Red Hat JBoss BPM Suite 6.3

Getting Started Guide

The Getting Started Guide for Red Hat JBoss BPM Suite.

Last Updated: 2017-11-16

Red Hat JBoss BPM Suite 6.3 Getting Started Guide

The Getting Started Guide for Red Hat JBoss BPM Suite.

Red Content Services

Gemma Sheldon
gsheldon@redhat.com

Klara Kufova
kkufova@redhat.com

Marek Czernek
mczernek@redhat.com

Tomas Radej
tradej@redhat.com

Vidya Iyengar
viyengar@redhat.com

Legal Notice

Abstract

This guide provides information for getting started with Red Hat JBoss BPM Suite.

Table of Contents

CHAPTER 1. QUICK START WITH RED HAT JBOSS BPM SUITE 3

CHAPTER 2. INSTALLATION OPTIONS 4

2.1. EAP 6 BUNDLE INSTALLATION 4

2.2. GENERIC DEPLOYABLE BUNDLE INSTALLATION 8

2.3. STARTING SERVER 16

2.4. JAVA SECURITY MANAGER AND PERFORMANCE MANAGEMENT 18

CHAPTER 3. LOGGING ON TO BUSINESS CENTRAL 19

CHAPTER 4. HELLO WORLD PROJECT 20

4.1. CREATING REPOSITORY STRUCTURE 20

CHAPTER 5. HELLO WORLD PROCESS 22

5.1. CREATING BUSINESS PROCESS 22

5.2. MODELING BUSINESS PROCESS 22

5.3. DEFINING ELEMENT PROPERTIES 23

5.4. BUILDING AND DEPLOYING 24

5.5. INSTANTIATING BUSINESS PROCESS 25

5.6. ABORTING BUSINESS PROCESS 25

CHAPTER 6. HELLO WORLD BUSINESS RULE 27

6.1. CREATING BUSINESS RULE 27

6.2. ADDING BUSINESS RULE TASK 28

6.3. BUILDING AND DEPLOYING 28

6.4. INSTANTIATING BUSINESS PROCESS 29

CHAPTER 7. BAM 31

7.1. ACCESSING RED HAT JBOSS BPM SUITE DASHBUILDER 31

7.2. MONITORING INSTANCES 31

CHAPTER 8. RED HAT JBOSS DEVELOPER STUDIO 32

8.1. JBOSS CENTRAL 32

8.2. INSTALLING RED HAT JBOSS DEVELOPER STUDIO PLUG-INS 33

8.3. SETTING DROOLS RUNTIME 34

8.4. SETTING RED HAT JBOSS BPM SUITE RUNTIME 34

8.5. CONFIGURING RED HAT JBOSS BPM SUITE SERVER 35

8.6. IMPORTING PROJECTS FROM GIT REPOSITORY INTO RED HAT JBOSS DEVELOPER STUDIO 35

8.7. CREATING DROOLS PROJECT 38

8.8. CREATING RED HAT JBOSS BPM SUITE PROJECT 39

CHAPTER 9. BUSINESS RESOURCE PLANNER 40

9.1. INSTALLING BUSINESS RESOURCE PLANNER 40

9.2. RUNNING BUSINESS RESOURCE PLANNER EXAMPLES 40

APPENDIX A. REVISION HISTORY 41

CHAPTER 1. QUICK START WITH RED HAT JBOSS BPM SUITE

In this section we download, install and run Red Hat JBoss BPM Suite with the minimum number of instructions. For more detailed steps and alternate methods of installation, skip this section and continue reading the rest of this guide.

These instructions assume that you have the minimum supported Java version installed on your system. An existing Red Hat JBoss EAP server (version 6.4 or better) is also required as the installer does not install one.

NOTE

It may be necessary for you to update your Red Hat JBoss EAP server. If the installer informs you of such need, download the newest patch from the Customer Portal and follow the instructions for updating in the*Red Hat JBoss EAP Installation Guide*.

Procedure: Red Hat JBoss BPM Suite Quick Start

1. Download installer from the Customer Portal.
2. Run installer by executing the following command:

```
java -jar jboss-bpmsuite-6.3.0.GA-installer.jar
```

3. Step through the GUI installer steps making note of the username/password for accessing the Red Hat JBoss BPM Suite application after installation.
4. After successful installation, in a command prompt, navigate to the EAP home folder where you have installed Red Hat JBoss BPM Suite.
5. Start the server by issuing the following command:

```
bin/standalone.sh
```

6. Open a web browser and login to Business Central by navigating to **`http://localhost:8080/business-central/`**. Login using the username/password combination for accessing the Red Hat JBoss BPM Suite application that you created at installation time.

You have successfully installed and run Red Hat JBoss BPM Suite. You can now do the following:

- Learn more about roles in Section 2.1.3, “Defining Roles”.
- Create additional users (see Section 2.1.4, “Creating Users”).
- Create a Hello World project (see Chapter 4, *Hello World Project*).
- Look up detailed installation instructions for using the installer in the *Red Hat JBoss BPM Suite Installation Guide*.

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-6.3.0.GA-deployable-eap6.x.zip`**: version adapted for deployment on Red Hat JBoss Enterprise Application Platform.
  - **`jboss-bpmsuite-6.3.0.GA-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.

2.1. EAP 6 BUNDLE INSTALLATION

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

NOTE

The minimum supported version of Red Hat JBoss EAP for Red Hat JBoss BPM Suite installation is 6.4.

2.1.1. Downloading Red Hat JBoss EAP 6 Package

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

1. Go to the Red Hat Customer Portal and log in.
2. Click **DOWNLOADS** 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 version **6.3**.

5. On the **Software Downloads** page that opens, navigate to the**Red Hat JBoss BPM Suite 6.3.0 Deployable for EAP 6** row and click **Download**.

2.1.2. Installing Red Hat JBoss EAP 6 Package

Installation on Fresh Red Hat JBoss EAP Instance
To install the deployable package for the Red Hat JBoss EAP that has yet to be configured, do the following:

1. Extract the ZIP package deployable for Red Hat JBoss EAP you downloaded from the Red Hat Customer Portal.

2. Merge the extracted ZIP package deployable for JBoss EAP into the JBoss EAP ***SERVER_HOME*** directory.

   > **WARNING**
   >
   > This step must be performed by the same user account that was used to install Red Hat JBoss EAP. This account must *not* be a superuser account.

3. In this process, multiple files in the Red Hat JBoss EAP ***SERVER_HOME*** directory will be overwritten and you must let the unzip process overwrite these files.

4. Make sure that the ***SERVER_HOME*/bin/product.conf** file contains the string **slot=bpms**.

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

```
unzip -u jboss-bpmsuite-VERSION_TYPE.zip -d SERVER_HOME_PARENT_DIRECTORY
```

Example 2.1. The unzip Command

```
unzip -u jboss-bpmsuite-6.3.0.GA-deployable-eap6.x.zip -d
/home/john/myServers/
```

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

Installation on Existing Red Hat JBoss EAP Configuration

> **WARNING**
>
> These instructions are for installing, and *not* for updating an existing Red Hat JBoss BPM Suite instance. Make sure that there is no existing Red Hat JBoss BPM Suite install in the target Red Hat JBoss EAP.

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

1. Extract the ZIP package deployable for Red Hat JBoss EAP you downloaded from the Red Hat Customer Portal.

2. Unzip the downloaded ZIP archive; however, do not overwrite all of the files. Manually merge the following files into the ***SERVER_HOME*** directory.

   - **jboss-eap-6.4/domain/configuration/*** (Please be aware that Red Hat JBoss BMP Suite requires JMS, so JMS is added by default into all profiles in **domain.xml** provided by Red Hat JBoss BPM Suite distribution.)

   - **jboss-eap-6.4/standalone/configuration/*** (Please be aware that Red Hat JBoss BPM Suite requires JMS, so JMS is added by default into all profiles configuration files — especially into **standalone.xml** and **standalone-ha.xml** — provided by Red Hat JBoss BPM Suite distribution.)

   - **jboss-eap-6.4/modules/layers.conf**

   - **jboss-eap-6.4/bin/product.conf**

3. Ensure the target Red Hat JBoss EAP does not include a deployment with a colliding name. Copy the folder **jboss-eap-6.4/standalone/deployments** into the ***EAP_HOME*** directory from the Red Hat JBoss BPM Suite distribution.

4. Make sure no JBoss EAP module layer is already called Red Hat JBoss BPM Suite and copy the folder **jboss-eap-6.4/modules/system/layers/bpms** into the EAP 6.4 folder.

2.1.3. 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. For further information about roles, see the Access Control section from *Red Hat JBoss Administration and Configuration Guide*. For further information about roles and task interactions, see the User Task section of *Red Hat JBoss BPM Suite User Guide*.

2.1.4. 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 2.1.3, “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.

2.2. 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-6.3.0.GA-deployable-generic.zip`**: contains the **`business-central.war`**, **`dashbuilder.war`**, and **`kie-server.war`** web applications.

- **`jboss-bpmsuite-6.3.0.GA-engine.zip`**: supported execution engine libraries for embedding the engine into your application and other libraries needed for generic deployment.

2.2.1. 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 **6.3** (if not already selected).

