



# Red Hat JBoss BPM Suite 6.4

## IBM WebSphere Installation and Configuration Guide

For Red Hat JBoss BPM Suite



# Red Hat JBoss BPM Suite 6.4 IBM WebSphere Installation and Configuration Guide

---

For Red Hat JBoss BPM Suite

Red Customer Content Services

[brms-docs@redhat.com](mailto:brms-docs@redhat.com)

Emily Murphy

Gemma Sheldon

Michele Haglund

Mikhail Ramendik

Stetson Robinson

Vidya Iyengar

## Legal Notice

Copyright © 2019 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, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

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

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

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

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

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

The OpenStack<sup>®</sup> 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

A guide to installing and configuring Red Hat JBoss BPM Suite on IBM WebSphere Application Server.

## Table of Contents

<b>CHAPTER 1. INTRODUCTION</b> .....	<b>3</b>
1.1. ABOUT RED HAT JBOSS BPM SUITE	3
1.2. SUPPORTED PLATFORMS	3
1.3. USE CASE: PROCESS-BASED SOLUTIONS IN LOAN INDUSTRY	3
1.4. ABOUT IBM WEBSHERE APPLICATION SERVER	4
1.4.1. Getting Started with IBM WebSphere Application Server	4
Downloading and Installing WAS	4
Creating Users and Installation Verification	5
Starting Server	5
1.5. ABOUT RED HAT JBOSS BPM SUITE FOR IBM WEBSHERE APPLICATION SERVER	6
<b>CHAPTER 2. DOWNLOAD AND EXTRACT</b> .....	<b>7</b>
2.1. DOWNLOADING RED HAT JBOSS BPM SUITE FOR IBM WEBSHERE APPLICATION SERVER	7
2.2. EXTRACTING RED HAT JBOSS BPM SUITE FOR IBM WEBSHERE APPLICATION SERVER	7
<b>CHAPTER 3. CONFIGURE</b> .....	<b>8</b>
3.1. INCREASING JVM HEAP SIZE	8
3.2. MODIFYING SECURITY SETTINGS	12
3.3. CREATING USERS AND GROUPS	12
3.4. SESSION MANAGEMENT CUSTOM SETTINGS	14
3.5. SETTING UP DATA SOURCE	14
Creating JDBC Providers	15
Setting up Data Source	16
3.6. SETTING UP JMS RESOURCES	19
Setting up Buses	19
Creating Service Bus	19
Adding Bus Member	20
Creating JMS Connection Factories	20
Creating JMS Queues	21
Creating JMS Activation Specifications	21
3.7. ADDING CUSTOM JVM PROPERTIES	22
3.8. CONFIGURING LDAP PRINCIPAL AND ROLE NAMES MATCHING CRITERIA	24
3.9. CONFIGURING UNIFIED EXECUTION SERVERS	25
<b>CHAPTER 4. INSTALL</b> .....	<b>26</b>
4.1. INSTALLING BUSINESS CENTRAL	26
Mapping Groups to Roles	29
Class-Loading Configuration	30
Enabling Bouncy Castle Crypto API	31
4.2. INSTALLING DASHBUILDER	32
Mapping Groups to Roles	34
Class-Loading Configuration	35
4.3. INSTALLING INTELLIGENT PROCESS SERVER	36
Class-Loading Configuration	36
Mapping Groups to Roles	37
<b>APPENDIX A. VERSIONING INFORMATION</b> .....	<b>38</b>



# 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 fully supported and tested on the following platforms:

- Red Hat JBoss Enterprise Application Platform 6.4.7
- Red Hat JBoss Web Server 2.1, 3.0
- IBM WebSphere Application Server 8.5.5
- Oracle WebLogic Server 12.1.3 (12c)
- Red Hat JBoss Fuse 6.2.x

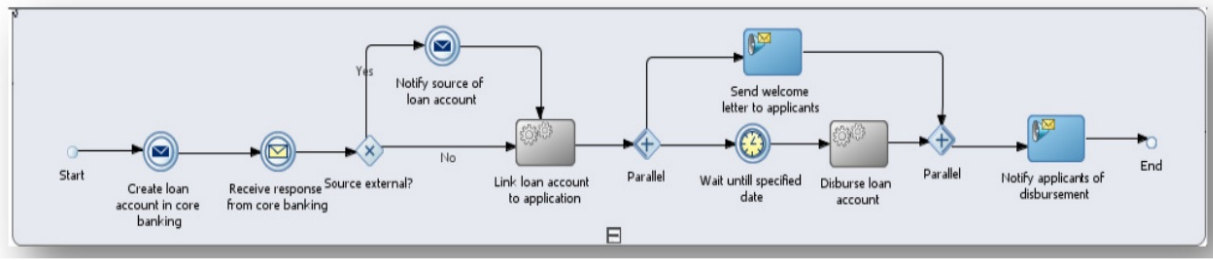
## 1.3. USE CASE: PROCESS-BASED SOLUTIONS IN LOAN INDUSTRY

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:

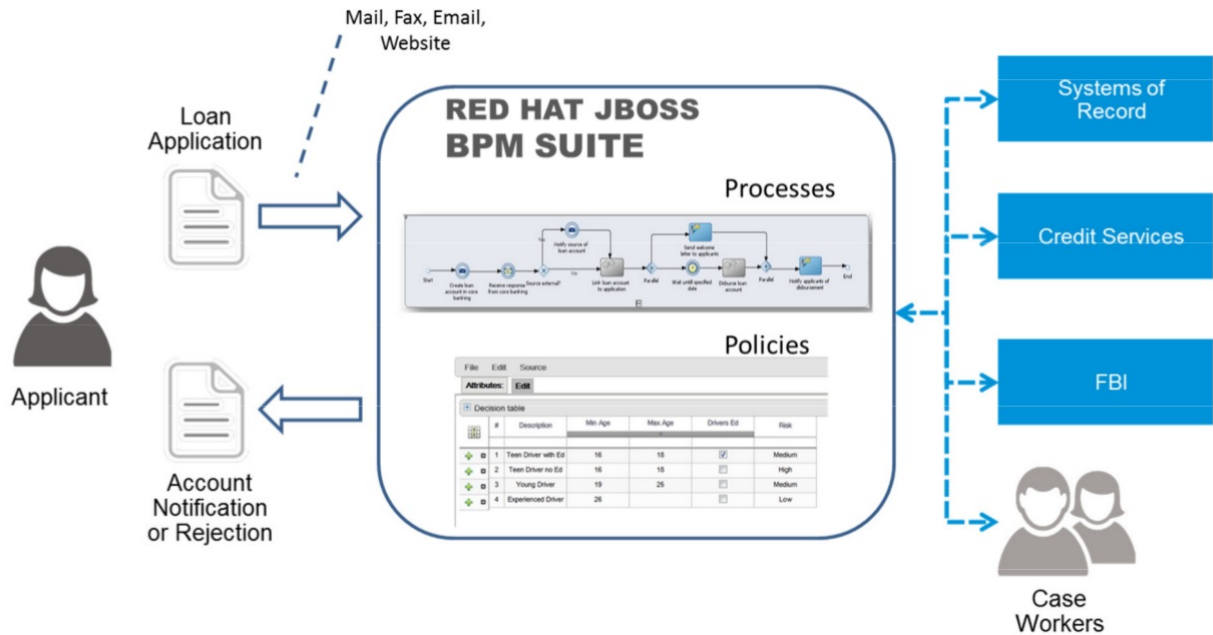
Figure 1.1. High-Level Loan Application Process Flow



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.

## 1.4. ABOUT IBM WEBSHERE APPLICATION SERVER

IBM WebSphere Application Server (hereinafter referred to as WAS) is a flexible and secure web application server that hosts Java-based web applications and provides Java EE-certified runtime environments. WAS 8.5.5 supports Java SE 8 and is fully compliant with Java EE 7 since version 8.5.5.6.

### 1.4.1. Getting Started with IBM WebSphere Application Server

#### Downloading and Installing WAS

In order to install IBM WebSphere Application Server, you need to download and install IBM Installation Manager first.



1. Download IBM Installation Manager version 1.8.5 or later from the [IBM Installation Manager and Packaging Utility download links](#) page.
2. Extract the downloaded archive, change to root, and run the following command in the new directory:

```
./install
```

**IBM Installation Manager** opens. The installer will guide you through the entire process of installing the manager.

3. Open the installed manager, go to **File → Preferences** and click **Add Repository**. The **Add Repository** dialog window opens.
4. Enter the repository URL for IBM WebSphere Application Server 8.5. You can find all the repository URLs in the [Online product repositories for WebSphere Application Server offerings](#) page of the *IBM Knowledge Center*. For example:

```
http://www.ibm.com/software/repositorymanager/com.ibm.websphere.APPCLIENT.v85
```

5. Enter your IBM id credentials when prompted and after the **Connection** status turns green, click **OK**.
6. Click **Install**.
7. Choose the packages you want to install and click **Next**. If asked, install all the recommended fixes as well.

### Creating Users and Installation Verification

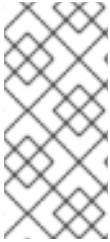
1. In the **WebSphere Customization Toolbox 8.5**, open the **Profile Management Tool**.
2. Click **Create...** and create a user for the **Application Server** environment.
3. In the **WebSphere Application Server - First Steps** window that opens, click **Installation Verification** and verify that your server was installed properly.

### Starting Server

1. Change into the **bin** directory of the installed application server (by default at **/opt/IBM/WebSphere/AppServer**).
2. Change to root and run **./startServer.sh APPLICATION\_SERVER\_NAME**, for example:

```
./startServer.sh server1
```

3. Navigate to **http://TARGET\_SERVER:9060/ibm/console** in your web browser and log in with the user credentials created in the previous procedure. The **Integrated Solutions Console** opens.

**NOTE**

Do not forget to stop the server after you are no longer using it. Log out of the console and run `./stopServer.sh APPLICATION_SERVER_NAME` as root. For example:

```
./stopServer.sh server1
```

For further information, see [WebSphere Application Server, version 8.5.5 documentation](#).

## 1.5. ABOUT RED HAT JBOSS BPM SUITE FOR IBM WEBSHERE APPLICATION SERVER

Red Hat JBoss BPM Suite for IBM WebSphere Application Server is provided as three deployable web application archives: **business-central.war**, **kie-server.war**, and **dashbuilder.war**. It is then deployed and configured as any other web application.

**NOTE**

Red Hat JBoss BPM Suite 6.4 is supported on the version 8.5.5 of IBM WebSphere Application Server.

Installation of Red Hat JBoss BPM Suite on IBM WebSphere Application Server is supported since version 6.0.2 of Red Hat JBoss BPM Suite. This guide covers the installation and configuration of Red Hat JBoss BPM Suite on a full profile version of IBM WebSphere Application Server 8.5.5.

Before installation, several configuration steps need to be performed to enable a successful setup. Follow the procedures in this guide to configure the server.

Before you proceed, ensure you have root access to IBM WebSphere Application Server and that you are able to successfully access the IBM WebSphere's administrative console using a web browser (usually at [http://TARGET\\_SERVER:9060/ibm/console](http://TARGET_SERVER:9060/ibm/console)).

