On Windows:

```
./domain.bat
```

9. 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 database (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 in 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**).



NOTE

It is important users explicitly check deployment unit readiness with every cluster member.

When a deployment unit is created on a cluster node, it takes some time before it is distributed among all cluster members. Deployment status can be checked using the UI and REST, however, if the query goes to the node where the deployment was originally issued, the answer is **deployed**. Any request targeting this deployment unit sent to a different cluster member fails with **DeploymentNotFoundException**.

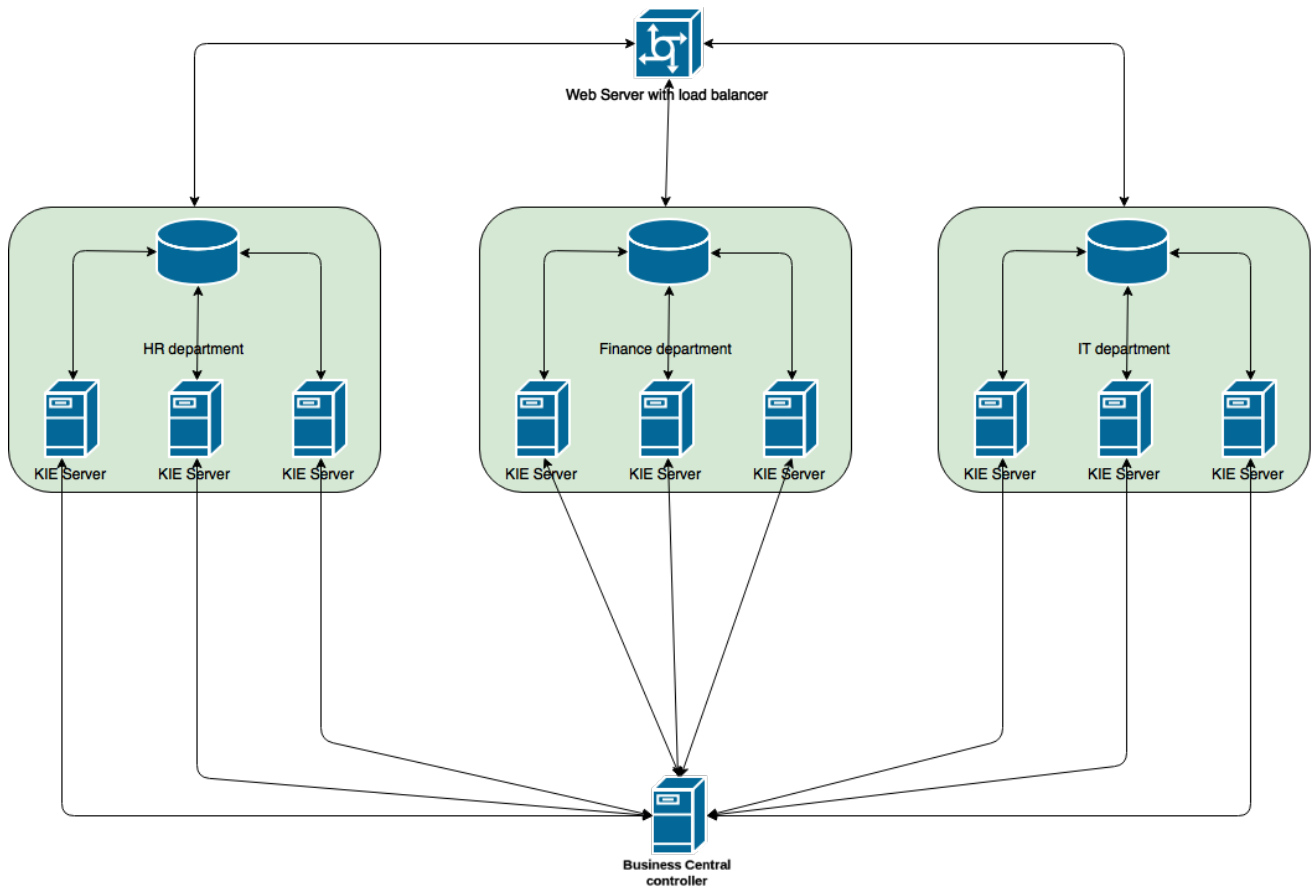
5.5.5. Clustering the Intelligent Process Server

The Intelligent Process Server is a lightweight and scalable component. Clustering it provides many benefits. For example:

- You can partition your resources based on deployed containers.
- You can scale individual instances independently from each other.
- You can distribute the cluster across network and manage it by a single controller.
 - The controller can be clustered into a ZooKeeper ensemble.
- No further components are required.

The basic runtime cluster consists of:

- Multiple Red Hat JBoss EAP instances with Intelligent Process Server
- A controller instance with Business Central



This section describes how to start Intelligent Process Server cluster on Red Hat JBoss EAP 6.4.