5. In the Software Downloads section that comes up, navigate to the **Red Hat JBoss BPM Suite 6.3.0 Deployable for All Supported Containers** row and then click **Download**.

6. Also navigate to the **Red Hat JBoss BPM Suite 6.3.0 Core Engine** files row and click **Download** to download the Red Hat JBoss BPM Suite Core Engine files.

2.2.2. 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 2.2.2.1, “Setting up Transaction Manager for Red Hat JBoss Web Server 2.1 (Tomcat 7)”).
- Set up the Business Central application: set up users and roles and set up persistence (see Section 2.2.2.2, “Setting up Business Central for Red Hat JBoss Web Server 2.1 (Tomcat 7)”).
- Set up the Dashbuilder application: set up users and roles and set up persistence (see Section 2.2.2.3, “Setting up Dashbuilder for Red Hat JBoss Web Server 2.0 (Tomcat 7)”).

2.2.2.1. Setting up Transaction Manager for Red Hat JBoss Web Server 2.1 (Tomcat 7)

1. Extract the generic deployable ZIP package you downloaded from 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 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:

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

In addition, download the following library and copy it into the ***$TOMCAT_DIR*/lib/** folder as well: javax.security.jacc-api.jar.

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

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

- **resources.properties** (the **resource.ds1.uniqueName** defines the 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 2.2. H2 Data Source Definition

```
resource.ds1.className=bitronix.tm.resource.jdbc.lrc.LrcXADataS
```

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

Example 2.3. MySQL 5.5 Data Source Definition

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

Example 2.4. 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/sa
mpledb
resource.ds1.driverProperties.user=dbuser
resource.ds1.driverProperties.password=dbpassword
resource.ds1.allowLocalTransactions=true
```

Example 2.5. 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 2.6. Microsoft SQL Server Data Source Definition

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

7. Set up the transaction manager listener in **$*TOMCAT_DIR*/conf/server.xml** to start and stop Bitronix on container startup and shutdown:
   Add the following element as the last **<Listener>** element into the**<Server>** element:

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

8. Define the **btm.root** system property and location where Bitronix 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/TransactionSynchronizationRegist
   ry"
   ```

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

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

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 -
Dbtm.root=C:/Tomcat -
Dbitronix.tm.configuration=C:/Tomcat/conf/btm-
config.properties -
Dorg.jbpm.designer.perspective=RuleFlow"
```

2.2.2.2. Setting up Business Central for Red Hat JBoss Web Server 2.1 (Tomcat 7)

To set up Business Central, do the following:

1. Set up a **Valve** so that the Business Central web application can load the users set up in Tomcat:

   a. Define users and roles in **$*TOMCAT_DIR*/conf/tomcat-users.xml**. Note that Business Central requires users to have the roles specified as **admin** and/or **analyst** (for more information about user and role definitions, 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 **$*TOMCAT_DIR*/webapps/business-central/WEB-INF/lib/** to **$*TOMCAT_DIR*/lib/**.

   c. Copy **jboss-jaxb-api-*VERSION*.jar** from **$*TOMCAT_DIR*/webapps/business-central/WEB-INF/lib/** to **$*TOMCAT_DIR*/lib/**.

   d. In **$*TOMCAT_DIR*/conf/server.xml**, add the Tomcat Valve declaration in the relevant **<host>** element:

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

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** 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 login to Business Central.

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

```
./startup.sh
```

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

c. Login with the user name/password you defined earlier in **`tomcat-users.xml`** file.

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

> **NOTE**
>
> Before setting up Dashbuilder on Red Hat JBoss Web Server, you must ensure that you have correctly installed and started Business Central as described in Section 2.2.2.2, “Setting up Business Central for Red Hat JBoss Web Server 2.1 (Tomcat 7)”. Dashbuilder requires the history log database tables to exist, which are only provided by Business Central. If these tables are not present in the database before attempting the steps below, you may get initialization errors.

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

1. Define users and roles in **`$`*`TOMCAT_DIR`*`/conf/tomcat-users.xml`**. Note that Dashbuilder requires users to have the role specified as **`admin`** and/or **`analyst`**. If you have already defined these users earlier for Business Central, you 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 setup 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.OracleXADataSou
> rce" 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</p
roperty>
```

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

2.3. STARTING SERVER

If you have installed Red Hat JBoss BPM Suite using the Red Hat JBoss EAP 6 bundle install, you can now start your server in one of two modes.

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

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

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

   ```
   ./standalone.sh
   ```

   In a Windows environment run:

   ```
   ./standalone.bat
   ```

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

NOTE

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

WARNING

The use of a security manager imposes a significant performance penalty that you should be aware of. The tradeoff between security and performance must be made by taking into consideration individual circumstances. See Section 2.4, “Java Security Manager and Performance Management”.

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

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

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

   In a Windows environment run:

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

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 install. The installer doesn’t provide a separate **standalone-secure.sh** script.

NOTE

If you are starting the server in the domain mode, the corresponding scripts are **`domain.sh`** and **`domain-secure.sh`** respectively.

2.4. 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.

CHAPTER 3. LOGGING ON TO BUSINESS CENTRAL

Log into Business Central after the server has successfully started.

1. Navigate to `http://localhost:8080/business-central` in a web browser. If the user interface has been configured to run from a domain name, substitute **`localhost`** for the domain name. For example `http://www.example.com:8080/business-central`.

2. Log in with the user credentials that were created during installation. For example, user: **`helloworlduser`** and password: **`Helloworld@123`**.

CHAPTER 4. HELLO WORLD PROJECT

To demonstrate basic features of Red Hat JBoss BPM Suite, this chapter will instruct you on how to set up the Hello World project with a Hello World business process. All that this business process will do, is display the **`Hello World!`** message and finish execution. For additional example that makes use of the full when-then rule functionality, see the Getting Started with Rules and Facts chapter from *Red Hat JBoss BPM Suite Development Guide*

You will be instructed to do the following:

1. Create a Hello World repository in the Artifact repository.
2. Create the Hello World project with the Hello World business process definition.
3. Model the Hello World process logic in the business process using the graphical Process Designer tool.
4. Build and deploy the project on the locally running Execution Engine.
5. Run the Hello World process.
6. Monitor the process execution.

4.1. CREATING REPOSITORY STRUCTURE

All business assets live in repositories that live in organizational units, which are directories in the artifact repository. By default, the artifact repository does not contain any organizational units. Therefore to be able to create your own business assets, such as a business process, you need to create an organizational unit in the artifact repository and a (Git) repository in it. The repository can then contain your project. A project can contain an arbitrary package structure, which allows you to structure its content into logical "directory" structure. Any package can contain your business assets (for more information on the artifact repository, see the *Red Hat JBoss BPM Suite User Guide*).

To create the repository structure, do the following:

1. Open Business Central in your web browser and log in as a user with the **`admin`** role (for example **`helloworlduser`**).
2. Create an organizational unit in the artifact repository:
    a. Go to **Authoring → Administration**.
    b. On the perspective menu, click **Organizational Units → Manage Organizational Units**.
    c. In the displayed **Organizational Unit Manager** view, click **Add**.
    In the displayed **Add New Organizational Unit** dialog box, define the unit properties and click **Ok**:
        - **Name**: **`helloworld`**
        - **Owner**: **`helloworlduser`**
3. Create a new repository in the **`helloworld`** organizational unit:
    a. Go to **Authoring → Administration**.

b. On the perspective menu, click **Repositories** → **New repository**.

c. In the displayed **New Repository** dialog box, define the repository properties:

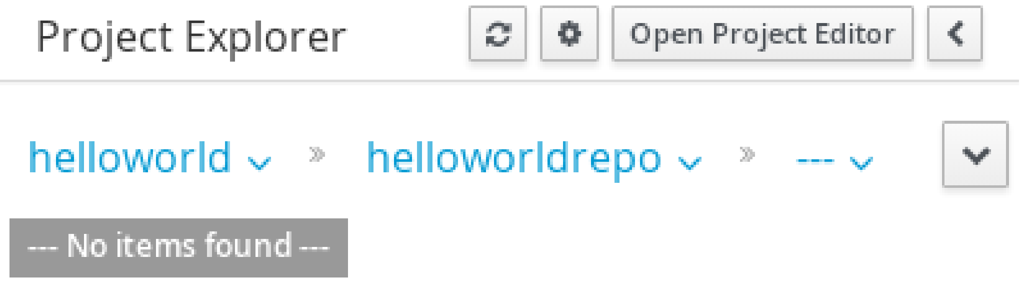
- **Repository Name**: `helloworldrepo`
- **Organizational Unit**: `helloworld`

d. Click **Finish**.

4. Go to **Authoring** → **Project Authoring**.

5. In the Project Explorer in the organizational unit drop-down box, select **`helloworld`**, and in the repository drop-down box select **`helloworldrepo`**.

Figure 4.1. Selecting the helloworldrepo Repository in the helloworld Organizational Group in Project Explorer



6. Create a project in the **`helloworld`** repository:

a. On the perspective menu, go to **New Item** → **Project**.

b. In the displayed **New Project** wizard, define your project properties:

- **Project Name**: `HelloWorld`
- **Group ID**: `org.bpms`
- **Artifact ID**: `HelloWorld`
- **Version ID**: `1.0`

c. Click **Finish**.