## CHAPTER 2. DOWNLOAD AND EXTRACT

Follow the steps outlined in this chapter to download and extract Red Hat JBoss BPM Suite for IBM WebSphere Application Server.

### 2.1. DOWNLOADING RED HAT JBOSS BPM SUITE FOR IBM WEBSHERE APPLICATION SERVER

To download the deployable Red Hat JBoss BPM Suite package file for IBM WebSphere Application Server from the Red Hat Customer Portal:

1. Go to the [Red Hat Customer Portal](#) and log in.
2. Click **DOWNLOADS** at the top of the page.
3. In the **Product Downloads** page that opens, click **Red Hat JBoss BPM Suite**
4. From the **Version** drop-down menu, select **6.4**.
5. Navigate to **Red Hat JBoss BPM Suite 6.4.0 Deployable for WebSphere 8.5** and click **Download**.

### 2.2. EXTRACTING RED HAT JBOSS BPM SUITE FOR IBM WEBSHERE APPLICATION SERVER

The downloaded installation ZIP file for Red Hat JBoss BPM Suite (**jboss-bpmsuite-6.4.0.GA-deployable-was8.zip**) contains the Business Central WAR deployable archive (**business-central.war**), the Intelligent Process Server WAR deployable archive (**kie-server.war**), and the Dashbuilder WAR deployable archive (**dashbuilder.war**) in an unextracted format.

Extract the downloaded ZIP file so that you have access to the deployable WAR files:

```
unzip jboss-bpmsuite-VERSION-deployable-was8.zip
```

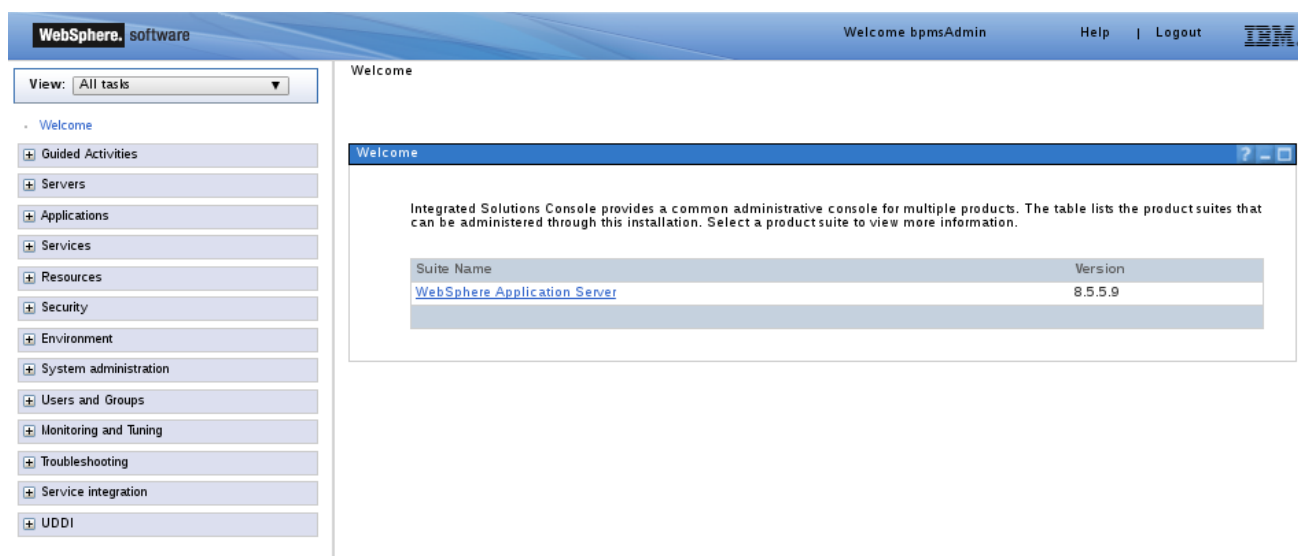
## CHAPTER 3. CONFIGURE

Before you can deploy Red Hat JBoss BPM Suite as a web archive on IBM WebSphere Application Server, configure the server to accept the deployable WAR files. Follow the steps outlined in this section to deploy Red Hat JBoss BPM Suite on IBM WebSphere Application Server.

Log in to your IBM WebSphere console using an administrative login before performing any of these steps. The usual login URL is [http://TARGET\\_SERVER:9060/ibm/console](http://TARGET_SERVER:9060/ibm/console) (for example <http://localhost:9060/ibm/console>).

The **IBM Integrated Solutions Console** with the welcome screen opens. The main menu on the left side of the console contains all the links necessary for setting the application server.

Figure 3.1. IBM Integrated Solutions Console



### 3.1. INCREASING JVM HEAP SIZE

With the default JVM heap size, the IBM WebSphere Application Server freezes or causes deployment errors when deploying Business Central. To increase the heap size:

1. In the **Integrated Solutions Console**, go to **Servers** → **Server Types** → **WebSphere Application Servers**.
2. In the list of application servers, click on the server on which you are going to deploy Business Central. For example **server1**.  
The configuration page for that server opens.
3. Under **Server Infrastructure** heading on the right side, click **Java and Process Management** → **Process Definition**.

Figure 3.2. Application Server Configuration Page

Application servers

[Application servers](#) > **server1**

Use this page to configure an application server. An application server is a server that provides services required to run enterprise applications.

**Runtime** | Configuration

---

**General Properties**

Name

Node name

Run in development mode

Parallel start

Start components as needed

Access to internal server classes

**Server-specific Application Settings**

ClassLoader policy

Class loading mode

**Container Settings**

- [Session management](#)
- ▣ SIP Container Settings
- ▣ Web Container Settings
- ▣ Portlet Container Settings
- ▣ EJB Container Settings
- ▣ Container Services
- ▣ Business Process Services

**Applications**

- [Installed applications](#)

**Server messaging**

- [Messaging engines](#)
- [Messaging engine inbound transports](#)
- [WebSphere MQ link inbound transports](#)
- [SIB service](#)

**Server Infrastructure**

- ▣ Java and Process Management
  - [Class loader](#)
  - [Process definition](#)
  - [Process execution](#)
- ▣ Administration
  - [Java SDKs](#)

4. Click **Java Virtual Machine** under the **Additional Properties** heading on the right.

Figure 3.3. Process Definition Configuration Page

Application servers

[Application servers](#) > [server1](#) > [Process definition](#)

Use this page to configure a process definition. A process definition defines the command line information necessary to start or initialize a process.

Configuration

General Properties	Additional Properties
Executable name <input type="text"/>	<ul style="list-style-type: none"><li>Java Virtual Machine</li><li>Environment Entries</li><li>Process execution</li><li>Process Logs</li><li>Logging and tracing</li></ul>
Executable arguments <input type="text"/>	
Start command <input type="text"/>	
Start command arguments <input type="text"/>	
Stop command <input type="text"/>	
Stop command arguments <input type="text"/>	
Working directory <input type="text" value="\${USER_INSTALL_ROOT}"/>	
Executable target type JAVA_CLASS	
Executable target <input type="text" value="com.ibm.ws.runtime.WsServer"/>	

Apply OK Reset Cancel

This will open up the configuration properties for the JVM that is used to start the server.

5. Change both the **Initial Heap Size** and **Maximum Heap Size** to **2048**. This is the configuration Red Hat JBoss BPM Suite is tested with.

Figure 3.4. JVM Configuration Page

Application servers > server1 > Process definition > Java Virtual Machine

Use this page to configure advanced Java(TM) virtual machine settings.

Configuration **Runtime**

**General Properties** **Additional Properties**

Classpath

Boot Classpath

Verbose class loading

Verbose garbage collection

Verbose JNI

Initial heap size  
2048 MB

Maximum heap size  
2048 MB

Run HProf

HProf Arguments

Debug Mode

Debug arguments  
-agentlib:jdwp=transport=dt\_socket,server=y,suspend=n,address=7777

Generic JVM arguments

Executable JAR file name

Disable JIT

Operating system name  
linux

Apply OK Reset Cancel

- Click **Apply** at the bottom.

**Messages** pop-up window appears at the top of the **Application Servers** configuration page. You can choose to save these configuration settings to the master configuration at this stage.

Figure 3.5. Messages Pop-up

**Messages**

⚠ Changes have been made to your local configuration. You can:

- [Save](#) directly to the master configuration.
- [Review](#) changes before saving or discarding.

⚠ The server may need to be restarted for these changes to take effect.

- Restart the server at this point or wait till other configuration changes have been made.

## 3.2. MODIFYING SECURITY SETTINGS

For the Business Central application to work, you need to modify several security settings on IBM WebSphere Application Server. To enable the container-managed authentication mechanisms provided by the server:

- In the main menu, click **Security** → **Global Security**. Ensure that the option **Enable Application Security** is checked. This may already be checked and overridden at the server level.

Figure 3.6. Global Security Configuration Page

The screenshot shows the 'Global security' configuration page. It includes sections for Administrative security, Application security, Java 2 security, User account repository, and Authentication. The 'Application security' section has 'Enable application security' checked. The 'Authentication' section has 'LTPA' selected. The 'Custom properties' link is highlighted with a red box.

- Click **Custom Properties** on the right side and then **New...** to enter a new custom property with the following details:
  - Name:** `com.ibm.ws.security.web.logoutOnHTTPSessionExpire`
  - Value:** `true`

This property instructs the server to invalidate LTPA tokens on session invalidation, which makes the logout process consistent across multiple users using the same browser.

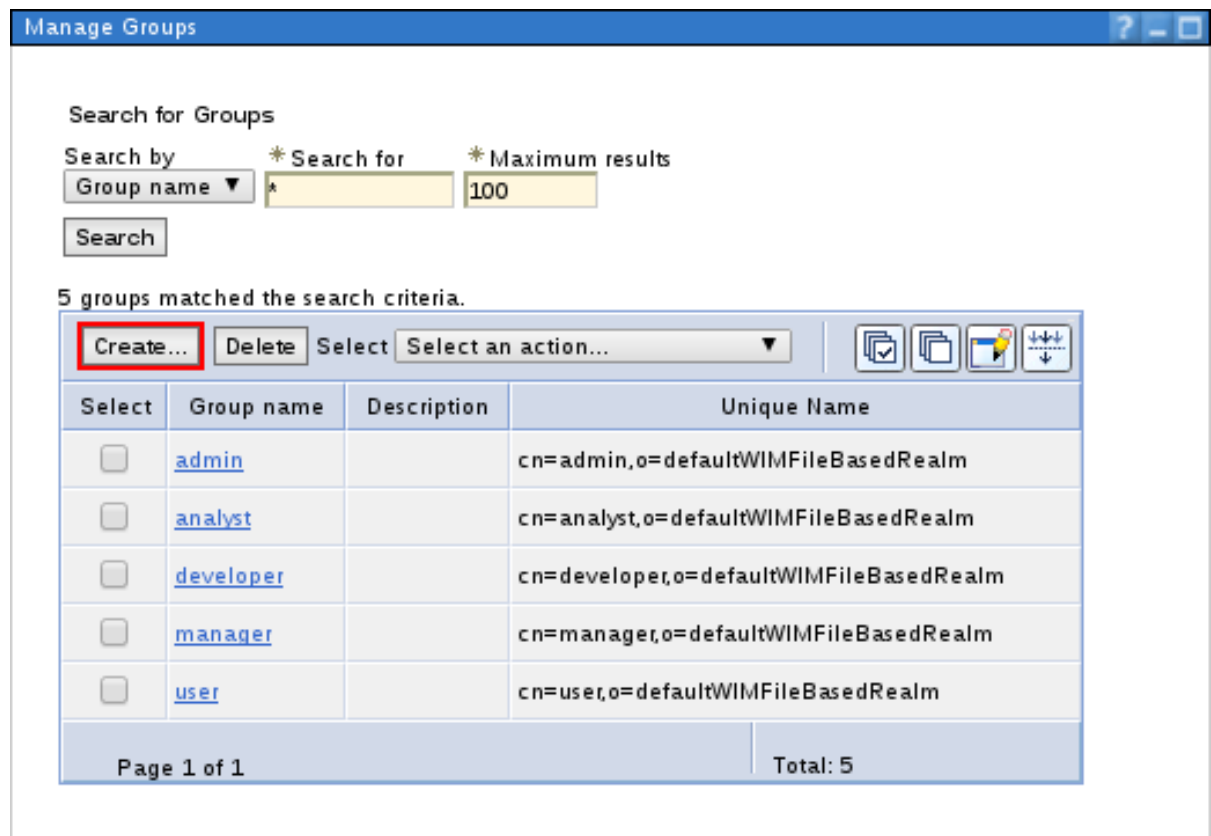
- Click **Apply** and then **OK**.