Creating an Intelligent Process Server Cluster

1. Change into **CONTROLLER_HOME/bin**.
2. Add a user with the **kie-server** role:

```
$ ./add-user.sh -a --user kieserver --password kieserver1! --role kie-server
```

3. Start your controller:

```
$ ./standalone.sh
```

4. Change into **SERVER_1_HOME**.
5. Deploy **kie-server.war**. Clustered servers do not need **business-central.war** or other applications.
6. See the **<servers>** part of the following **host.xml** as an example of required properties:

```
<server name="server-one" group="main-server-group">
  <system-properties>
    <property name="org.kie.server.location" value="http://localhost:8180/kie-
server/services/rest/server"></property> 1
    <property name="org.kie.server.controller" value="http://localhost:8080/business-
central/rest/controller"></property> 2
    <property name="org.kie.server.controller.user" value="kieserver"></property> 3
    <property name="org.kie.server.controller.pwd" value="kieserver1!"></property> 4
  </system-properties>
</server>
```

```

<property name="org.kie.server.id" value="HR"></property> 5
</system-properties>
<socket-bindings port-offset="100"/>
</server>

<server name="server-two" group="main-server-group" auto-start="true">
<system-properties>
  <property name="org.kie.server.location" value="http://localhost:8230/kie-
server/services/rest/server"></property>
  <property name="org.kie.server.controller" value="http://localhost:8080/business-
central/rest/controller"></property>
  <property name="org.kie.server.controller.user" value="kieserver"></property>
  <property name="org.kie.server.controller.pwd" value="kieserver1!"></property>
  <property name="org.kie.server.id" value="HR"></property>
</system-properties>
<socket-bindings port-offset="150"/>
</server>

```

- 1 **org.kie.server.location:** URL of the server instance.
- 2 **org.kie.server.controller:** Comma-separated list of the controller URL(s).
- 3 **org.kie.server.controller.user:** Username you created for controller authentication. Uses **kieserver** by default.
- 4 **org.kie.server.controller.pwd:** Password for controller authentication. Uses **kieserver1!** by default.
- 5 **org.kie.server.id:** Server identifier that corresponds to template ID defined by the controller instance. Give the same ID to multiple server instances that represent one template.

The example above is defined for Red Hat JBoss EAP domain mode. For further list of bootstrap switches, see section [Bootstrap Switches](#) of the *Red Hat JBoss BPM Suite Administration and Configuration Guide*.

7. Repeat the previous step for as many servers as you need. To start Red Hat JBoss EAP in the domain mode, execute:

```
$ ./SERVER_HOME/bin/domain.sh
```

After connecting the servers to your controller, check the controller log:

```

13:54:40,315 INFO [org.kie.server.controller.impl.KieServerControllerImpl] (http-
localhost/127.0.0.1:8080-1) Server http://localhost:8180/kie-server/services/rest/server connected to
controller
13:54:40,331 INFO [org.kie.server.controller.impl.KieServerControllerImpl] (http-
localhost/127.0.0.1:8080-2) Server http://localhost:8230/kie-server/services/rest/server connected to
controller
13:54:40,348 INFO [org.kie.server.controller.rest.RestKieServerControllerImpl] (http-
localhost/127.0.0.1:8080-1) Server with id 'HR' connected
13:54:40,348 INFO [org.kie.server.controller.rest.RestKieServerControllerImpl] (http-
localhost/127.0.0.1:8080-2) Server with id 'HR' connected

```

Alternatively, to verify in controller Business Central:

1. Log into the controller Business Central.
2. Click **Deploy** → **Execution Servers**.
3. View the remote servers connected to each template.

5.6. GENERIC BUNDLE CLUSTERING

5.6.1. Setting a Cluster



NOTE

If you do not use Business Central, skip this section.

To cluster your Git (VFS) repository in Business Central:

1. Download the **jboss-bpmsuite-brms-*VERSION*-supplementary-tools.zip**, which contains Apache ZooKeeper, Apache Helix, and Quartz DDL scripts.
2. Unzip the archive: the **ZooKeeper** directory (**ZOOKEEPER_HOME**) and the **Helix** directory (**HELIX_HOME**) are created.
3. Configure Apache ZooKeeper:
 - a. In the ZooKeeper directory, change to **conf** and execute:

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

- b. Edit **zoo.cfg**:

```
# The directory where the snapshot is stored.
dataDir=$ZOOKEEPER_HOME/data/

# The port at which the clients connects.
clientPort=2181

# Defining ZooKeeper ensemble.
# server.{ZooKeeperNodeID}={server}:{port:range}
server.1=localhost:2888:3888
server.2=localhost:2889:3889
server.3=localhost:2890:3890
```



NOTE

Multiple ZooKeeper nodes are not required for clustering.

Make sure the **dataDir** location exists and is accessible.

- c. Assign a node ID to each member that will run ZooKeeper. For example, use **1**, **2**, and **3** for node 1, node 2 and node 3 respectively.
The ZooKeeper node ID is specified in a field called **myid** under the data directory of ZooKeeper on each node. For example, on node 1, execute:

```
echo "1" > /zookeeper/data/myid
```

4. Provide further ZooKeeper configuration if necessary.
5. Change to **ZOOKEEPER_HOME/bin/** and start ZooKeeper:

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

You can check the ZooKeeper log in the **ZOOKEEPER_HOME/bin/zookeeper.out** file. Check this log to ensure that the ensemble (cluster) is formed successfully. One of the nodes should be elected as leader with the other two nodes following it.