CHAPTER 5. HELLO WORLD PROCESS

In this chapter, we will continue the Hello World example and create an end to end business process to demonstrate the creation of a basic Hello World process.

5.1. CREATING BUSINESS PROCESS

To create a new business process definition, do the following:

1. Display the Project Authoring perspective (**Authoring** → **Project Authoring**).

2. In the Project Explorer view on the left, select the **`helloworld`** organizational unit, the **`helloworldrepo`** repository, and the **`HelloWorld`** project.

3. On the perspective menu, click **New Item** → **Business Process** and define the process definition details:

   - Enter the name of the business process as **`HelloWorld`**.

   - Select the **`org.bpms.helloworld`** package. This defines the location in the artifact repository, where you will create your business process definition.

   PACKAGE SELECTION

   Make sure to select the package **`org.bpms.helloworld`**. Selection of an incorrect package causes the deployment to fail.

4. Click **Ok**.

The Process Designer with the canvas of the created process definition opens.

5.2. MODELING BUSINESS PROCESS

After you have created the business process definition, you can design the business process in the Business Process Designer: the Business Process Designer should be open in a tab on the right. If you closed the tab, in the Project Explorer, click **`HelloWorld`** under **BUSINESS PROCESSES** to re-open the Business Process Designer. Make sure you select the correct package (**`org.bpms.helloworld`**). To design the process content, do the following:

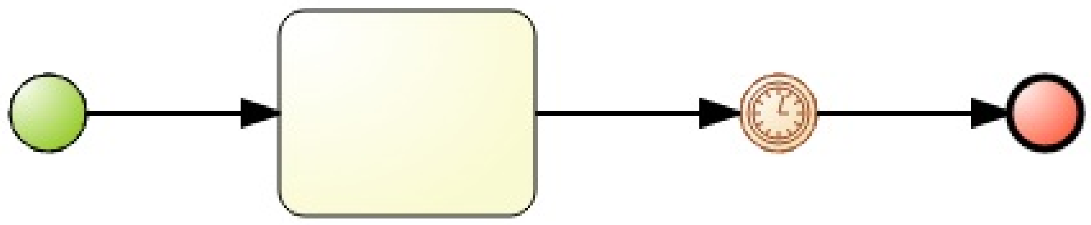
1. Expand the **Object Library** palette with Process Elements: click the double-arrow button (») in the upper left hand corner of the Business Process Designer tab. A Start Event element appears on the canvas.

2. Click the Start Event element: the quick linker items appear around the node. Click the **Task** (□) icon to create an outgoing Sequence Flow and a Task element connected to the Start Event.

3. Create a Timer Event: from the palette, drag-and-drop the **Timer** (under **Catching Intermediate Events**) and connect the Task to the Timer Element using the quick linker menu.

4. Create the End Event element connected to the Timer Event element (either using the quick linker feature or dragging it from the palette).

Figure 5.1. Draft of the HelloWorld Process



5.3. DEFINING ELEMENT PROPERTIES

Now, you need to define the element properties:

1. Click the double-arrow («) in the upper right hand corner to expand the **Properties** tab.

2. Click anywhere on the canvas. The **Properties** tab displays the properties of the Hello World process. The required properties are pre-defined; however, you can change their values if necessary.

3. The Start Element and End Element have only optional properties. However, Task Elements need their type and then type-specific properties defined: click the Task Element and define the required properties:

   - **Name**: `ScriptTask`

   - **Task Type**: `Script`

   - **Script Language**: **`java`**

   - **Script**: **`System.out.println("Hello World!");`**

4. Click **Ok**.
   You have defined a Task that will perform a script on execution. The script is defined in Java and executes the method **`System.out.println("Hello World!")`**, that means it will write **`Hello World!`** to the standard output of the server. By default, the standard output of the server is the console the server was started in.

5. You still need to define properties of the Timer Event to make the process wait on runtime, so you can inspect its execution: click on the Timer element and define the time it will wait for execution:

   - **Time Duration**: **`1m`**

   Click **Ok**. The process will therefore wait for 1 minute on the Timer Event after it executes the Script Task.

6. Give the Timer Event a name using the same tab and then save your process: click **Save** at the top of the Business Process Editor.

7. The **Save this item** prompt is displayed. Enter the**Check-in comment**.

NOTE

The check-in comment would be a short description of the kind of change you made and has to be entered every time an asset is saved.

8. Click ⚠ on the toolbar of the Process Designer with the process to make sure the defined process is valid. The outline of any element with one or multiple validation mistakes turns orange. For information on the validation feature, see the *Red Hat JBoss BPM Suite User Guide*.

Further information on Process Elements and their properties is available in the *Red Hat JBoss BPM Suite User Guide*.

5.4. BUILDING AND DEPLOYING

Before you can instantiate your Hello World process, you need to build the entire project and deploy it to the Execution Server:

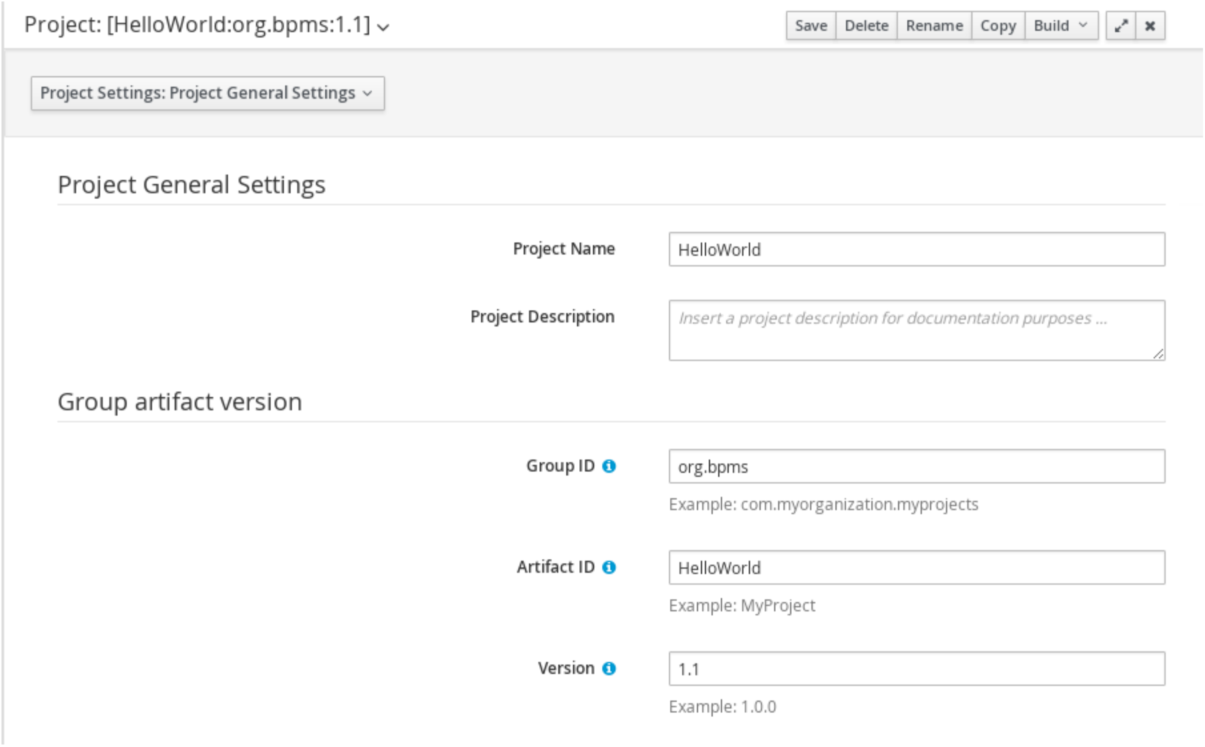
1. On the main menu of Business Central, go to **Authoring → Project Authoring**.

2. In the Project Explorer, locate the Hello World project.

3. Open the project in the Project Editor: in Project Explorer, click **Open Project Editor**.

4. Check that the correct project details are displayed on the project screen and choose **Build → Build & Deploy** in the upper right hand corner of the project screen view.

Figure 5.2. Project Editor with the HelloWorld Project Properties



Project: [HelloWorld:org.bpms:1.1]

Project General Settings

Project Name: HelloWorld

Project Description: Insert a project description for documentation purposes ...

Group artifact version

Group ID: org.bpms
Example: com.myorganization.myprojects

Artifact ID: HelloWorld
Example: MyProject

Version: 1.1
Example: 1.0.0

A green notification appears in the upper part of the screen informing you that the project has been built and deployed to the Execution Server and is available for instantiation.