## 3.3. CREATING USERS AND GROUPS

- In the main menu on the left, click **Users and Groups** → **Manage Groups**.
- Create five new groups: **admin**, **analyst**, **developer**, **manager**, and **user** by clicking **Create...**



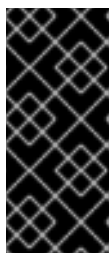
Figure 3.7. Created Groups



#### NOTE

Add the **kie-server** group as well if you are going to install the Intelligent Process Server. Also add the **rest-all** and other REST API groups if you are going to use the API. For further information about API roles, see chapter [Remote API](#) of *Red Hat JBoss BPM Suite Development Guide*.

3. In the main menu on the left, click **Users and Groups** → **Manage Users**.
4. Click **Create...** and fill in the user credentials.



#### IMPORTANT

Make sure that the selected **User ID** 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**.

Figure 3.8. Create User Dialog Window

The screenshot shows a window titled "Manage Users" with a "Create a User" form. The form includes the following fields and controls:

- \* User ID:** A text field containing "business-central-admin". To its right is a dropdown menu labeled "Group Membership", which is highlighted with a red rectangular box.
- \* First name:** A text field containing "Klara".
- \* Last name:** A text field containing "Kufova".
- E-mail:** An empty text field.
- \* Password:** A password field with masked characters (dots).
- \* Confirm password:** A password field with masked characters (dots).
- At the bottom, there are two buttons: "Create" and "Cancel".

5. Click **Group Membership** and assign the user to the **admin** group that you created previously.



#### NOTE

You may assign this user to any of the groups you have just created. In the production systems, you are likely to create separate users for separate groups that align with business roles. The **admin** group is all encompassing and is therefore useful for the purposes of this setup.

6. Click **Create**.

### 3.4. SESSION MANAGEMENT CUSTOM SETTINGS

1. In the main menu on the left, go to **Servers** → **Server Types** → **WebSphere Application Servers** and select the server on which you are deploying Business Central.
2. Click **Session Management** under the **Container Settings** heading on the right.
3. In the **Additional Properties** section on the right, click **Custom Properties** and then **New...**
4. Fill in the required information:
  - **Name:** `InvalidateOnUnauthorizedSessionRequestException`
  - **Value:** `true`
5. Click **Apply** and then **OK**.

### 3.5. SETTING UP DATA SOURCE

The Business Central application requires a data source which must be created prior to the deployment of the actual WAR file. This means that you must have access to an underlying database to which the data source connects. Whatever your underlying database, make sure you have the data source ready. Follow the steps below to set the data source.



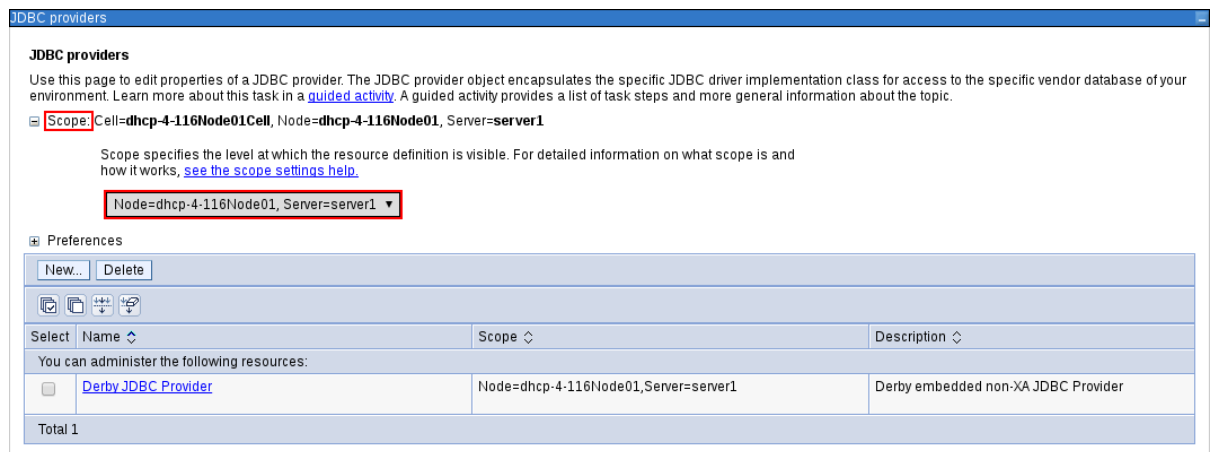
## NOTE

In the following procedure, the data source setup is demonstrated on the Oracle Database 12c.

## Creating JDBC Providers

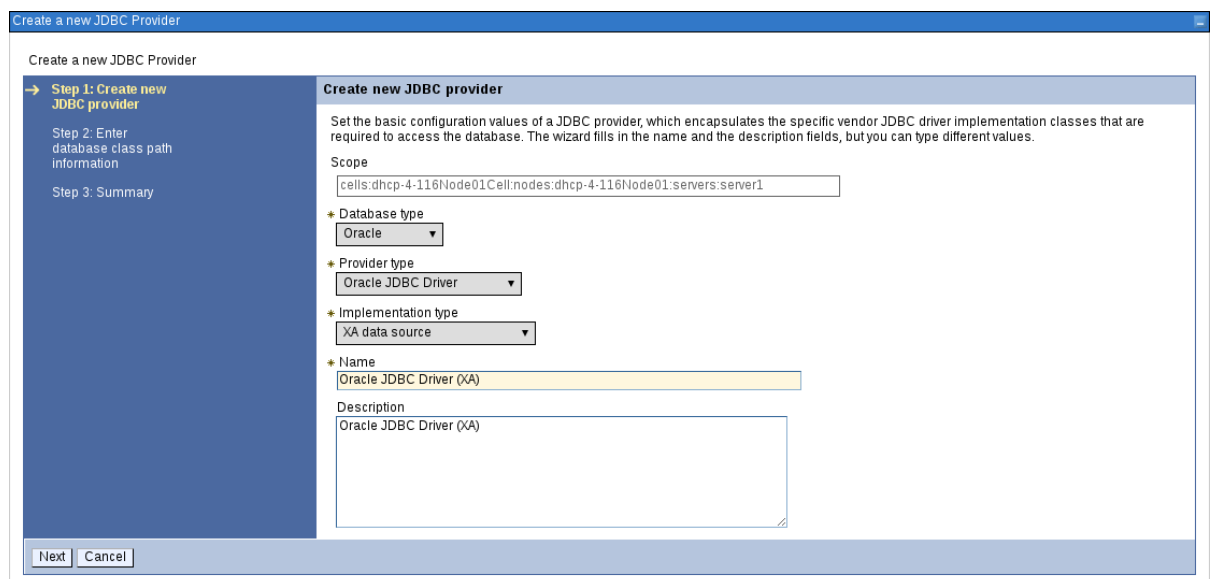
1. Open up the **JDBC Providers** page by clicking **Resources** → **JDBC** → **JDBC Providers**.
2. At the top of the **JDBC Providers** page, click **Scope**. Select the scope of this JDBC provider to include your server and node. Note that it *cannot* be **All scopes**.

Figure 3.9. Selecting Scope of JDBC Provider



3. Click **New...**.  
The **Create a New JDBC Provider** page opens.
4. Fill in the form based on the database driver that you have available.

Figure 3.10. First Step of Creating New JDBC Provider

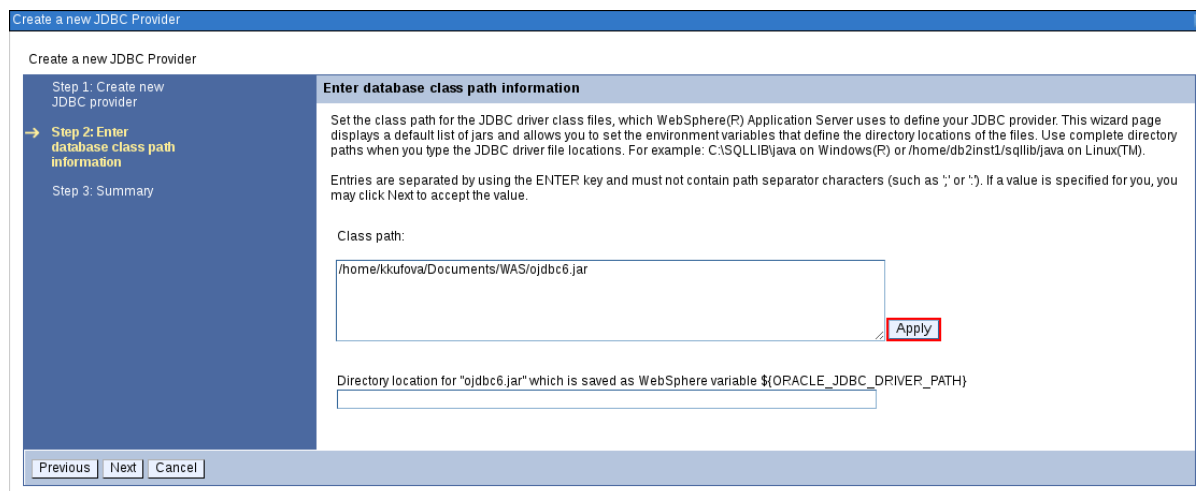


If your database is not listed, select the **User-Defined** option from the **Database Type** selection box and provide the implementation class name.

For example, for H2, PostgreSQL, or MySQL, the implementation class name will be **org.h2.jdbcx.JdbcDataSource**, **org.postgresql.xa.PGXADatasource**, and **com.mysql.jdbc.jdbc2.optional.MysqlXADataSource** respectively.