6. Once the ZooKeeper ensemble is started, configure and start Helix. Helix needs to be configured from a single node only. The configuration is then stored by the ZooKeeper ensemble and shared as appropriate. Configure the cluster with the ZooKeeper server as the master of the configuration:
 - a. Create the cluster by providing the ZooKeeper Host and port as a comma-separated list:

```
$HELIX_HOME/bin/helix-admin.sh --zkSvr ZOOKEEPER_HOST:ZOOKEEPER_PORT -  
-addCluster <clustername>
```

- b. Add your nodes to the cluster:

```
HELIX_HOME/bin/helix-admin.sh --zkSvr ZOOKEEPER_HOST:ZOOKEEPER_PORT --  
addNode <clustername> <name_uniqueID>
```

Example 5.7. Adding Three Cluster Nodes

```
./helix-admin.sh --zkSvr server1:2181,server2:2182,server3:2183 --addNode bpms-  
cluster nodeOne:12345  
./helix-admin.sh --zkSvr server1:2181,server2:2182,server3:2183 --addNode bpms-  
cluster nodeTwo:12346  
./helix-admin.sh --zkSvr server1:2181,server2:2182,server3:2183 --addNode bpms-  
cluster nodeThree:12347
```

7. Add resources to the cluster.

```
helix-admin.sh --zkSvr ZOOKEEPER_HOST:ZOOKEEPER_PORT --addResource  
<clustername> <resourceName> <numPartitions> <stateModelName>
```

Learn more about state machine configuration at [Helix Tutorial: State Machine Configuration](#).

Example 5.8. Adding vfs-repo as Resource

```
./helix-admin.sh --zkSvr server1:2181,server2:2182,server3:2183 --addResource bpms-  
cluster vfs-repo 1 LeaderStandby AUTO_REBALANCE
```

8. Rebalance the cluster with the three nodes.

```
helix-admin.sh --zkSvr ZOOKEEPER_HOST:ZOOKEEPER_PORT --rebalance
<clustername> <resourcename> <replicas>
```

Learn more about rebalancing at [Helix Tutorial: Rebalancing Algorithms](#).

Example 5.9. Rebalancing bpms-cluster

```
./helix-admin.sh --zkSvr server1:2181,server2:2182,server3:2183 --rebalance bpms-
cluster vfs-repo 3
```

In this command, **3** stands for three bpms-cluster nodes.

9. Start the Helix controller in all the nodes in the cluster.

Example 5.10. Starting Helix Controller

```
./run-helix-controller.sh --zkSvr server1:2181,server2:2182,server3:2183 --cluster bpms-
cluster 2>&1 > ./controller.log &
```



NOTE

In case you decide to cluster ZooKeeper, add an odd number of instances in order to recover from failure. After a failure, the remaining number of nodes still need to be able to form a majority. For example a cluster of five ZooKeeper nodes can withstand loss of two nodes in order to fully recover. One ZooKeeper instance is still possible, replication will work, however no recover possibilities are available if it fails.

5.6.2. Starting and Stopping a Cluster

To start your cluster, see [Section 5.5.2, “Starting a Cluster”](#). To stop your cluster, see [Section 5.5.3, “Stopping a Cluster”](#).

5.6.3. Setting Quartz



NOTE

If you are not using Quartz (timers) in your business processes, or if you are not using the Intelligent Process Server, skip this section. If you want to replicate timers in your business process, use the Quartz component.

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.

To configure Quartz:

1. Configure the database. Make sure to use one of the supported non-JTA data sources. Since Quartz needs 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. The database must be connected to your application server.

2. Create Quartz tables on your database to allow timer events synchronization. To do so, use the DDL 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/MODE/configuration/** directory and define the Quartz properties.

Example 5.11. Quartz Configuration File for 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.jdbcjobstore.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 data sources that will accommodate the two Quartz schemes at the very end of the file.



NOTE

For Microsoft SQL Server, add the **acquireTriggersWithinLock** property to the **quartz-definition.properties** file:

```
org.quartz.jobStore.acquireTriggersWithinLock=true
```



CLUSTER NODE CHECK INTERVAL

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.

The **org.quartz.jobStore.driverDelegateClass** property that defines the database dialect. If you use Oracle, set it to **org.quartz.impl.jdbcjobstore.oracle.OracleDelegate**.

4. Provide the absolute path to your **quartz-definition.properties** file in the **org.quartz.properties** property. For further details, see [_cluster_properties_BRMS](#).

Note: To configure the number of retries and delay for the Quartz trigger, you can update the following system properties:

- **org.jbpm.timer.quartz.retries** (default value is 5)
- **org.jbpm.timer.quartz.delay** in milliseconds (default value is 1000)

CHAPTER 6. MAVEN REPOSITORIES

6.1. ABOUT MAVEN

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

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

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

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

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

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



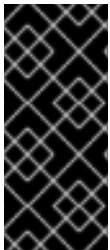
NOTE

Your Red Hat JBoss product has been built with Maven 3.0.x. Therefore, this is the recommended Maven version for building your own SwitchYard applications.

6.2. ABOUT 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.



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 the [JBoss Enterprise Maven Repository](#).

6.3. CONFIGURING MAVEN TO USE FILE SYSTEM REPOSITORIES

Overview

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

1. Download the following ZIP archives containing the required repositories:
 - <https://access.redhat.com/jbossnetwork/restricted/softwareDetail.html?softwareId=48461&product=bpm.suite&version=&downloadType=distributions>
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 `~/.m2/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
         Red Hat JBoss BRMS/Red Hat JBoss BPM Suite -->
    <profile>
      <id>brms-bpms-local-profile</id>
      <repositories>
        <repository>
          <id>jboss-brms-bpmsuite-repository</id>
          <name>BRMS/BPMS 6.4.0 GA Repository</name>
          <url>
            <!-- path to the repository -->
            file:///jboss-brms-bpmsuite-6.4.0.GA-maven-repository/maven-repository
          </url>
          <layout>default</layout>
          <releases>
            <enabled>>true</enabled>
            <updatePolicy>never</updatePolicy>
          </releases>
          <snapshots>
            <enabled>>false</enabled>
            <updatePolicy>never</updatePolicy>
          </snapshots>
        </repository>
      </repositories>
      <pluginRepositories>
        <pluginRepository>
          <id>jboss-brms-bpmsuite-repository</id>
          <name>BRMS/BPMS 6.4.0.GA Repository</name>
          <url>
            <!-- path to the repository -->
            file:///jboss-brms-bpmsuite-6.4.0.GA-maven-repository/maven-repository
          </url>
          <layout>default</layout>
          <releases>
            <enabled>>true</enabled>
            <updatePolicy>never</updatePolicy>
          </releases>
        </pluginRepository>
      </pluginRepositories>
    </profile>
  </profiles>
</settings>
```