5.5. INSTANTIATING BUSINESS PROCESS

To create an instance of the Hello World process, that is, to run the business process, do the following:

1. Click **Process Management** → **Process Definitions** on the main menu.

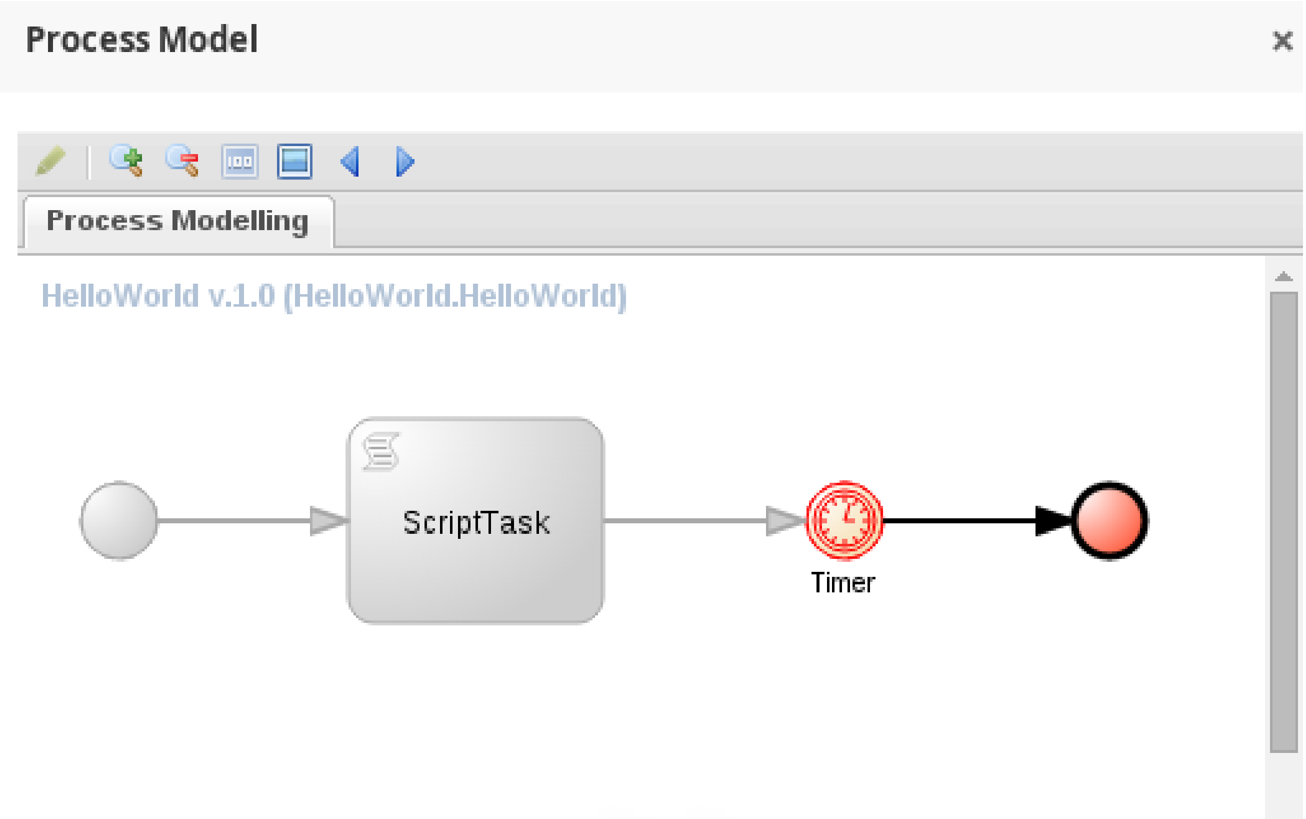
2. On the displayed **Process Definitions** tab, locate the **`HelloWorld`** process. You might need to click the **Refresh** button to have the deployment displayed on the list.

3. Click the **Start** (Start) button next to the process definition entry and in the dialog box, click **Submit** to confirm that you want to instantiate the process.
   The process is instantiated with the currently logged-in user as the process owner and the process form is displayed. The form can serve to request input from a user on process instantiation if defined; see the *Red Hat JBoss BPM Suite User Guide*for further information.

The **Process Instance Details** view is displayed with details about the started process instance. The **`Hello World!`** message appears in standard output, typically the terminal emulator in which the server was started. Then the process instance waits on the Timer Event.

Click **Options** → **Process Model** to see the current execution status.

Figure 5.3. HelloWorld Realtime Execution Diagram: the Timer Event Being Executed



5.6. ABORTING BUSINESS PROCESS

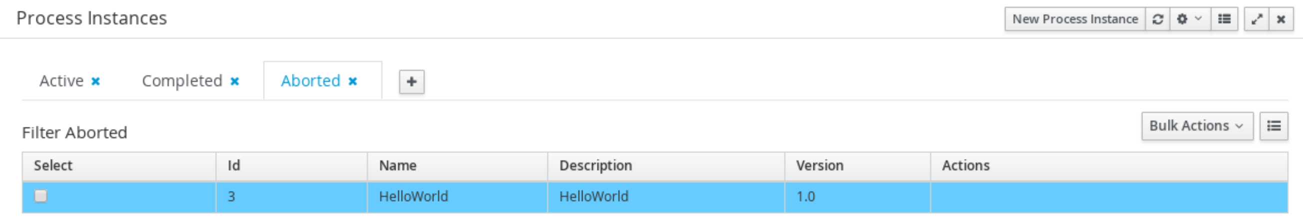
At the moment, an instance of the Hello World process is running on the Execution Server. You can check in the **Process Instances** view: to display the view, go to**Process**

Management → **Process Instances**.

From here, you can view the instance details by clicking on the row of a chosen process. Also, you can perform basic administration actions, such as sending a signal to a process instance (Signal) or aborting it (Abort).

After a business process is instantiated, a process instance is waiting for the Timer Event. To abort the instance, click Abort on the row with the instance information. The instance will disappear from the list of **Active** process instances and appear in the**Aborted** list.

Figure 5.4. List of Aborted Process Instances



Select	Id	Name	Description	Version	Actions
□	3	HelloWorld	HelloWorld	1.0	

CHAPTER 6. HELLO WORLD BUSINESS RULE

Since Red Hat JBoss BPM Suite comes with Red Hat JBoss BRMS integrated, this chapter provides an introduction to how business rules work and integrate into Red Hat JBoss BPM Suite.

You will add a new Business Rule Task to the Hello World process, which will fire (check) a business rule of a particular business rule group.

Note that this is a very simple integration case and in production you might need to use a more advanced concept and techniques, such as decision tables. For more information on business rules and BRMS see the *Red Hat JBoss BRMS User Guide*.

6.1. CREATING BUSINESS RULE

Business rules are defined as when-then statements: when something is true, then something happens (for example, when the person is older than 18, they can access the particular content). Business rules are stored in DRL files.

To define the **helloworldrule** business rule in the Hello World project, do the following:

1. In Project Explorer, select the **helloworld** organizational unit, the **helloworldrepo** repository, the **HelloWorld** project, and the **default** package.

   PACKAGE SELECTION

   Make sure to select the **default** package. Selection of an incorrect package causes the deployment to fail.

2. Create the DRL file:

   a. On the perspective menu, click **New Item** → **DRL file**.

   b. In the **Create new DRL file** dialog box, define the resource name as **helloworldrule** and check that the package is set to**org.bpms.helloworld**.

   c. Click **Ok**.

3. In the displayed DRL editor with the **helloworldrule.drl** file, define the rule:

```
rule "helloworldrule"
ruleflow-group "helloworldgroup"

when
then
  System.out.println("Hello World!");
end
```

   This rule does not define any when condition, that is, it will always be true when fired (called to execute) and the **Hello World!** phrase will be printed.

4. Click **Save**.

5. The **Save this item** prompt is displayed. Enter the**Check-in comment** and click **Save**.

6.2. ADDING BUSINESS RULE TASK

A Business Rule Task is a task which fires rules that belong to a certain ruleflow group.

To add a Business Rule Task to your process, do the following:

1. Open your Business Process in Process Designer: in Project Explorer, select the `HelloWorld` project, the `org.bpms.helloworld` package, click on **BUSINESS PROCESSES** and choose the `HelloWorld` process.

   PACKAGE SELECTION

   Make sure to select the `org.bpms.helloworld` package. Selection of an incorrect package causes the deployment to fail.

2. In the displayed Business Process Designer with the Hello World process, expand the **Object Library** palette with Process Elements: click the double-arrow button (») in the upper left hand corner of the Business Process Designer tab.

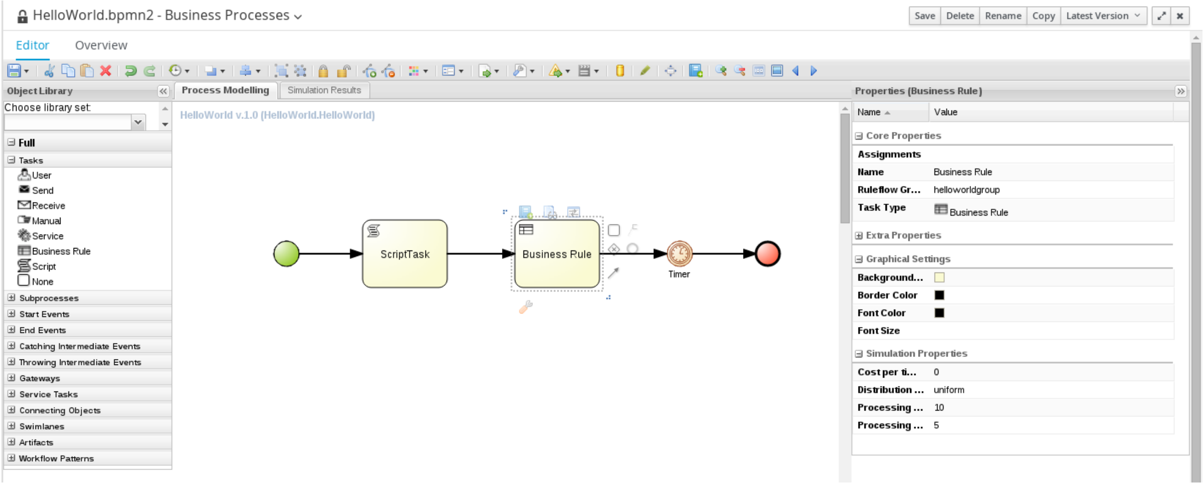
3. Expand the **Tasks** menu and drag-and-drop the Business Rule Task to the canvas on the right.
   Adjust flow connections so as to integrate the task into the workflow.