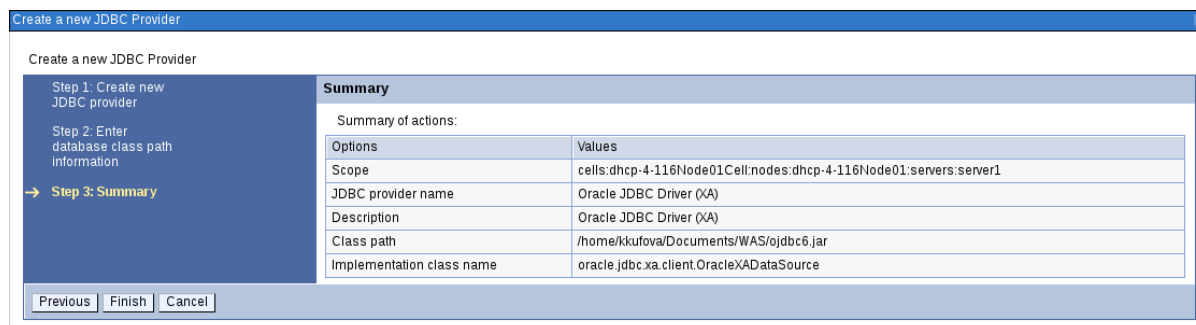
5. Give the JDBC Provider a descriptive name and click **Next**.
6. Provide the class path information for the JDBC driver class files you defined. Click **Apply**.

Figure 3.11. Defining Database Class Path



7. Click **Next**.
8. Click **Finish** to accept and add this new JDBC provider.

Figure 3.12. JDBC Provider Summary Page



Using this new JDBC provider, you will now need to set up the actual data source for Business Central.

Before you create the data source, open the **persistence.xml** file located in the **WEB-INF/classes/META-INF** directory of the Business Central WAR file (**business-central.war**) that you have downloaded. You will need to know the JNDI name of the data source defined within the **<jta-data-source>** tag. For Business Central, it is **jdbc/jbpm**.

Also change the **hibernate.dialect** property to suit your database. For example, if your underlying database is Oracle Database 12c, change the property value to **org.hibernate.dialect.Oracle10gDialect**.

## Setting up Data Source

1. Open the **Data Sources** page by clicking **Resources** → **JDBC** → **Data Sources** in the main menu on the left and make sure that the appropriate scope has been selected.

2. Click **New...**
3. Enter a unique **Data Source Name** by which you will refer to this data source and the JNDI name that you found in the **persistence.xml** file.

**Figure 3.13. First Step of Creating New Data Source**

The screenshot shows the 'Create a data source' wizard at Step 1: Enter basic data source information. The left sidebar lists five steps, with Step 1 highlighted. The main area contains the following text and fields:

**Enter basic data source information**

Set the basic configuration values of a datasource for association with your JDBC provider. A datasource supplies the physical connections between the application server and the database.

Requirement: Use the Datasources (WebSphere(R) Application Server V4) console pages if your applications are based on the Enterprise JavaBeans(TM) (EJB) 1.0 specification or the Java(TM) Servlet 2.2 specification.

Scope

+ Data source name

+ JNDI name

Buttons: **Next** | **Cancel**

Click **Next**.

4. From the **Select an Existing JDBC Provider** drop-down menu, select the JDBC provider created earlier and click **Next**.

**Figure 3.14. Selecting JDBC Provider**

The screenshot shows the 'Create a data source' wizard at Step 2: Select JDBC provider. The left sidebar lists five steps, with Step 2 highlighted. The main area contains the following text and fields:

**Select JDBC provider**

Specify a JDBC provider to support the datasource. If you choose to create a new JDBC provider, it will be created at the same scope as the datasource. If you are selecting an existing JDBC provider, only those providers at the current scope are available from the list.

Create new JDBC provider

Select an existing JDBC provider

Buttons: **Previous** | **Next** | **Cancel**

5. In the **Enter Database Specific Properties for the Data Source** step, enter the database JDBC URL and click **Next**.

**Figure 3.15. Enter Database Specific Properties for Data Source Screen**

The screenshot shows the 'Create a data source' wizard at Step 3: Enter database specific properties for the data source. The left sidebar lists five steps, with Step 3 highlighted. The main area contains the following text and fields:

**Enter database specific properties for the data source**

Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through the datasource.

Name	Value
+ URL	<input type="text" value="jdbc:oracle:thin:@dev151.mw.lab.eng.boc"/>

+ Data store helper class name

Use this data source in container managed persistence (CMP)

Buttons: **Previous** | **Next** | **Cancel**

6. In the **Setup Security Aliases** screen, set the authentication values for connecting to this data source. If the aliases are not yet created, click **Global J2C Authentication Alias** at the bottom. Note that in this case, the **Create a Data Source** wizard will be canceled.
  - a. Click **New...**

- b. Fill in the **Alias**, **User ID**, and **Password**.

Figure 3.16. Creating New Security Alias

**Data sources** > **JAAS - J2C authentication data** > **New...**

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

**General Properties**

\* **Alias**  
jbpmDSalias

\* **User ID**  
dballo13

\* **Password**  
\*\*\*\*\*

**Description**  
[Empty text box]

Apply OK Reset Cancel

- c. Click **OK**.

Go back to the **Setup Security Aliases** screen and set the **Component-Managed Authentication Alias** to the newly created alias and the **Mapping-Configuration Alias** to **DefaultPrincipalMapping**.

You can also create and set a different alias for XA recovery. If the **Authentication Alias for XA Recovery** is set to **(none)**, the component-managed authentication alias is used by default.

Figure 3.17. Setting Security Aliases

**Create a data source**

Step 1: Enter basic data source information  
Step 2: Select JDBC provider  
Step 3: Enter database specific properties for the data source  
→ **Step 4: Setup security aliases**  
Step 5: Summary

**Setup security aliases**

Select the authentication values for this resource.

Authentication alias for XA recovery  
(none)

Component-managed authentication alias  
dhcp-4-116Node01/jbpmDSalias

Mapping-configuration alias  
DefaultPrincipalMapping

Container-managed authentication alias  
(none)

Note: You can create a new J2C authentication alias by accessing one of the following links. Clicking on a link will cancel the wizard and your current wizard selections will be lost.

[Global J2C authentication alias](#)  
[Security domains](#)

Previous Next Cancel

Click **Next**.

7. In the **Summary** screen, check the values and click **Finish**. Choose to save the changes to the master configuration as well.

Figure 3.18. Creating Data Source Summary Screen

Summary of actions:	
Options	Values
Scope	cells:dhcp-4-116Node01Cell:nodes:dhcp-4-116Node01:servers:server1
Data source name	jbpmDS
JNDI name	jdbc/jbpm
Select an existing JDBC provider	Oracle JDBC Driver (XA)
Implementation class name	oracle.jdbc.xa.client.OracleXADataSource
URL	jdbc:oracle:thin:@dev151.mw.lab.eng.bos.redhat.com:1521:qaora12
Data store helper class name	com.ibm.websphere.rsadapter.Oracle11gDataStoreHelper
Use this data source in container managed persistence (CMP)	true
Authentication alias for XA recovery	(none)
Component-managed authentication alias	dhcp-4-116Node01/jbpmDSalias
Mapping-configuration alias	DefaultPrincipalMapping
Container-managed authentication alias	(none)

- Choose the created data source from a list of all data sources to provide the basic meta properties.
- Click **Custom Properties** under the **Additional Properties** section on the right. Properties like **serverName**, **databaseName**, **userName**, and **password** must now be defined and vary for different databases. Some example database properties are shown below.

Table 3.1. Custom Properties for Different Databases

Database	Properties
H2	<b>URL, user, password</b>
MySQL	<b>serverName, databaseName, port, user, password</b>
PostgreSQL	<b>serverName, databaseName, portNumber, user, password</b>
Oracle	<b>jdbcURL</b>

Once all the connection properties have been defined, click **Test Connection** to ensure the validity of the data source. If the connection was successful, the following message appears at the top of the screen:

The test connection operation for data source DATA\_SOURCE\_NAME on server SERVER\_NAME at node NODE\_NAME was successful.

## 3.6. SETTING UP JMS RESOURCES

IBM WebSphere Application Server must be configured to send and receive JMS messages through Red Hat JBoss BPM Suite. However, before you do this, a service bus must be present. Follow the steps below to create a service bus if one does not already exist.

### Setting up Buses Creating Service Bus

- In the main menu on the left, click **Service Integration** → **Buses**.

2. Click **New...**
3. Enter the name and make sure that the **Bus Security** option is *unchecked*.
4. Click **Next** and then **Finish** to create the service bus.

### Adding Bus Member

Before you continue, add a new bus member. A bus member is a server or a cluster that has been added to this service bus.

1. Go to **Service Integration** → **Buses** and click on the service bus that you have created.
2. Under the **Topology** heading on the right, click **Bus Members**.
3. Click **Add**.
4. In the **Add a New Bus Member** wizard, choose the server and the type of message store for the persistence in the first two steps. Depending on the previous selection, you can also specify the properties of the message store.
5. Click **Finish** in the last step to add a new bus member.

### Creating JMS Connection Factories

To send and receive messages from Red Hat JBoss BPM Suite, you have to create the JMS connection factories, which are needed for establishing connections used for sending messages into queues.

For Business Central, create the following connection factories: **KIE.SIGNAL**, **KIE.INPUT** (for queries), **KIE.RESPONSE.ALL** (for responses), and **KIE.EXECUTOR** (for the executor).

If you will be installing the Intelligent Process Server as well, you need to create **KIE.SERVER.REQUEST**, **KIE.SERVER.RESPONSE**, and **KIE.SERVER.EXECUTOR** connection factories.



#### NOTE

The factory names shown above are suggestions only and you can change them to suit your needs and company guidelines.

1. In the main menu on the left, go to **Resources** → **JMS** → **Connection Factories**.
2. Make sure the correct scope is selected and click **New**.
3. Select the **Default Messaging Provider** option and click **OK**.
4. Enter the name and JNDI name of the factory. For example:
  - Name: **KIE.SIGNAL**
  - JNDI name: **jms/conn/KIE.SIGNAL**



#### NOTE

The JNDI names for **KIE.INPUT**, **KIE.RESPONSE.ALL**, and **KIE.EXECUTOR** are **jms/conn/KIE.INPUT**, **jms/conn/KIE.RESPONSE.ALL**, and **jms/conn/KIE.EXECUTOR** respectively.



5. From the **Bus Name** drop-down list, select the service bus created earlier.  
The rest of the options are not mandatory and can be left with default values.
6. Click **Apply** and choose to save the changes to the master configuration.

### Creating JMS Queues

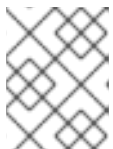
The next step is to create the JMS queues. These queues are the destination end points for point-to-point messaging.

For Business Central, create the following queues: **KIE.RESPONSE.ALL** (for responses), **KIE.AUDIT** (for asynchronous audit logs), **KIE.SESSION** (for **ksession**-based operations), **KIE.TASK** (for task-based operations), **KIE.EXECUTOR** (for Business Central executor services), and **KIE.SIGNAL** (for sending external global signals to processes).

For Intelligent Process Server, create the following: **KIE.SERVER.REQUEST** (for requests), **KIE.SERVER.RESPONSE** (for responses), and **KIE.SERVER.EXECUTOR** (for executor services).