```

    <snapshots>
      <enabled>>false</enabled>
      <updatePolicy>never</updatePolicy>
    </snapshots>
  </pluginRepository>
</pluginRepositories>
</profile>
</profiles>

<servers>
  <!-- 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>
  -->
</servers>

<activeProfiles>
  <!-- Activation of the Red Hat JBoss BRMS/Red Hat JBoss BPM Suite profile -->
  <activeProfile>brms-bpms-local-profile</activeProfile>
</activeProfiles>
</settings>

```

Result

The Maven repositories are downloaded, unzipped in a local file system, registered in Maven's `~/.m2/settings.xml` file, and ready to be used when performing Maven builds.



NOTE

To use the Maven Central Repositories offline, perform a KJAR build in the Maven online environment, then copy the `~/.m2/repository` folder to the offline environment.

6.3.1. Troubleshooting

6.3.1.1. Why do I still get errors when building or deploying my applications?

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***

Your cached local Maven repository might contain outdated artifacts.

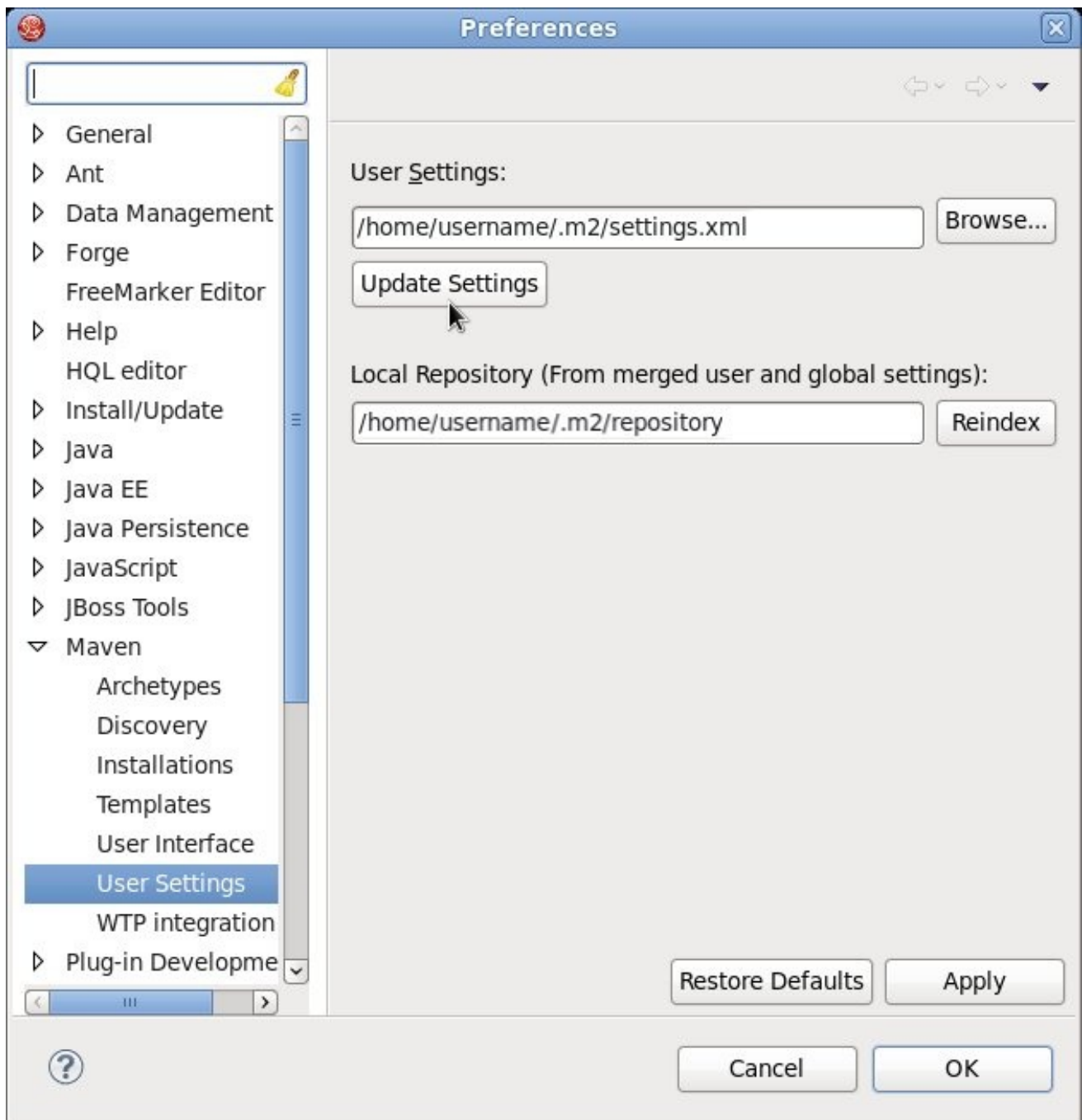
To resolve the issue, delete the cached local repository – the `~/.m2/repository/` directory on Linux or the `%SystemDrive%\Users\USERNAME\.m2\repository\` directory on Windows – and run **`mvn clean install -U`**. This will force Maven to download correct versions of required artifacts when performing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

6.3.1.2. Why is Red Hat JBoss Developer Studio using my old Maven configuration?

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