4. Select the Business Rule Task and define its properties in the **Properties** panel:

   - **Name**: `BusinessRule`

   - **Ruleflow Group**: `helloworldgroup`
     The **Ruleflow Group** property defines the group of rules that should be fired when the task is being executed. In this example, only one rule, the `helloworldrule` rule defined in the`helloworldrule.drl` file, is present in the `helloworldgroup` group.

   Figure 6.1. The HelloWorld Process with the Business Rule Task and Its Properties



Name	Value
Core Properties	
Assignments	
Name	Business Rule
Ruleflow Gr...	helloworldgroup
Task Type	Business Rule
Extra Properties	
Graphical Settings	
Background...	
Border Color	■
Font Color	■
Font Size	
Simulation Properties	
Cost per ti...	0
Distribution ...	uniform
Processing ...	10
Processing ...	5

5. Click the save button to save the changes.

6.3. BUILDING AND DEPLOYING

Build the entire project and deploy it to the Execution Server:

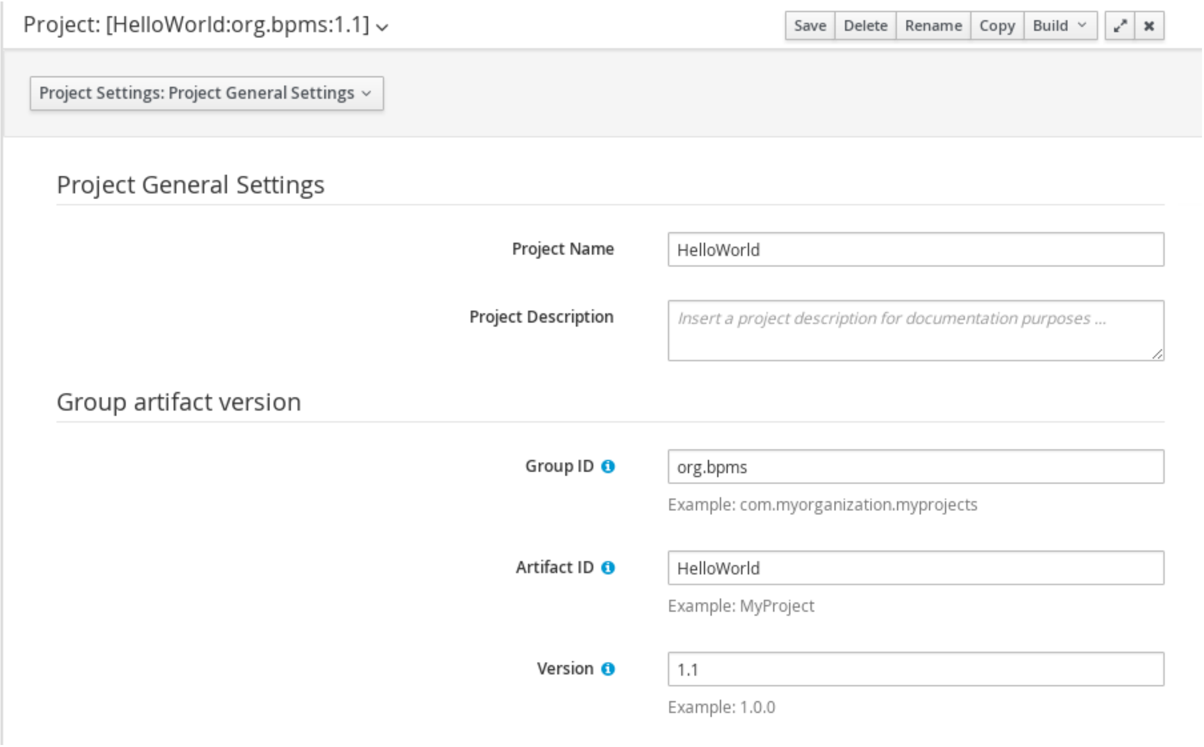
1. On the main menu of Business Central, go to **Authoring** → **Project Authoring**.

2. In the Project Explorer, locate the Hello World project.

3. Open the project in the Project Editor: in Project Explorer, click **Open Project Editor**.

4. Check that the correct project details are displayed on the project screen and click **Build** → **Build & Deploy** in the upper right hand corner of the project screen view.

   Figure 6.2. Project Editor with the HelloWorld Project Properties

   Project: [HelloWorld:org.bpms:1.1]

   Project General Settings

   Project Name: HelloWorld

   Project Description: *Insert a project description for documentation purposes ...*

   Group artifact version

   Group ID: org.bpms
   Example: com.myorganization.myprojects

   Artifact ID: HelloWorld
   Example: MyProject

   Version: 1.1
   Example: 1.0.0

   A green notification appears in the upper part of the screen informing you that the project has been built and deployed to the Execution Server and is available for instantiation.

NOTE

The previous version of the Hello World deployment is substituted with the new one containing the Business Rule Task. To keep both deployments, change the version number of the project in the Project Editor.

6.4. INSTANTIATING BUSINESS PROCESS

To create an instance of the Hello World process — that is to run the business process — do the following:

1. Click **Process Management** → **Process Definitions** on the main menu.

2. On the displayed **Process Definitions** tab, locate the `HelloWorld`. You might need to click **Refresh** to have the deployment displayed on the list.

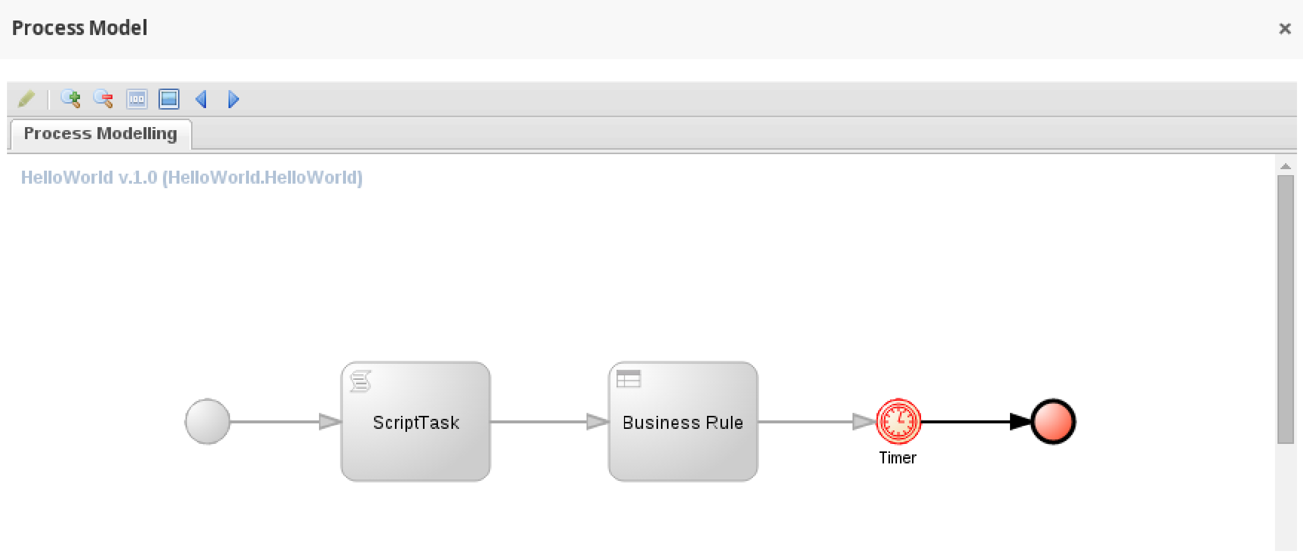
3. Click the **Start** (Start) next to the process definition entry and in the dialog box, click **Submit** to confirm that you want to instantiate the process.

> The process is instantiated with the currently logged-in user as the process owner and the process form is displayed (the form can serve to request input from a user on process instantiation if defined; see the *Red Hat JBoss BPM Suite User Guide*).

The **Process Instance Details** view is displayed with details about the started process instance. The **`Hello World!`** message appears in standard output twice, typically the terminal emulator in which the server was started. Then the process instance waits on the Timer Event.

Choose **Options** → **Process Model** to see the current execution status.