To create these queues:

1. In the main menu, go to **Resources** → **JMS** → **Queues**.
2. Make sure the correct scope is selected and click **New**.
3. Select the **Default Messaging Provider** radio button and click **OK**.
4. Enter the name and the JNDI name of the queue, for example:
  - **Name: KIE.RESPONSE.ALL**
  - **JNDI name: jms/KIE.RESPONSE.ALL**



#### NOTE

All of the JNDI names of other queues follow the same convention as the example above.

5. From the **Bus Name** drop-down list, select the service bus created earlier.
6. From the **Queue Name** drop-down list, make sure to select the **Create Service Integration Bus Destination**.  
This will open up the **Create New Queue** form for creating a new service integration bus. In this form, enter a unique identifier and select the bus member created earlier in this section.
7. Click **Apply** at the bottom and choose to save the changes to the master configuration.

### Creating JMS Activation Specifications

A JMS activation specification is required to be the bridge between the queue and the message-driven bean.

For Business Central, create the following activation specifications: **KIE.RESPONSE.ALL** (for responses), **KIE.AUDIT** (for asynchronous audit logs), **KIE.SESSION** (for **ksession**-based operations), **KIE.TASK** (for task-based operations), **KIE.EXECUTOR** (for Business Central executor services), and **KIE.SIGNAL** (for sending external global signals to processes).

For Intelligent Process Server, create the following: **KIE.SERVER.REQUEST** (for requests), **KIE.SERVER.RESPONSE** (for responses), and **KIE.SERVER.EXECUTOR** (for executor services).

1. In the main menu, go to **Resources** → **JMS** → **Activation Specifications**.
2. Make sure the correct scope is selected and click **New**.
3. Check the **Default Messaging Provider** radio button and click **OK**.
4. Enter the name and the JNDI name of the activation specification, for example:
  - **Name: KIE.RESPONSE.ALL**
  - **JNDI name: jms/activation/KIE.RESPONSE.ALL**



#### NOTE

All of the JNDI names of other activation specifications follow the same convention as the example above.

5. From the **Destination Type** drop-down list, make sure to select **Queue**.
6. Enter the **Destination JNDI Name** (as created in the previous procedure), for example **jms/KIE.RESPONSE.ALL**.
7. From the **Bus Name** drop-down list, choose the service bus created earlier.
8. Click **OK** at the bottom with the rest of the field values as default and choose to save the changes to the master configuration.

You have now successfully completed the JMS configurations required for setting up Red Hat JBoss BPM Suite on IBM WebSphere Application Server.

### 3.7. ADDING CUSTOM JVM PROPERTIES

You must add custom properties to the JVM that is used to start IBM WebSphere Application Server. These custom properties take into consideration the configuration changes that have been outlined in previous sections of this guide.

1. In the main menu, go to **Servers** → **Server Types** → **WebSphere Application Servers**.
2. In the list of application servers, choose the server on which you are going to deploy Business Central.
3. Under the **Server Infrastructure** heading on the right, click **Java and Process Management** → **Process Definition**.
4. Click **Java Virtual Machine** under the **Additional Properties** heading.  
This opens up the configuration properties for the JVM that is used to start WebSphere Application Server.
5. Click **Custom Properties** under **Additional Properties**.
6. Create the following properties by clicking **New...**  
**Custom JVM Properties**

**Table 3.2. Properties Required for Business Central and Intelligent Process Server**

Name	Value	Description
<b>org.kie.executor.jms.queue</b>	<b>jms/KIE.EXECUTOR</b> or <b>jms/KIE.SERVER.EXECUTOR</b>	The JNDI name of the JMS queue for scheduling asynchronous job execution by Business Central ( <b>jms/KIE.EXECUTOR</b> ) or Intelligent Process Server ( <b>jms/KIE.SERVER.EXECUTOR</b> ). Choose the right value based on the application being used for execution.
<b>org.kie.executor.jms.cf</b>	<b>jms/conn/KIE.EXECUTOR</b>	The JNDI name of connection factory to be used for sending messages.
<b>org.jboss.logging.provider</b>	<b>jdk</b>	This property is only required where a <b>CA SiteMinder TAI (SMTAI)</b> is installed in the environment. Using this property forces Hibernate to use <b>JDK</b> instead of <b>log4j</b> for logging within Dashbuilder. <b>CA SiteMinder TAI (SMTAI)</b> contains an old version of <b>log4j</b> , which causes conflicts.
<b>org.apache.wink.jaxbcontextcache</b>	<b>off</b>	This property ensures that the IBM WebSphere Apache Wink framework does not cache <b>JAXBContexts</b> , which negatively impacts the performance and interferes with the custom-type serialization for the REST API.

Table 3.3. Properties Required for Business Central

Name	Value	Description
<b>jbpm.ut.jndi.lookup</b>	<b>jta/usertransaction</b>	Used to look up user transactions from within non-managed threads, such as timers.
<b>kie.services.jms.queues.response</b>	<b>jms/KIE.RESPONSE.ALL</b>	The JNDI name of the response queue for the JMS remote API created earlier. You may have given this a different name from <b>KIE.RESPONSE.ALL</b> . Change if required.
<b>org.uberfire.start.method</b>	<b>ejb</b>	<p>Defines startable beans for Uberfire.</p> <p>Set this property if following warning message appears in the logs during the deployment of <b>business-central.war</b>:</p> <pre>WARNING: Unable to instantiate EJB Asynchronous Bean. Falling back to Executors' CachedThreadPool</pre>

**NOTE**

Red Hat JBoss BPM Suite uses an embedded version of Git for its artifact versioning. This version of Git uses ports **9418** and **8001** for standard and SSH access (**org.uberfire.nio.git.ssh.port**) respectively.

Ensure that these embedded Git ports are not already in use in your version of IBM WebSphere Application Server.

If these ports are being used and you need to change the default Git ports, they can be changed by setting the **org.uberfire.nio.git.daemon.port** and **org.uberfire.nio.git.ssh.port** properties using the steps described above.

For more information, see section *Configuring LDAP Principal and Role Names Matching Criteria* below.

**Table 3.4. Properties Required for Intelligent Process Server**

Name	Value	Description
<b>kie.server.jms.queues.response</b>	<b>jms/conn/KIE.SERVER.RESPONSE</b>	The JNDI name of connection factory for responses used by the Intelligent Process Server.
<b>org.kie.server.domain</b>	<b>WSLogin</b>	JAAS <b>LoginContext</b> domain used to authenticate users when using JMS.
<b>org.kie.server.persistence.ds</b>	Set according to database type and configuration.	Datasource JNDI name.
<b>org.kie.server.persistence.tm</b>	<b>org.hibernate.service.jta.platform.internal.WebSphereExtendedJtaPlatform</b>	Transaction manager platform for setting Hibernate properties.
<b>org.kie.server.persistence.dialect</b>	Set according to database type and configuration.	Specifies the Hibernate dialect to be used.

7. Save these configuration settings to the master configuration.
8. Restart IBM WebSphere Application Server for these changes to take effect.

### 3.8. CONFIGURING LDAP PRINCIPAL AND ROLE NAMES MATCHING CRITERIA

The client applications using ssh to interact with the Git server bundled with Business Central are authenticated and authorised to perform git operations using the security API offered by the Uberfire server. If your Red Hat JBoss BPM Suite application is deployed on WebSphere Application Server (WAS) using an LDAP security realm, the git clients may not be authorized as expected. This is because the distinguished name (DN) for the principal (user or group name) assigned by WAS is the more

complex DN associated with that principal by LDAP, which leads to a mismatch of names when the Uberfire server tries to map the roles. To ensure that the role mapping does not fail, use the system property **org.uberfire ldap.regex.role\_mapper** to control the matching criteria of LDAP principal to role names.

The system property **org.uberfire ldap.regex.role\_mapper** is a regex pattern used to map LDAP principal names to application role names. Ensure that this pattern contains the variable **role** as it is substituted by the application role name when matching a principal value to the role name. Only after the pattern is matched, the role is added to the user.

For example, if the distinguished name (DN) for the admin group in LDAP is **cn=admin,ou=groups,dc=example,dc=com** and the intended role is **admin**, then setting the following value for property **org.uberfire ldap.regex.role\_mapper** finds a match on **admin** role:

```
cn[\\ ]*=[\\ ]*role
```

### 3.9. 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*.

## CHAPTER 4. INSTALL

Now that the basic configuration is done and IBM WebSphere Application Server is set to deploy Red Hat JBoss BPM Suite, you can upload the WAR deployables that were extracted earlier.

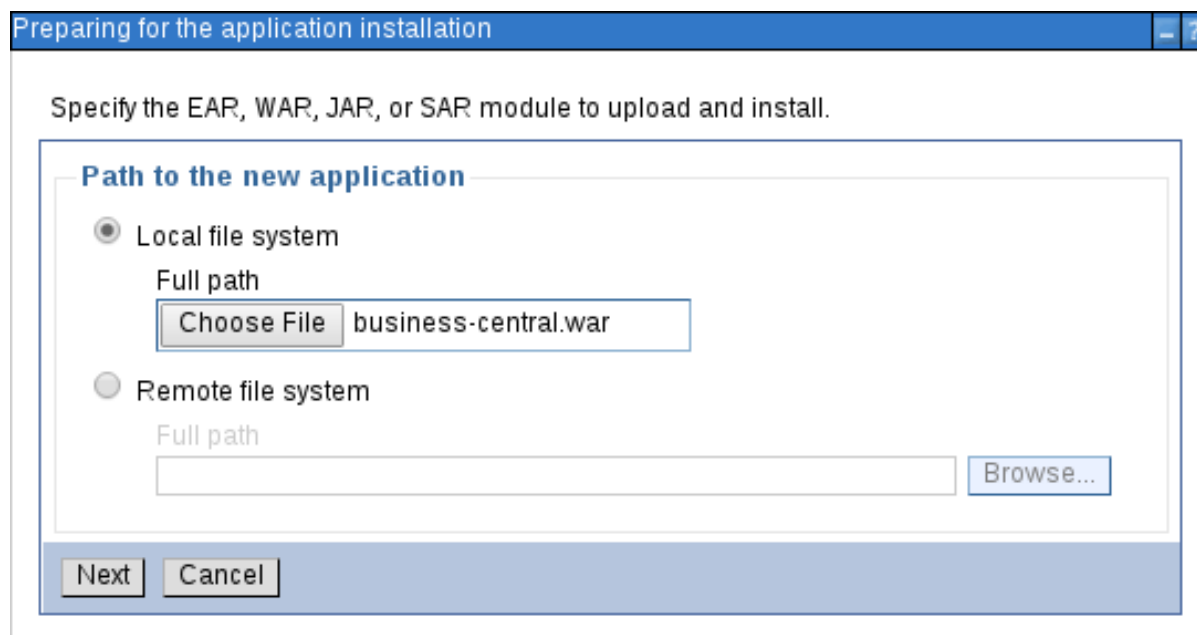
As noted previously, the Red Hat JBoss BPM Suite ZIP file for IBM WebSphere Application Server contains the deployable WAR files for Business Central, Intelligent Process Server, and Dashbuilder.