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

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 Red Hat JBoss Developer Studio.



6.4. CONFIGURING MAVEN TO USE ONLINE REPOSITORIES

The online repositories required for Red Hat JBoss BPM Suite applications are located at <https://maven.repository.redhat.com/ga/>.

It is possible to configure Maven to use online repositories using the project's POM file, but this is not recommended.

Procedure: Configuring Maven to Use 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>https://maven.repository.redhat.com/ga/</url>
        <releases>
          <enabled>>true</enabled>
        </releases>
        <snapshots>
          <enabled>>false</enabled>
        </snapshots>
      </repository>
    </repositories>
    <pluginRepositories>
      <pluginRepository>
        <id>jboss-ga-plugin-repository</id>
        <url>https://maven.repository.redhat.com/ga/</url>
        <releases>
          <enabled>>true</enabled>
        </releases>
        <snapshots>
          <enabled>>false</enabled>
        </snapshots>
      </pluginRepository>
    </pluginRepositories>
  </profile>
</profiles>

<servers>
  <!-- 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>
-->
</servers>

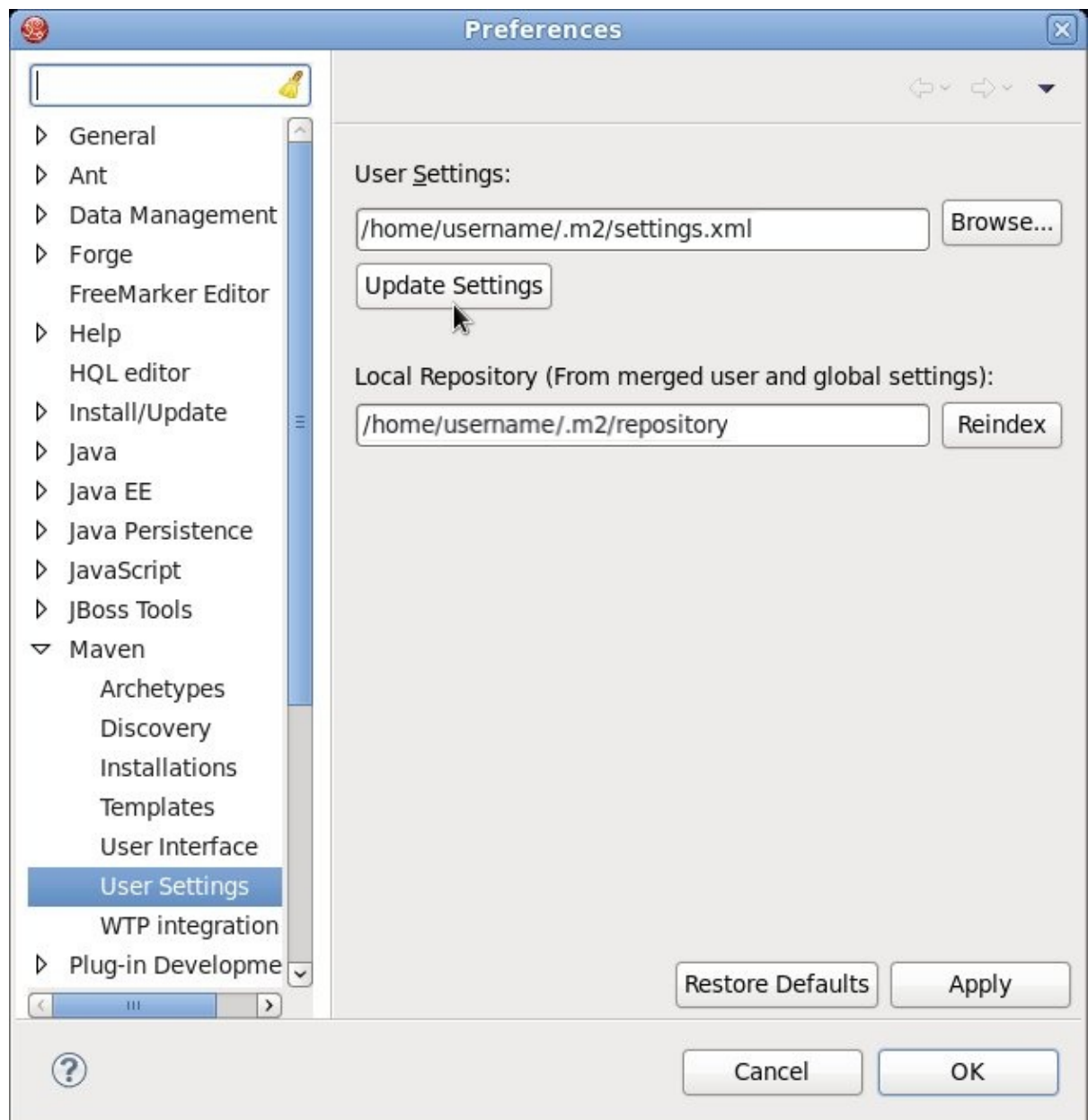
<activeProfiles>
  <!-- Activation of the BRMS/BPMS profile -->
  <activeProfile>brms-bpms-online-profile</activeProfile>
</activeProfiles>

</settings>

```

- If you modified the **settings.xml** file while Red Hat 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 Red Hat JBoss Developer Studio.