Figure 6.3. HelloWorld Realtime Execution Diagram: the Timer Event Being Executed



CHAPTER 7. BAM

7.1. ACCESSING RED HAT JBOSS BPM SUITE DASHBUILDER

Dashbuilder is the Red Hat JBoss BPM Suite web-based user interface for Business Activity Monitoring. To access the Dashbuilder from Business Central, go to **Dashboards** → **Process & Task Dashboards**.

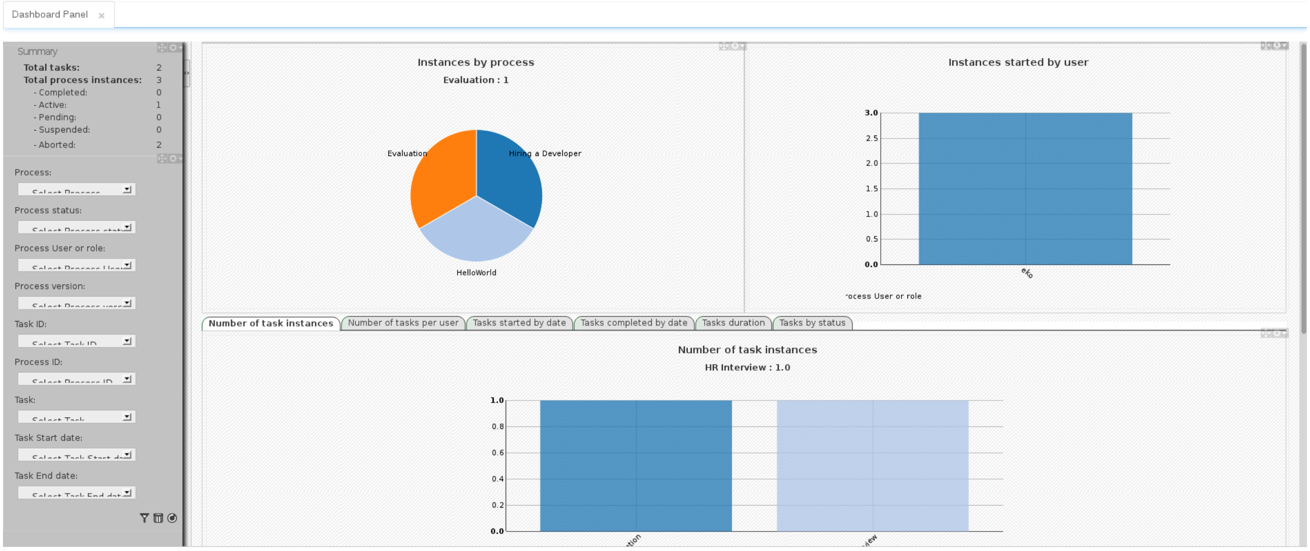
The displayed dashboard provides statistics on runtime data selected on the left. You can create your own dashboard in the Dashbuilder. To do so, display the Dashbuilder by clicking **Dashboards** → **Business Dashboards**.

7.2. MONITORING INSTANCES

The Dashbuilder provides a special dashboard that allows you to monitor the status of runtime data (process instances and tasks) on your Execution Engine.

To view this data, go to **Dashboards** → **Process & Task Dashboard**.

Figure 7.1. Process and Task Dashboard



Summary

Total tasks: 2
Total process instances: 3
- Completed: 0
- Active: 1
- Pending: 0
- Suspended: 0
- Aborted: 2

On the left panel, you can select an entity to display the statistics for. The charts and data on the right is then updated accordingly.

CHAPTER 8. 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 Red Hat 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"**.

8.1. JBOSS CENTRAL

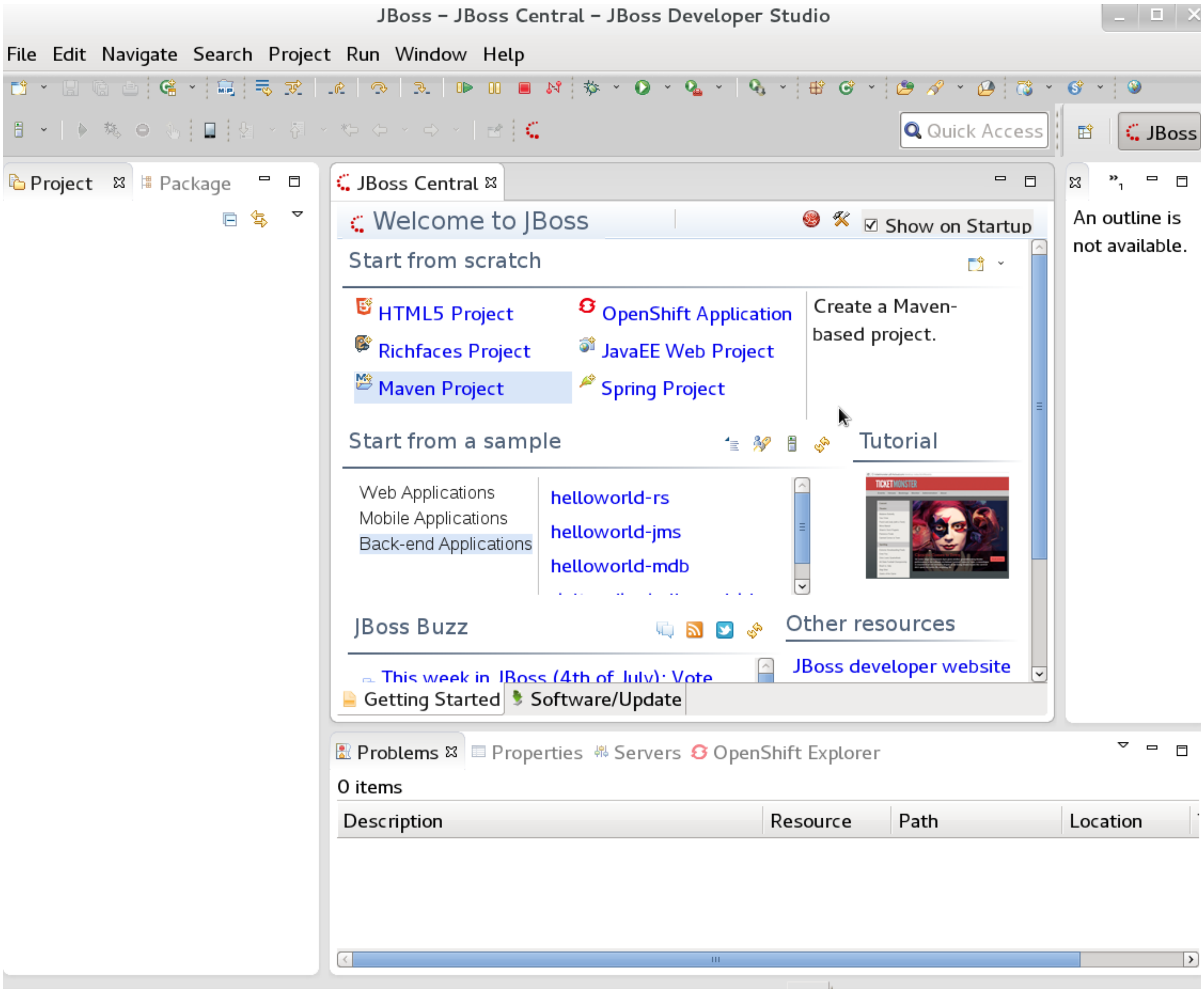
When Red Hat JBoss Developer Studio 8.0 is first started, JBoss Central is displayed in the main window of the workbench. From JBoss Central it is possible to create new projects by selecting the menu options under **Start from scratch**. Example projects can be started by selecting the links under **Start from a sample**.

NOTE

Due to issues in GWT3, JBDS should be configured to not use it. You can do this by setting the system property to **0**:

```
# Do not use GTK 3.
export SWT_GTK3=0
```

Figure 8.1. JBoss Central



Create a Maven-based project.

An outline is not available.

0 items

Description	Resource	Path	Location

8.2. INSTALLING RED HAT JBOSS DEVELOPER STUDIO PLUG-INS

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

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

1. Start Red Hat JBoss Developer Studio.
2. Go to **Help** → **Install New Software**.
3. Click **Add** to enter the **Add Repository** menu.
4. Give the software site a name next to **Name** field and add the following URL in the **Location** field: `https://devstudio.jboss.com/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. After installation of the plug-ins has completed, restart Red Hat JBoss Developer Studio.

8.3. SETTING DROOLS RUNTIME

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

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

Procedure: Configuring Red Hat JBoss BRMS Runtime

1. Extract the runtime JAR files located in the **`jboss-brms-`*`VERSION`*`-engine.zip`** archive that you can download from Red Hat Customer Portal.

2. From the Red Hat JBoss Developer Studio menu, select **Window** and click **Preferences**.

3. Select **Drools** → **Installed Drools Runtimes**.

4. Click **Add...**; provide a name for the new runtime, and click**Browse** to navigate to the directory where you extracted the runtime files in the first step. Click **OK** to register the selected runtime in JBDS.

5. Mark the runtime you have created as the default Drools runtime by clicking on the check box next to it.

6. Click **OK**. If you have existing projects, a dialog box will indicate that you have to restart Red Hat JBoss Developer Studio to update the Runtime.

8.4. SETTING RED HAT JBOSS BPM SUITE RUNTIME

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

A runtime is a collection of JAR files that represent a specific release of the software.