### 4.1. INSTALLING BUSINESS CENTRAL

Business Central is uploaded as a web archive and can then be accessed at **http://TARGET\_SERVER:PORT/business-central**. Start the deployment by installing the Business Central WAR as a WebSphere application.

1. In the main menu, go to **Applications** → **Application Types** → **WebSphere Enterprise Applications**.  
This will show you all the existing applications in the system and allow you to install a new one.
2. Click **Install** to start the installation process.
3. Upload the Business Central WAR file (**business-central.war**) from the local file system. See [Section 2.2, "Extracting Red Hat JBoss BPM Suite for IBM WebSphere Application Server"](#) for more information.

Figure 4.1. Preparing for Application Installation Wizard



4. Click **Next**. This process may take some time.



## IMPORTANT

You may encounter an error message similar to the following:

The EAR file could be corrupt and/or incomplete. Make sure that the application is at a compatible Java(TM) Platform, Enterprise Edition (Java EE) level for the current version of WebSphere(R) Application Server.  
`java.lang.NullPointerException`

In that case, run **ulimit -n** in the command line. If the result is **1024**, increase the number of open file descriptors. The recommended value is 100 000.

5. Select the **Fast Path** radio button and click **Next**.
6. Change the **Application Name** to **business-central** in the **Select Installation Options** step and click **Next**.

Figure 4.2. Selecting Installation Options

Install New Application

Specify options for installing enterprise applications and modules.

**Step 1: Select installation options**

Step 2 Map modules to servers

Step 3 Bind listeners for message-driven beans

Step 4 Map resource references to resources

Step 5 Map virtual hosts for Web modules

Step 6 Map context roots for Web modules

Step 7 Metadata for modules

Step 8 Summary

**Select installation options**

Specify the various options that are available for your application.

Precompile JavaServer Pages files

Directory to install application

Distribute application

Use Binary Configuration

Deploy enterprise beans

Application name  
**business-central**

Create MBeans for resources

Override class reloading settings for Web and EJB modules

Reload interval in seconds

Deploy Web services

Validate input off/warn/fail  
 warn

Process embedded configuration

**File Permission**

Allow all files to be read but not written to  
 Allow executables to execute  
 Allow HTML and image files to be read by everyone

.\*\.\*=755#.\*\.\*=755#.\*\.\*=755#.\*\.\*=755

Application Build ID  
 6.4.0.Final-redhat-3

Allow dispatching includes to remote resources

Allow servicing includes from remote resources

Business level application name  
 Create New BLA

Asynchronous Request Dispatch Type  
 Disabled

Allow EJB reference targets to resolve automatically

Deploy client modules

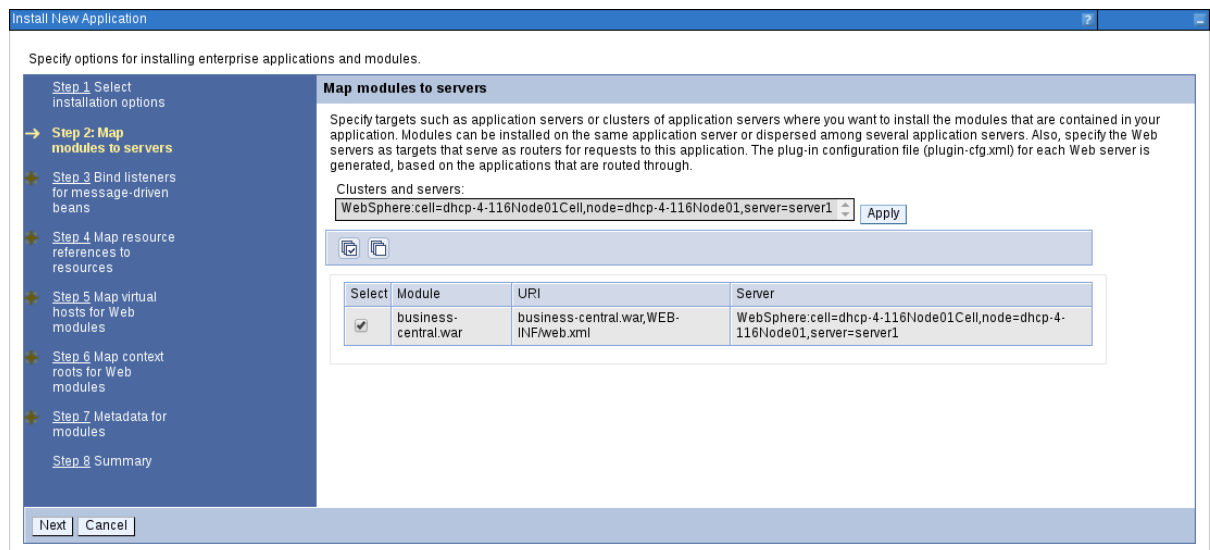
Client deployment mode  
 Isolated

Validate schema

Next Cancel

7. In the **Map Modules to Servers** step, map the Business Central modules to servers according to your requirements.

Figure 4.3. Mapping Modules to Servers

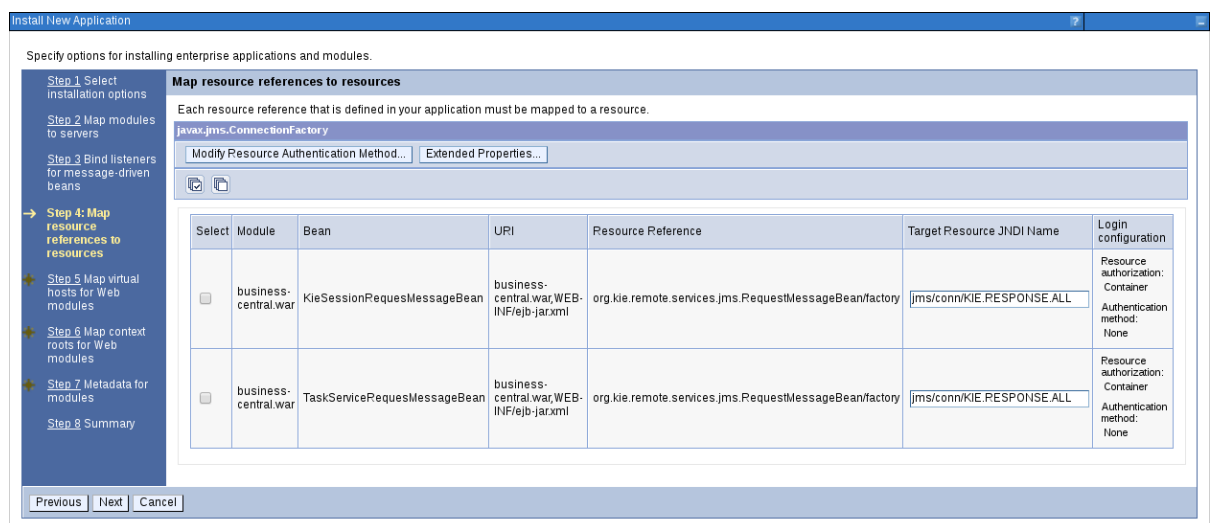


8. Next, for each bean, select the **Activation Specification** radio button and enter the JMS Activation JNDI names created earlier into the **Target Resource JNDI Name** field:

- For **KieSessionRequesMessageBean**, enter **jms/activation/KIE.SESSION**.
- For **TaskServiceRequesMessageBean**, enter **jms/activation/KIE.TASK**.
- For **JMSAuditProcessor**, enter **jms/activation/KIE.AUDIT**.
- For **JmsAvailableJobsExecutor**, enter **jms/activation/KIE.EXECUTOR**.
- For **JMSSignalReceiver**, enter **jms/activation/KIE.SIGNAL**.

9. In the next step, where you have to map resource references to actual resources, you will be asked to fill the **Target Resource JNDI Name** field for two beans: **KieSessionRequesMessageBean** and **TaskServiceRequesMessageBean**. Enter the JNDI name for the **KIE.RESPONSE.ALL** connection factory that you created earlier: **jms/conn/KIE.RESPONSE.ALL**.

Figure 4.4. Mapping Resource References to Resources



10. In the **Map Virtual Hosts for Web Modules** step, leave the default values and click **Next**.

11. In the next step, set the context root to **business-central**.



12. In the **Metadata for Modules** step, leave the default values and click **Next**.
13. In the **Summary** page, click **Finish** to install Business Central. This process can take a while. Save the changes to the master configuration at the end of this process.

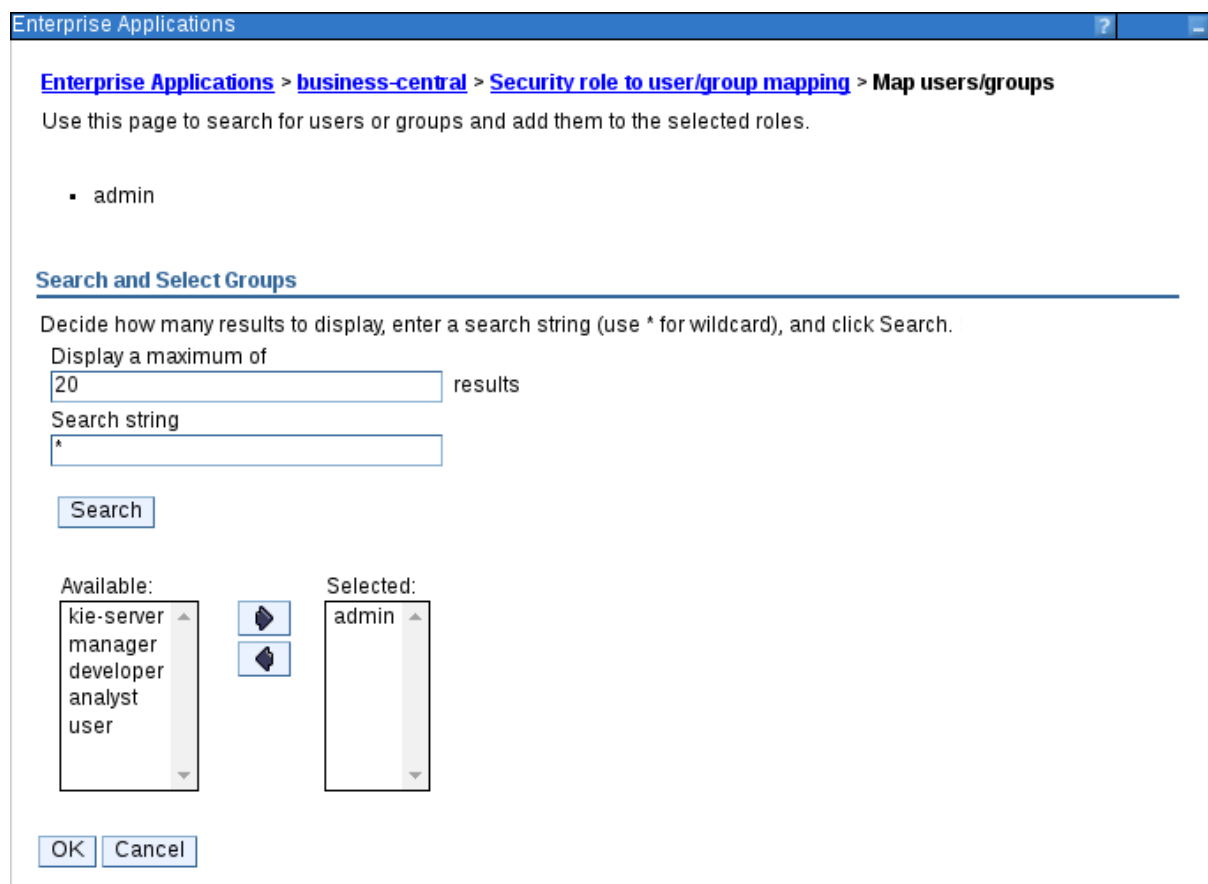
You will be returned to the **WebSphere Enterprise Applications** page where **business-central** will be listed as a new application. However, it will be stopped at this stage. Before you start the application, you need to map groups to roles, configure class loading, and enable the Bouncy Castle Crypto API.