Figure 6.1. Update Maven User Settings



Result

Maven has been configured to use the online repositories provided for your Red Hat JBoss product.



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 – and run **`mvn clean install -U`**. This will force Maven to download correct versions of required artifacts when performing the next build.

6.5. DEPENDENCY MANAGEMENT

In order to use the correct Maven dependencies in your Red Hat JBoss BPM Suite project, you must add relevant Bill Of Materials (BOM) files to the project's `pom.xml` file. Adding the BOM files ensures that the correct versions of transitive dependencies from the provided Maven repositories are included in the project.

To use the Red Hat JBoss BPM Suite Maven artifacts, you need to configure one of the following:

- Red Hat JBoss BPM Suite product online Maven repository – for instructions, see [Section 6.4, “Configuring Maven to Use Online Repositories”](#).
- Red Hat JBoss BPM Suite product local-filesystem-based Maven repository – for instructions, see [Section 6.3, “Configuring Maven to Use File System Repositories”](#).

The Maven Central online repository, which is also required, is configured by default in Maven. It has to be reachable for your project to function properly, even if you selected the local Maven repository.

Depending on your project requirements, declare one of the following the dependencies in your POM file in the dependencies section. For further information about the versions, see [Section 1.3, “Supported Component Versions”](#).

- **`org.jboss.bom.brms:jboss-brms-bpmsuite-platform-bom:$VERSION`**
- **`org.jboss.bom.brms:jboss-brms-bpmsuite-bom:$VERSION`**

CHAPTER 7. RED HAT JBOSS DEVELOPER STUDIO

7.1. RED HAT JBOSS DEVELOPER STUDIO

Red Hat JBoss Developer Studio is the JBoss Integrated Development Environment (IDE) based on Eclipse. Get the latest Red Hat JBoss Developer Studio from the [Red Hat Customer Portal](#). Red Hat 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 Red Hat JBoss BRMS plug-in is called the Drools plug-in and the Red Hat JBoss BPM Suite plug-in is called the jBPM plug-in.

See the *Red Hat JBoss Developer Studio* documentation for installation and setup instructions.



WARNING

Due to an issue in the way multi-byte rule names are handled, you must ensure that the instance of JBoss Developer Studio 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"`.

7.2. INSTALLING THE RED HAT JBOSS DEVELOPER STUDIO PLUG-INS

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

Procedure: Installing the Drools and jBPM Red Hat JBoss Developer Studio Plug-in

1. Start Red Hat 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.redhat.com/9.0/stable/updates/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. Once the plug-in installation is complete, restart Red Hat JBoss Developer Studio.

7.3. SETTING THE DROOLS AND JBPM RUNTIME ENVIRONMENTS

To use the Red Hat JBoss BRMS and Red Hat JBoss BPM Suite plug-ins with Red Hat JBoss Developer Studio, you must set up the runtimes.

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

Procedure: Configuring Red Hat JBoss BRMS and Red Hat JBoss BPM Suite Runtimes

1. Extract the runtime JAR files located in the **jboss-brms-*VERSION*-engine.zip** or **jboss-bpmsuite-*VERSION*-engine.zip** archive that you can download from the [Red Hat Customer Portal](#).
2. From the Red Hat JBoss Developer Studio menu, select **Window** and click **Preferences**.
3. To install the Drools runtime, select **Drools** → **Installed Drools Runtimes**.
To install the jBPM runtime, select **jBPM** → **Installed jBPM Runtimes**.
4. Click **Add...**, provide a name and a version of the new runtime, and click **Browse** to navigate to the directory where you extracted the runtime files in the first step. Click **OK** to register the selected runtime in Red Hat JBoss Developer Studio.
5. Mark the runtime you have created as the default 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 Red Hat JBoss Developer Studio to update the runtime.

7.4. CONFIGURING RED HAT JBOSS BPM SUITE SERVER

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

Procedure: Configuring Server

1. Open the jBPM view by selecting **Window** → **Open Perspective** → **Other**. 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 6.4+** and click **Next**.
5. Set the home directory by clicking the **Browse** button. Navigate to and select the installation directory for JBoss EAP 6.4 which has Red Hat 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**.

7.5. IMPORTING PROJECTS FROM GIT REPOSITORY INTO RED HAT JBOSS DEVELOPER STUDIO

You can configure Red Hat JBoss Developer Studio to connect to a central Git asset repository. The repository stores rules, models, functions, and processes.

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

Procedure: Cloning Remote Git Repository

1. Start the Red Hat JBoss BPM Suite server by selecting the server from the **Servers** 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 fails. In that case, you may safely ignore this step.

```
■ /sbin/service sshd start
```

3. In Red Hat 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**.
6. Select the branch you wish to import in the following window and click **Next**.
7. 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: Importing Local Git Repository

1. Start the Red Hat JBoss BPM Suite server by selecting the server from the **Servers** tab and click the start icon.
2. In Red Hat 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**.
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** and click **Next**. Name the project and click **Finish**.