If you have previously downloaded the Red Hat JBoss BPM Suite Generic Deployable ZIP archive from Red Hat Customer Portal, the JAR files that make up the runtime are located in the **`jboss-bpms-engine.zip`** archive.

Procedure: Configuring jBPM Runtime

1. From the Red Hat JBoss Developer Studio menu, select **Window** and click **Preferences**.

2. Select **jBPM** → **Installed jBPM Runtimes**.

3. Click **Add...** ; provide a name for the new runtime, and click**Browse** to navigate to the directory where the runtime is located.

4. Click **OK**, select the new runtime and click**OK** again. If you have existing projects, a

dialog box will indicate that you have to restart Red Hat JBoss Developer Studio to update the Runtime.

8.5. 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** → **Perspective** → **Open Perspective** → **Other**, then **jBPM**, and click **OK**.

2. Add the **Server** view by selecting **Window** → **Show View** → **Other…** and then **Server** → **Servers**.

3. Open the server menu by right clicking the **Servers** panel and select **New** → **Server**.

4. Define the server by selecting **JBoss Enterprise Middleware** → **JBoss Enterprise Application Platform 6.1+** and clicking **Next**.

5. Set the home directory by clicking **Browse**. Navigate to and select the installation directory for Red Hat 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**.

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

NOTE

This is an additional feature not required for working with 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 **Server** tab and clicking 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.

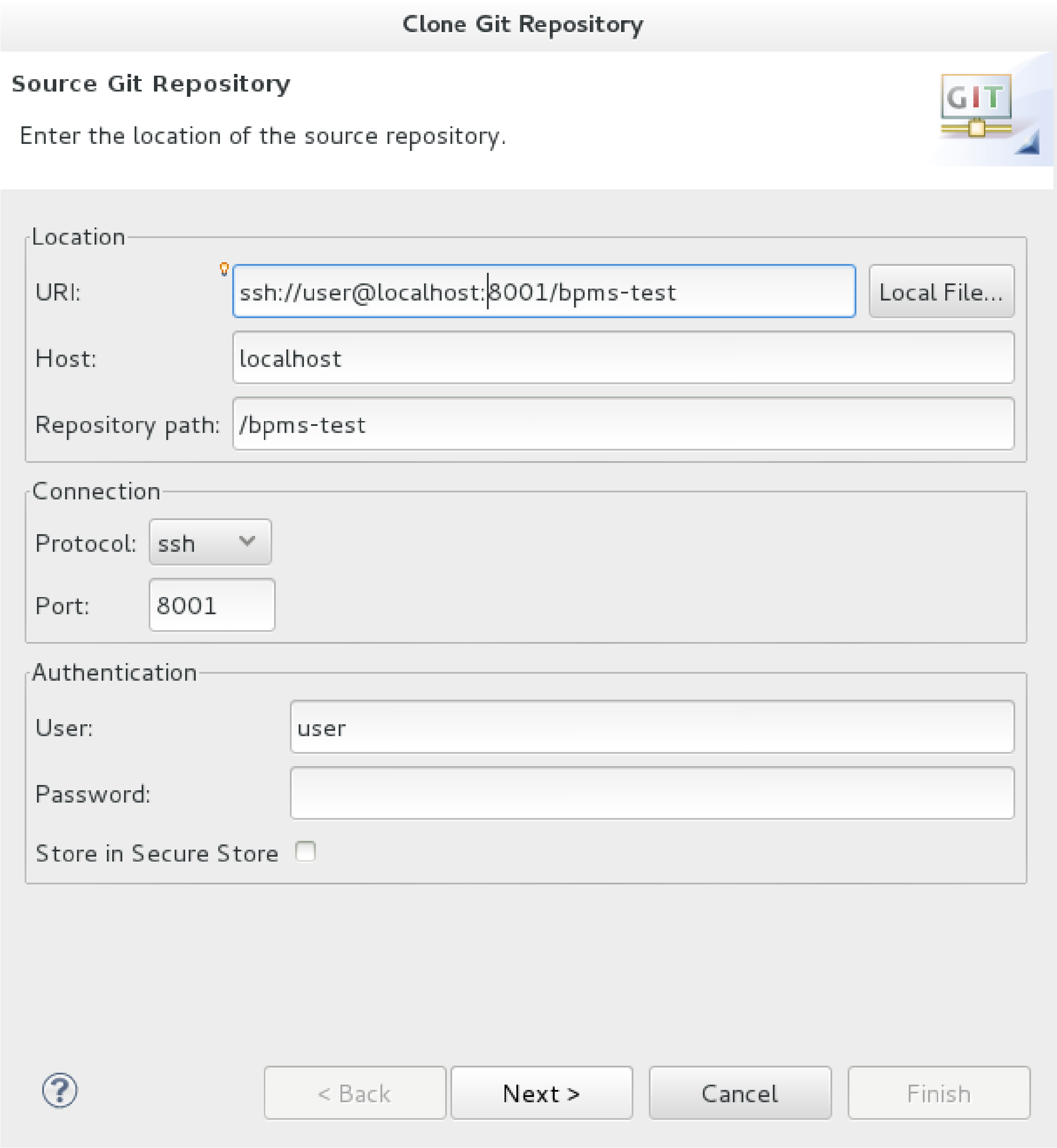
   ```
   /sbin/service sshd start
   ```

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**.

Figure 8.2. Git Repository Details



Clone Git Repository

Source Git Repository

Enter the location of the source repository.

Location

URI: ssh://user@localhost:8001/bpms-test

Host: localhost

Repository path: /bpms-test

Connection

Protocol: ssh

Port: 8001

Authentication

User: user

Password:

Store in Secure Store □

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

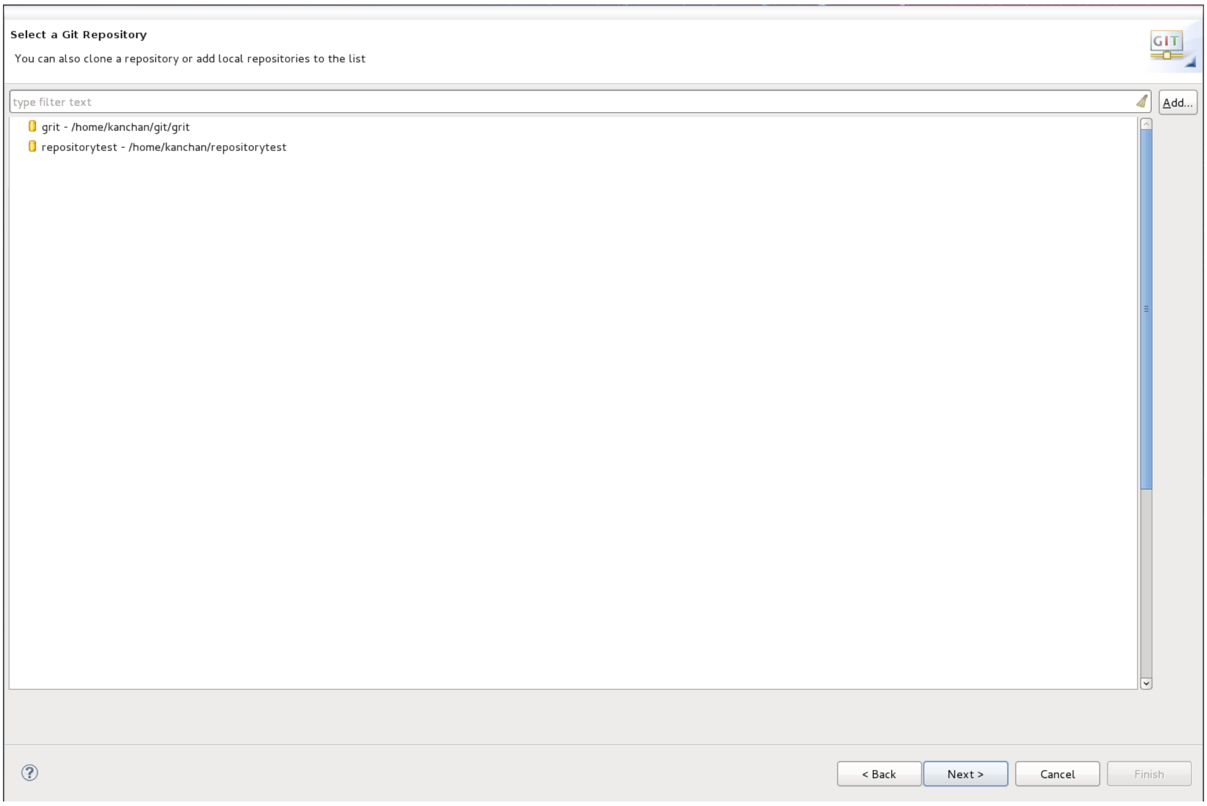
1. Start the Red Hat JBoss BPM Suite server by selecting the server from the **Server**

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**.