### Mapping Groups to Roles

1. Click on the **business-central** application to open the application configuration page.
2. Click **Security Role to User/Group Mapping** under the **Detail Properties** heading on the left.
3. Select the **admin** role and click **Map Groups...** at the top.
4. Search for the **admin** group (or just click the **Search** button) and move it from the **Available** list to the **Selected** list. Click **OK**.

This mapping gives the previously created administrator user access to the Business Central application.

Figure 4.5. Mapping Groups



5. Follow the same procedure for the **analyst** role as well and save the configuration.

If you have other groups or users that should have access to Business Central, use the same steps to map them to the **admin** or **analyst** roles, or any other roles that you have, such as **developer**, **manager**, or **user**.



## NOTE

If you are also installing the Intelligent Process Server, give this user access to the **kie-server** role. Additionally, map appropriate REST API roles if you are going to use the REST API. For further information about API roles, see chapter [Remote API](#) of *Red Hat JBoss BPM Suite Development Guide*.

### Class-Loading Configuration

Ensure correct class-loading configuration by following the steps below.

1. In the main menu, go to **Applications** → **Application Types** → **WebSphere Enterprise Applications**.
2. Click **business-central**.
3. Click **Class Loading and Update Detection** under the **Detail Properties** heading on the left.
4. Check the following options:
  - *Class Loader Order*: **Classes loaded with local class loader first (parent last)**
  - *WAR Class Loader Policy*: **Single class loader for application**

Figure 4.6. Configuring Class Loading

The screenshot shows the 'Enterprise Applications' configuration window for 'business-central' under the 'Class loader' tab. The page title is 'Enterprise Applications > business-central > Class loader'. Below the title, there is a description: 'Use this page to configure the reloading of classes when application files are updated.' The main configuration area is titled 'Configuration' and contains three sections: 'General Properties', 'Class reloading options', 'Class loader order', and 'WAR class loader policy'. At the bottom, there are four buttons: 'Apply', 'OK', 'Reset', and 'Cancel'.

**Enterprise Applications** > **business-central** > **Class loader**

Use this page to configure the reloading of classes when application files are updated.

Configuration

**General Properties**

**Class reloading options**

Override class reloading settings for Web and EJB modules

Polling interval for updated files  Seconds

**Class loader order**

Classes loaded with parent class loader first

Classes loaded with local class loader first (parent last)

**WAR class loader policy**

Class loader for each WAR file in application

Single class loader for application

5. Click **OK**, save the changes to the master configuration and restart the IBM WebSphere Application Server.

### Enabling Bouncy Castle Crypto API

To enable Git SSH repository cloning and **kie-config-cli** from within Business Central, the Bouncy Castle Crypto API must be set up. Enable the API by following the steps below.

1. Set the **org.apache.sshd.registerBouncyCastle** property to **true** and **org.uberfire.domain** property to **WSLogin**. See [Section 3.7, “Adding Custom JVM Properties”](#) for detailed instructions on how to set custom properties.
2. Set up the Bouncy Castle API as a shared library referenced from Business Central using the appropriate version of Bouncy Castle:
  - i. In the main menu on the left, navigate to **Environment** → **Shared Libraries**, select the appropriate scope and click **New...** to create a new library.
  - ii. Give this library a name and set the class path to the Bouncy Castle library (**bcprov-jdk16-1.46.jar**). Click **OK** and save the configuration.

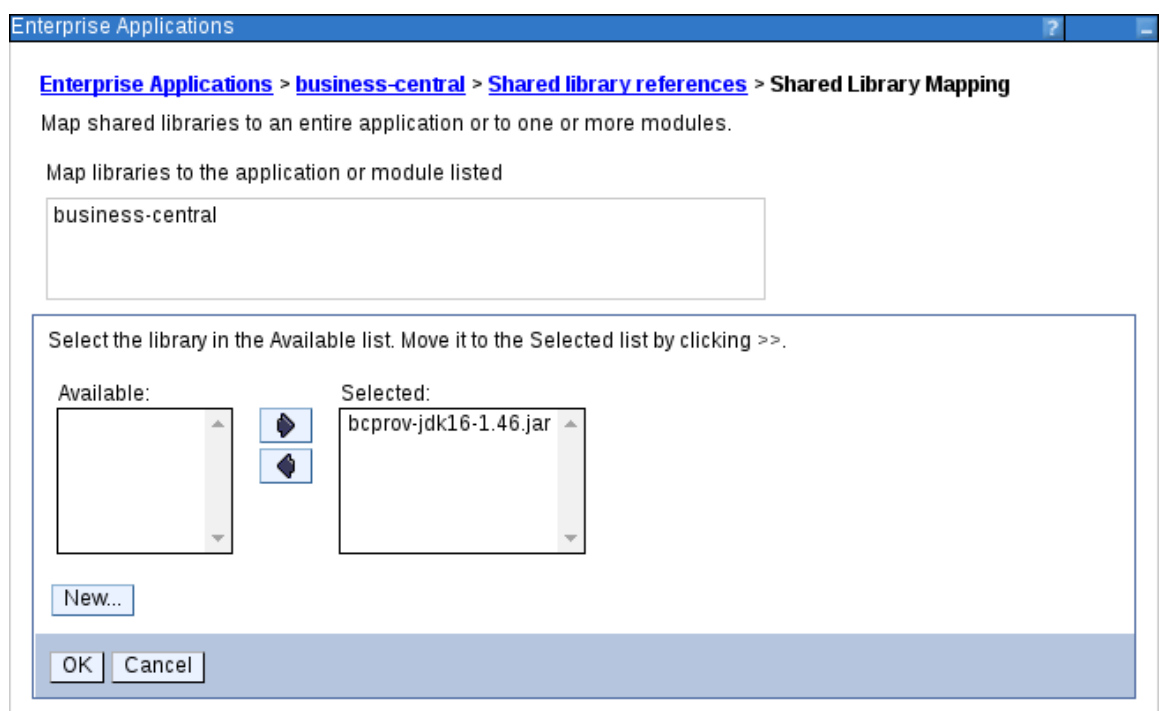


#### NOTE

For more information about the **bcprov-jdk16-1.46.jar** package, see the [Maven Repository – Bouncy Castle Provider](#) page.

- iii. Go to **Applications** → **Application Types** → **WebSphere Enterprise Applications** and click **business-central**.
- iv. Click **Shared Library References** under the **References** heading on the left, select the web module, click on **Reference Shared Libraries** and move the Bouncy Castle library created in the previous step from the **Available** to the **Selected** list.

Figure 4.7. Mapping Shared Libraries



- v. Click **OK** and save the configuration.

You have now successfully installed Business Central on IBM WebSphere Application Server.

To start the application, go back to **Applications → Application Types → WebSphere Enterprise Applications** page and select the **business-central** checkbox before clicking **Start**.

To access the application, navigate to **http://TARGET\_SERVER:PORT/business-central** in your web browser.

## 4.2. INSTALLING DASHBUILDER

Dashbuilder for IBM WebSphere Application Server is distributed as a deployable WAR file (**dashbuilder.war**) as a part of the Red Hat JBoss BPM Suite download.



### NOTE

Before installing Dashbuilder, set up a data source for Dashbuilder by following the steps outlined in [Section 3.5, "Setting up Data Source"](#).

Dashbuilder needs an access to the same database schema as Business Central. In the first step of creating a data source, specify the data source JNDI name as **jdbc/jbpm**.

1. In the main menu of the IBM Integrated Solutions Console, go to **Applications → Application Types → WebSphere Enterprise Applications** and click **Install**.
2. Upload the Dashbuilder WAR file (**dashbuilder.war**) from the local file system and click **Next**. This process may take some time.

Figure 4.8. Preparing for Application Installation Wizard

3. Select **Fast Path** radio button and click **Next**. The **Install New Application** window opens.
4. In the **Select Installation Options** step, change the **Application Name** to **dashbuilder** and click **Next**.

Figure 4.9. Selecting Installation Options

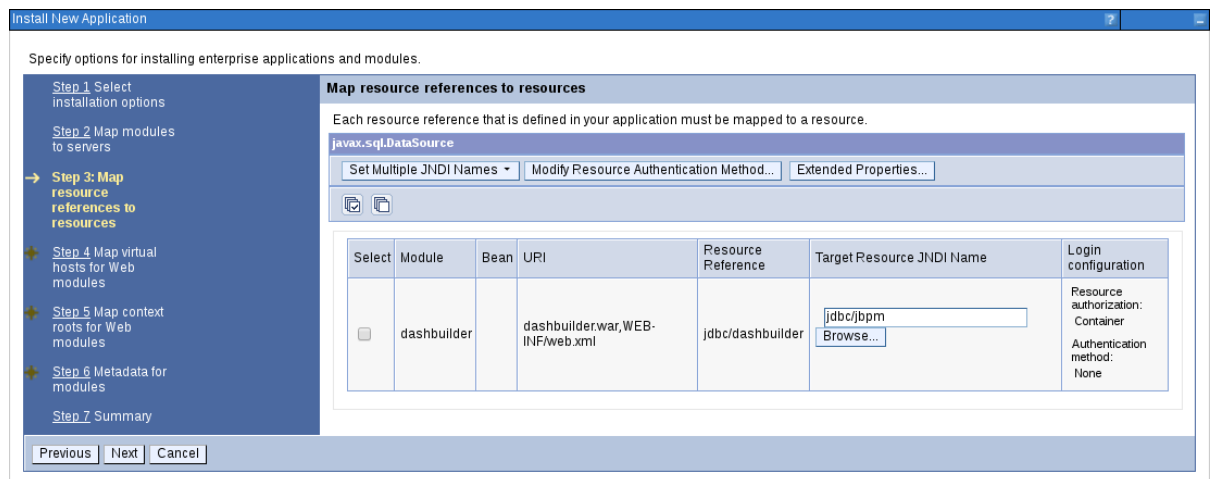
- In the **Map Modules to Servers** step, map the Dashbuilder modules to servers according to your requirements and click **Next**.