CHAPTER 8. PATCHING AND UPGRADING RED HAT JBOSS BPM SUITE

8.1. ABOUT PATCHES AND UPGRADES

Red Hat JBoss BPM Suite patches can be either an asynchronous update, or a planned update:

- *Asynchronous updates*: Individual patches which are released outside the normal update cycle of the existing product. These may include security patches, as well as other individual patches provided by Red Hat Global Support Services (GSS) to fix specific issues.
- *Planned updates*: The cumulative patches of an existing product, which includes all previously developed updates for that version of the product.

To download Red Hat JBoss BPM Suite patches:

1. Navigate to the [Software Downloads](#) section of the Customer Portal.
2. Click **Security Advisories**.

The following files are included as part of a Red Hat JBoss BRMS and Red Hat JBoss BPM Suite patch release.

- Red Hat JBoss BRMS customers – **jboss-brms-*VERSION*-patch.zip**.
- Red Hat JBoss BPM Suite customers – **jboss-bpmsuite-*VERSION*-patch.zip**.
- Maven repository updates (same for both Red Hat JBoss BRMS and Red Hat JBoss BPM Suite customers) – **jboss-brms-bpmsuite-*VERSION*-incremental-maven-repository.zip**.

8.2. APPLYING PATCHES IN RED HAT JBOSS BPM SUITE 6.4



IMPORTANT

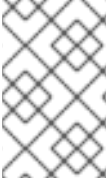
6.4 Update 6 introduces a small change into the database schema. You must apply the **bpms-6.4-to-7.0.sql** script to your database before you run Red Hat JBoss BPM Suite or Red Hat JBoss BRMS 6.4.6. This script is located in the **upgrade-scripts/<database-type>** directory, available from the Red Hat JBOSS BPM Suite 6.4 Update 6 and the Red Hat JBOSS BRMS 6.4 Update 6 zip files which you can download from the Red Hat Customer Portal.

In Red Hat JBoss BPM Suite, the client patching tool is distributed as a ZIP file that includes **.sh** and **.bat** scripts, allowing for easy and automatic application of updates to an existing Red Hat JBoss BPM Suite 6.1 (or better) installation.



IMPORTANT

The patching tool is for use with Red Hat JBoss BPM Suite 6.1 or better, and should not be used for earlier versions. For more information, see the [Maintenance Release Changes in BRMS and BPM Suite 6.1+](#) article at Red Hat Knowledgebase.

**NOTE**

Patch updates should not be applied while you are running an instance of Red Hat JBoss BPM Suite. Make sure that the server is shut down before running the following command.

The script requires two mandatory parameters: **<path-to-distribution-root>** and **<type-of-distribution>**. For example, the following command applies the updates to the specified Red Hat JBoss EAP 6.4 bundle:

```
$. /apply-updates.sh ~/EAP_HOME/jboss-eap-6.4 eap6.x
```

The following command applies the updates to the specified Red Hat JBoss EAP 7.0 bundle:

```
$. /apply-updates.sh ~/EAP_HOME/jboss-eap-7.0 eap7.x
```

The following distribution types are supported:

- **eap6.x**
- **eap6.x-bc**
- **eap6.x-dashbuilder**
- **eap6.x-kie-server**
- **eap7.x**
- **eap7.x-bc**
- **eap7.x-dashbuilder**
- **eap7.x-kie-server**
- **generic**
- **generic-bc**
- **generic-dashbuilder**
- **generic-kie-server**
- **was8**
- **was8-bc**
- **was8-dashbuilder**
- **was8-kie-server**
- **wls12c**
- **wls12c-bc**
- **wls12c-dashbuilder**

- **wls12c-kie-server**
- **bpmsuite-engine**
- **planner-engine**
- **supplementary-tools**

The quickstarts and migration tool are also included in the patch and are available for download as a ZIP file.



NOTE

Only updates for Red Hat JBoss BRMS or Red Hat JBoss BPM Suite are included in the patch distribution. Patches to EAP itself must be applied using the EAP patching mechanism. See the [Red Hat JBoss EAP Installation Guide](#).

Backup Feature

Before applying any updates, the client script takes a backup of the specified distribution. It copies the distribution file or directory into the **backup/CURRENT_TIMESTAMP** subdirectory. The top-level backup directory is created at the same filesystem level as the **apply-updates** script.

Blacklist Feature

The client patching tool provides a blacklist feature that allows you to tell the script the files that must not be updated. This is a feature that helps you preserve your configuration files from being overwritten automatically by the update process. You can specify non-configuration files as well if required.

To specify the blacklisted files, open the file **blacklist.txt** present within the patch distribution. Enter the relative path to the files that must not be updated. Each file must be specified on a line by itself.

```
# Lines with a '#' are comment lines, like this one.
# Blank lines are ignored.

# We have made changes to the web.xml that must be preserved:
WEB-INF/web.xml

# This file has custom modifications:
styles/base.css
```

Files specified in the **blacklist.txt** file that have updated content in the patch, are not touched by the update tool. Instead, the tool copies the new, updated file in the same location and appends the new suffix to it. For example, after running the patch tool, both these files will exist in the **styles** folder, continuing with the **blacklist.txt** file in the example above.

```
$ ls styles
base.css base.css.new
```

Now, compare the contents of the two files and merge the changes.

If there are files that are no longer being distributed but you want to preserve them, put them into the **blacklist.txt** file as well. The patch update tool will not delete these files, and instead create an empty marker file with the suffix **removed**. You can then choose to either keep or delete these files manually.

Continuing with the previous example, if the **base.css** file was removed and you had this file listed in the **blacklist.txt** file, then after the patch tool has run, the contents of the styles directory would be similar to:

```
$ ls styles
base.css base.css.removed
```

8.3. PATCHING OTHER PLATFORMS AND APPLICATIONS

Use the following commands for updating other supported platforms and common applications in Red Hat JBoss BPM Suite.