Figure 8.3. Git Repository Details



Select a Git Repository

You can also clone a repository or add local repositories to the list

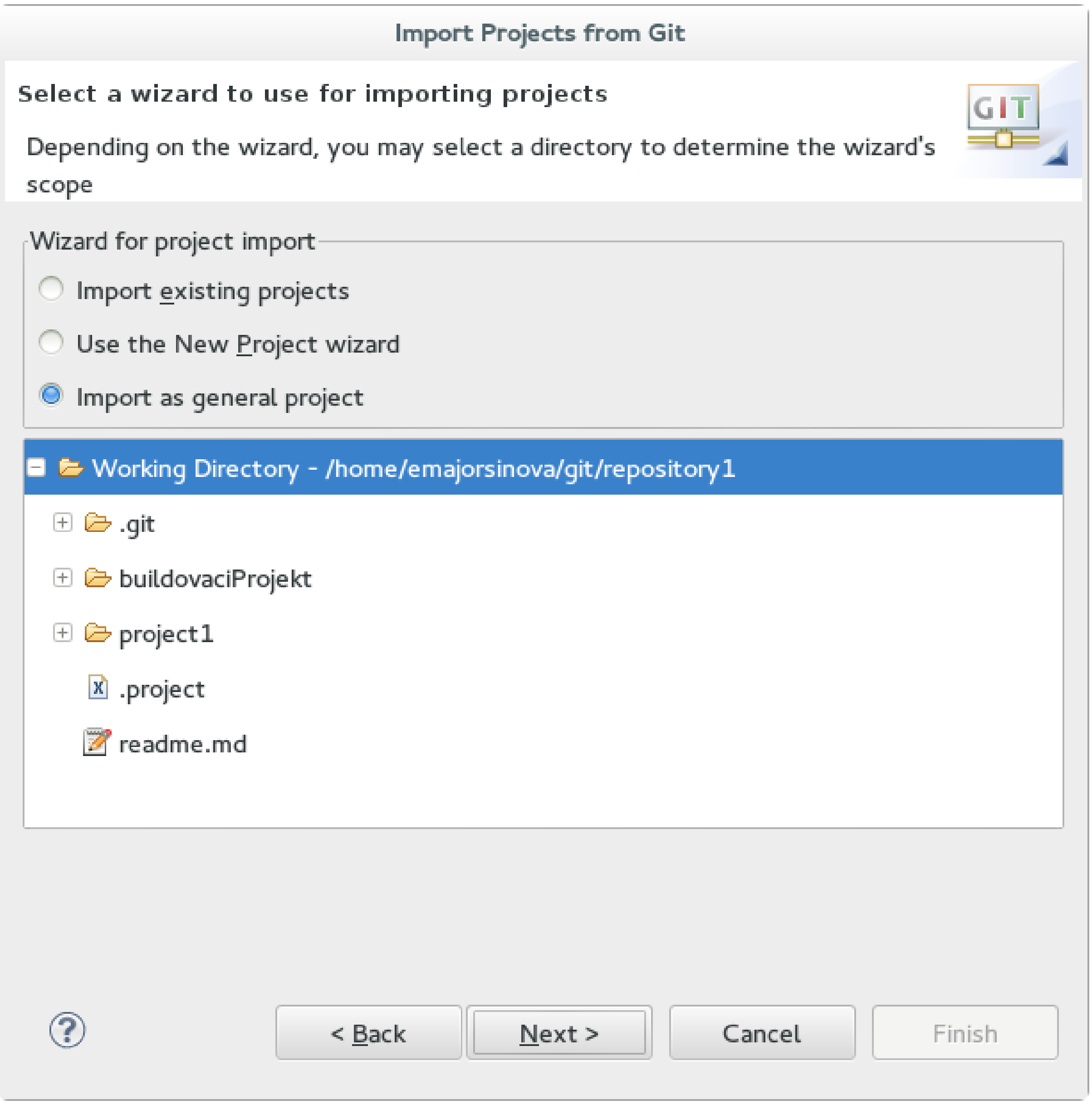
grit - /home/kanchan/git/grit

repositorytest - /home/kanchan/repositorytest

4. Select the repository that is to be configured from the list of available repositories and click **Next**.

5. In the dialog that opens, select the radio button **Import as general project** from the **Wizard for project import group** and click **Next**. Name the project and click **Finish**.

Figure 8.4. Wizard for Project Import



Import Projects from Git

Select a wizard to use for importing projects

Depending on the wizard, you may select a directory to determine the wizard's scope

Wizard for project import

○ Import existing projects

○ Use the New Project wizard

● Import as general project

- Working Directory - /home/emajorsinova/git/repository1
  - .git
  - buildovaciProjekt
  - project1
  - .project
  - readme.md

8.7. CREATING DROOLS PROJECT

Procedure: Creating New Red Hat JBoss Developer Studio Project

1. From the main menu, select **File** → **New** → **Project**.

2. Select **Drools** → **Drools Project** and click **Next**.

3. Enter a name for the project into the **Project name:** text box and click **Next**.

NOTE

Red Hat JBoss Developer Studio provides the option to add a sample Hello World rule file to the project. Accept this default by clicking **Next** to test the sample project in the following steps.

4. Select the Drools runtime (or use the default).

5. Select code compatible with **Drools 6.0.*x***. Enter a **GroupID**, an **ArtifactID**, and **Version**, and click **Finish**.

6. To test the project, right click the Java file that contains the main method and select **Run** → **run as** → **Java Application**.
   The output will be displayed on the console tab.

8.8. CREATING RED HAT JBOSS BPM SUITE PROJECT

Procedure: Creating New Red Hat JBoss BPM Suite Project in Red Hat JBoss Developer Studio

1. From the main menu, select **File** → **New** → **Project**.

2. Select **jBPM** → **jBPM Project** and click **Next**.

3. Enter a name for the project into the **Project name:** text box and click **Next**.

   SAMPLE HELLOWORLD RULE FILE

   Red Hat JBoss Developer Studio provides the option to add a sample Hello World rule file to the project. Accept this default by clicking **Next** to test the sample project in the following steps.

4. Select the jBPM runtime (or use the default).

5. Select generate code compatible with **jBPM 6 or above**, and click **Finish**.

6. To test the project, right click the Java file that contains the main method and select **Run** → **run as** → **Java Application**.
   The output will be displayed in the console tab.

CHAPTER 9. BUSINESS RESOURCE PLANNER

Business Resource Planner is a lightweight, embeddable planning engine that optimizes planning problems. It helps normal Java™ programmers solve planning problems efficiently, and it combines optimization heuristics and metaheuristics with very efficient score calculations.

Planner helps solve various use cases like the following:

- *Employee/Patient Rosters*: planner helps create timetables for nurses and keeps track of patient bed management.
- *Educational Timetables*: planner helps schedule lessons, courses, exams, and conference presentations.
- *Shop Schedules*: planner tracks car assembly lines, machine queue planning, and workforce task planning.
- *Cutting Stock*: planner minimizes waste by reducing the consumption of resources such as paper and steel.

9.1. INSTALLING BUSINESS RESOURCE PLANNER

1. Navigate to the Red Hat Customer Portaland log in with your user credentials.
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 version**6.3.0**.
5. Select **Red Hat JBoss BPM Suite 6.3 Business Resource Planner** and click **Download**.

9.2. RUNNING BUSINESS RESOURCE PLANNER EXAMPLES

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

   ```
   ./runExamples.sh
   ```

   In a Windows environment, run the following command:

   ```
   ./runExamples.bat
   ```

3. Pick an example from the **Examples** GUI application that opens and run it in your favorite IDE.

APPENDIX A. REVISION HISTORY

Note that revision numbers relate to the edition of this manual, not to version numbers of Red Hat JBoss BRMSBPM Suite.

Revision 6.3.0-17 **Mon Mar 20 2017** **Tomas Radej**

Rebuilt.

Revision 6.3.0-16 **Wed Feb 22 2017** **Tomas Radej**

Rebuilt.

Revision 6.3.0-15 **Fri Dec 23 2016** **Tomas Radej**

Rebuilt.

Revision 6.3.0-14 **Mon Nov 28 2016** **Tomas Radej**

Rebuilt.

Revision 6.3.0-13 **Wed Oct 5 2016** **Tomas Radej**

Rebuilt.

Revision 6.3.0-12 **Mon Oct 3 2016** **Tomas Radej**

Built for release 6.3.3.

Revision 6.3.0-11 **Thu Sep 15 2016** **Tomas Radej**

Rebuilt.

Revision 6.3.0-10 **Thu Sep 15 2016** **Tomas Radej**

Published the AsciiDoc version of the docs.

Revision 6.3.0-8 **Thu Jul 14 2016** **Tomas Radej**

Updated documentation with release 6.3.1.

Revision 6.3.0-7 **Thu Jun 2 2016** **Marek Czernek**

Releasing the newest documentation.

Revision 6.3.0-5 **Thu May 5 2016** **Tomas Radej**

Built with live links.

Revision 6.3.0-4 **Thu May 5 2016** **Tomas Radej**

Bootstrapping links.

Revision 6.3.0-3 **Thu Apr 28 2016** **Tomas Radej**

All books rebuilt.

Revision 6.3.0-2 **Thu Apr 28 2016** **Tomas Radej**

All books rebuilt.

Revision 6.3.0-1 **Thu Apr 28 2016** **Tomas Radej**

Initial build for release 6.3.0 of JBoss BPM SuiteJBoss BRMS.