Figure 4.10. Mapping Modules to Servers

Select	Module	URI	Server
<input checked="" type="checkbox"/>	dashbuilder	dashbuilder.war,WEB-INF/web.xml	WebSphere:cell=dhcp-4-116Node01Cell,node=dhcp-4-116Node01,server=server1

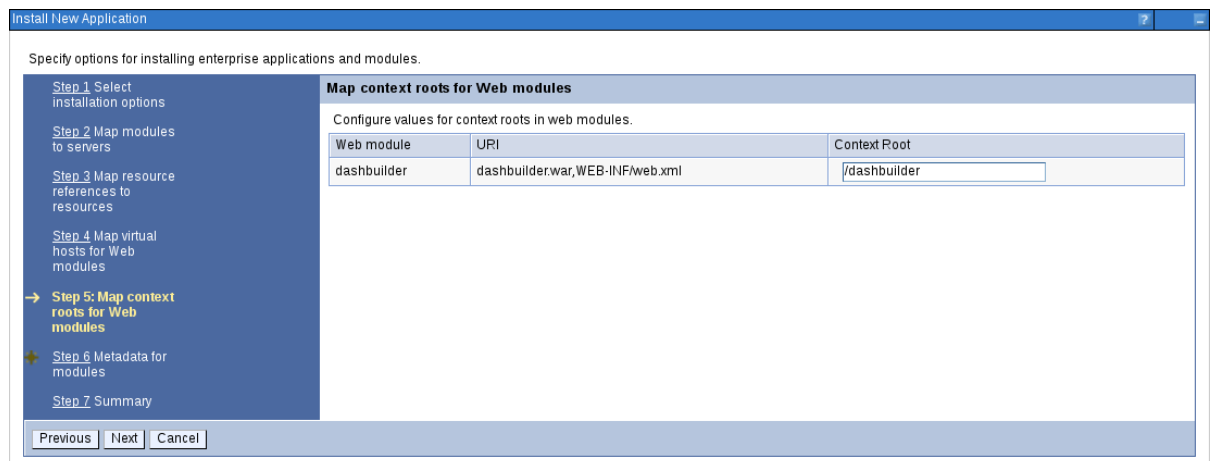
- In the **Map Resource References to Resources** step, set the **Target Resource JNDI Name** of the **dashbuilder** module to **jdbc/jbpm**. Click **Next**.

Figure 4.11. Mapping Resource References to Resources



- In the **Map Virtual Hosts for Web Modules** step, leave the default values and click **Next**.
- In the next step, set the context root to **/dashbuilder** and click **Next**.

Figure 4.12. Mapping Context Roots for Web Modules



- In the **Metadata for Modules** step, leave the default values and click **Next**.
- In the **Summary** page, click **Finish** to install Dashbuilder. This process may take a while. Save the changes to the master configuration at the end of this process.

You will be returned to the **Enterprise Applications** page where **dashbuilder** will be listed as a new application. However, it will be stopped at this stage. Before you start the application, you need to map groups to roles and configure correct class loading.

### Mapping Groups to Roles

- Click on the **dashbuilder** application to open the application configuration page.
- Click **Security Role to User/Group Mapping** under the **Detail Properties** heading on the left.
- Select the **admin** role and click **Map Groups...** at the top.
- Search for the **admin** group (or just click the **Search** button) and move it from the **Available** list to the **Selected** list. Click **OK**.  
This mapping gives the previously created administrator user access to the Dashbuilder application.

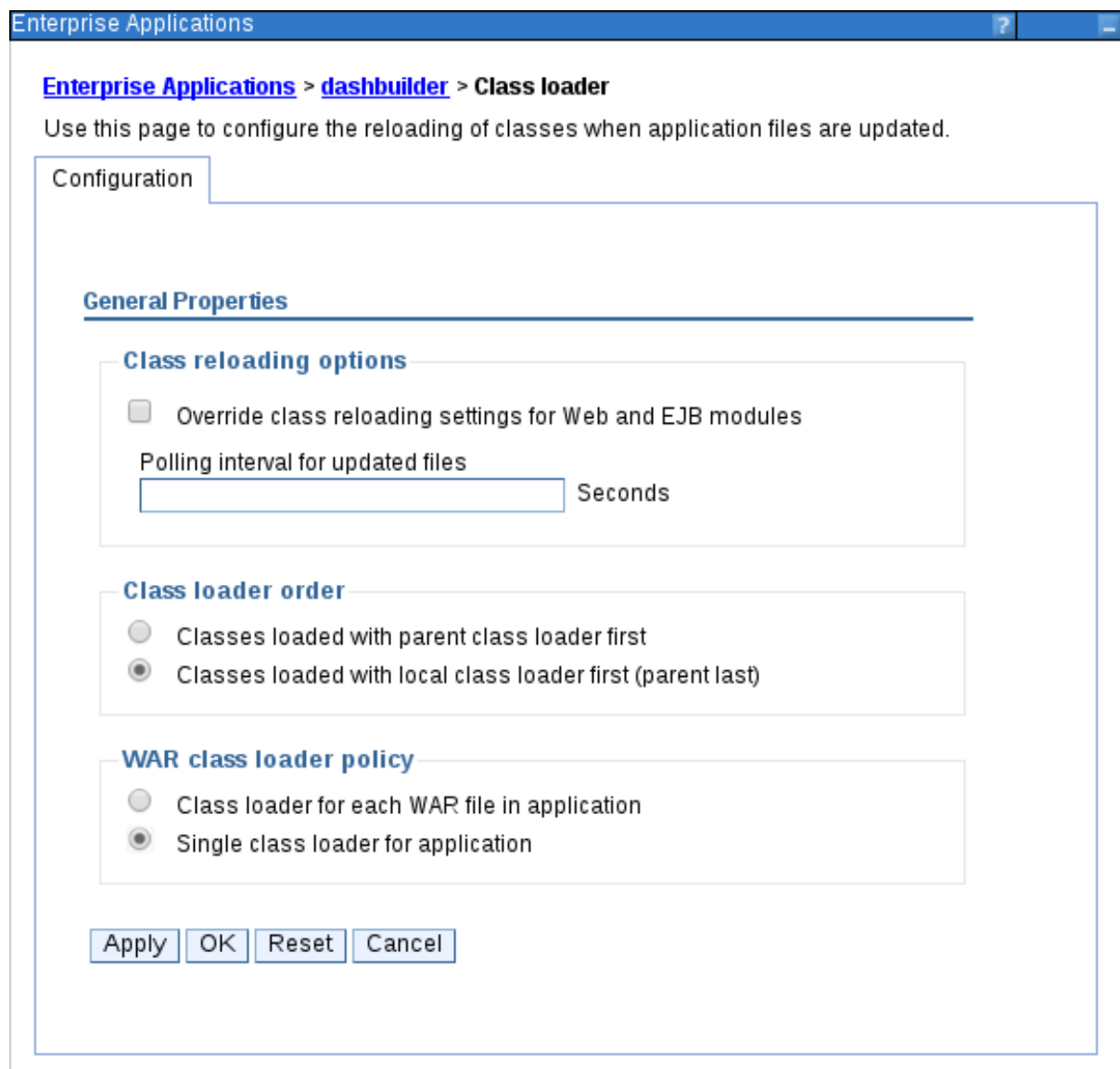
5. Follow the same procedure for the **analyst**, **developer**, **user**, and **manager** roles.

### Class-Loading Configuration

Ensure correct class-loading configuration by following the steps below.

1. In the main menu, go to **Applications** → **Application Types** → **WebSphere Enterprise Applications**.
2. Click **dashbuilder**.
3. Click **Class Loading and Update Detection** under the **Detail Properties** heading on the left.
4. Check the following options:
  - *Class Loader Order*: **Classes loaded with local class loader first (parent last)**
  - *WAR Class Loader Policy*: **Single class loader for application**

Figure 4.13. Configuring Class Loading



5. Click **OK**, save the changes to the master configuration and restart the IBM WebSphere Application Server.

You have now successfully installed Dashbuilder on IBM WebSphere Application Server.

To start the application, go back to **Applications → Application Types → WebSphere Enterprise Applications** page and select the **dashbuilder** checkbox before clicking **Start**.

You can now log in to Dashbuilder at **http://TARGET\_SERVER:PORT/dashbuilder** using the user that you have created in the previous steps.

### 4.3. INSTALLING INTELLIGENT PROCESS SERVER

The Intelligent Process Server is distributed as a web application archive file (**kie-server.war**) and is present in your Red Hat JBoss BPM Suite 6.4.0 Deployable for WebSphere 8.5 download.



#### NOTE

It is assumed that you followed the steps described in [Section 3.3, "Creating Users and Groups"](#) to create the role **kie-server** required by the Intelligent Process Server. Similarly, the JMS connection factories, queues, and activation specifications required by the Intelligent Process Server need to be created. If you have not done so, revisit the respective sections in this guide.

1. In the main menu on the left, go to **Applications → Application Types → WebSphere Enterprise Applications**.  
This will show you all the existing applications in the system and allow you to install a new one.
2. Click **Install** to start the installation process.
3. Upload the Intelligent Process Server WAR file (**kie-server.war**) from the local file system.
4. Select the **Fast Path** radio button and click **Next**.  
The **Install New Application** wizard opens.
5. Change the **Application Name** to **kie-server** in the first step and click **Next**.
6. In the next step, map the Intelligent Process Server modules to servers according to your requirements and click **Next**.
7. In the **Bind Listeners for Message-Driven Beans** step, select the **Activation Specification** radio button for both the beans. Enter **jms/activation/KIE.SERVER.EXECUTOR** as **Target Resource JNDI Name** for the **KieExecutorMDB** bean, and **jms/activation/KIE.SERVER.REQUEST** for the **KieServerMDB** bean.
8. In the next step, map resource references to actual resources. Enter the JNDI name for the **KIE.SERVER.REQUEST** connection factory that you created earlier: **jms/conn/KIE.SERVER.REQUEST**.
9. In the **Map Virtual Hosts for Web Modules** step, leave the default values and click **Next**.
10. In the next screen, set the context root to **kie-server**.
11. In the **Metadata for Modules** step, leave the default values and click **Next**.
12. Click **Finish** to install the Intelligent Process Server. Save the changes to the master configuration at the end of this process.

#### Class-Loading Configuration

Ensure correct class-loading configuration by following the steps below.



1. Navigate to **Applications → Application Types → WebSphere Enterprise Applications** and click **kie-server**.
2. Click **Class Loading and Update Detection** under the **Detail Properties** heading on the left.
3. In the properties, change *Class Loader Order* to **Classes loaded with local class loader first (parent last)** and *WAR Class Loader Policy* to **Single class loader for application**
4. Save the changes to the master configuration.

### Mapping Groups to Roles

If you have already mapped the **kie-server** role to a user or a group, you can ignore this procedure. Otherwise, do the following:

1. Go back to the main configuration page for the newly installed **kie-server** application (**Applications → Application Types → WebSphere Enterprise Applications**). Click **Security Role to User/Group Mapping** under the **Detail Properties** heading on the left.
2. Select the **kie-server** role, click **Map Groups...** and search for the **kie-server** group in the next screen (or just click the **Search** button).
3. Move it from the **Available** list to the **Selected** list. Click **OK**.

This mapping gives the previously created administrator user access to the Intelligent Process Server.

You can now save the changes and start the **kie-server** application.

Check whether the Intelligent Process Server REST API works by sending a GET request at **http://TARGET\_SERVER:PORT/kie-server/services/rest/server**.

## APPENDIX A. VERSIONING INFORMATION

Documentation last updated on: Monday, May 13, 2019.