IMPORTANT

On a Microsoft Windows system, run **./apply-updates.bat** instead of **./apply-updates.sh**.

Patch EAP 6.x Business Central WAR

```
$ ./apply-updates.sh PATH/jboss-eap-6.4/standalone/deployments/business-central.war eap6.x-bc
```

Patch EAP 7.x Business Central WAR

```
$ ./apply-updates.sh PATH/jboss-eap-7.0/standalone/deployments/business-central.war eap7.x-bc
```

Patch EAP 6.x KIE Server WAR

```
$ ./apply-updates.sh PATH/jboss-eap-6.4/standalone/deployments/kie-server.war eap6.x-kie-server
```

Patch EAP 7.x KIE Server WAR

```
$ ./apply-updates.sh PATH/jboss-eap-7.0/standalone/deployments/kie-server.war eap7.x-kie-server
```

Patch Generic KIE Server WAR

```
$ ./apply-updates.sh PATH_TO_TOMCAT_HOME/webapps/kie-server.war generic-kie-server
```

Patch Whole WebLogic 12c Bundle

```
$ ./apply-updates.sh PATH_TO_UNZIPPED_wlsc12c_BUNDLE wls12c
```

Patch Planner Engine Bundle

```
$ ./apply-updates.sh PATH_TO_UNZIPPED_PLANNER_BUNDLE planner-engine
```

Patch IBM WebSphere Application Server Bundle

```
$ ./apply-updates.sh PATH_TO_UNZIPPED_WAS_BUNDLE was8
```


**NOTE**

When patching the IBM WebSphere Application Server, do *not* extract the target WAR files.

See [Section 8.2, “Applying Patches in Red Hat JBoss BPM Suite 6.4”](#) and [Section 8.4, “Upgrading to Latest Minor Release”](#) for more information.

8.4. UPGRADING TO LATEST MINOR RELEASE

Apart from supporting upgrade to the latest micro release, Red Hat JBoss BPM Suite also supports upgrading between minor releases. For example, upgrading from:

- Red Hat JBoss BPM Suite 6.2.2 to Red Hat JBoss BPM Suite 6.3.0
- Red Hat JBoss BPM Suite 6.1.5 to Red Hat JBoss BPM Suite 6.3.0

The Red Hat JBoss BPM Suite upgrade tool is distributed as ZIP files with naming convention that states the upgrade path. For example, **jboss-bpmsuite-6.2.2-to-6.3.0-patch.zip** is used to upgrade from 6.2.x to the 6.3.0 version. These ZIP files can be downloaded from the [Red Hat Customer Portal](#):

- Use **jboss-bpmsuite-6.2.2-to-6.3.0-patch.zip** to upgrade from Red Hat JBoss BPM Suite 6.2.2 to Red Hat JBoss BPM Suite 6.3.0.
- Use **jboss-bpmsuite-6.1.5-to-6.3.0-patch.zip** to upgrade from Red Hat JBoss BPM Suite 6.1.5 to Red Hat JBoss BPM Suite 6.3.0.

Each ZIP file contains the following scripts:

- **apply-updates.bat**
- **apply-updates.sh**

To upgrade to the next minor release using these upgrade scripts, you must specify arguments indicating the path of distribution and the type of distribution you want to upgrade in your command:

```
$ ./apply-updates.sh DISTRIBUTION_PATH DISTRIBUTION_NAME
```

For example:

```
$ ./apply-updates.sh ~/EAP_HOME/jboss-eap-6.4 eap6.x
```

The supported distribution types are:

- **eap6.x**
- **eap6.x-bc**
- **eap6.x-dashbuilder**
- **eap6.x-kie-server**
- **generic**
- **generic-bc**

- **generic-dashbuilder**
- **generic-kie-server**
- **was8**
- **was8-bc**
- **was8-dashbuilder**
- **was8-kie-server**
- **wls12c**
- **wls12c-bc**
- **wls12c-dashbuilder**
- **wls12c-kie-server**
- **bpmsuite-engine**
- **planner-engine**
- **supplementary-tools**

The upgrade tool allows you to upgrade the entire distribution, or only a part of the distribution as per your requirement. For example, for the **eap6.x** distribution, you can choose to patch the entire **eap6.x** or choose to patch any of the war files (**eap6.x-bc**, **eap6.x-dashbuilder**, **eap6.x-kie-server**) that the patch contains.

Note that the upgrade tool does not upgrade the configuration files if you have your custom updates in them. The upgrade tool checks if the configuration files have any changes. If there are no changes made to the configuration files, the tool replaces the configuration files with the latest version. However, if the tool finds custom changes made to any of the configuration files, it adds those files to blacklist, and does not replace them with the latest version. So you do not need to manually compare the configuration files and place them in the blacklist to ensure that your custom configurations are intact.



NOTE

It is recommended that you add your custom changes to the **.new** files instead of trying to update the current configuration files with changes from the new Red Hat JBoss BPM Suite version. For example, if you have custom changes such as data source name/location in the **persistence.xml** file, the recommended approach is to add your custom changes to the **.new** files created by the upgrade tool. Once you have updated the **.new** files with all the required changes, rename them to their original names (without the **.new** suffix). This ensures that the applications pick the updated configuration files containing your custom changes.

APPENDIX A. VERSIONING INFORMATION

Documentation last updated on: Wednesday, Oct 23, 2019.