



Red Hat JBoss BPM Suite 6.1

Getting Started Guide

The Getting Started Guide for Red Hat JBoss BPM Suite

Red Hat JBoss BPM Suite 6.1 Getting Started Guide

The Getting Started Guide for Red Hat JBoss BPM Suite

Kanchan Desai
kadesai@redhat.com

Doug Hoffman

Eva Kopalova

Red Hat Content Services

Legal Notice

Copyright © 2015 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-sa/3.0/). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Abstract

This guide is to help users install and set up Red Hat JBoss BPM Suite for the first time.

Table of Contents

CHAPTER 1. QUICK START WITH RED HAT JBOSS BPM SUITE	3
CHAPTER 2. INSTALLATION OPTIONS	4
2.1. THE EAP6 BUNDLE INSTALLATION	4
2.2. THE GENERIC DEPLOYABLE BUNDLE INSTALLATION	8
2.3. STARTING THE SERVER	16
2.4. JAVA SECURITY MANAGER AND PERFORMANCE MANAGEMENT	17
CHAPTER 3. LOGGING ON TO BUSINESS CENTRAL	18
CHAPTER 4. HELLO WORLD PROJECT	19
4.1. CREATING REPOSITORY STRUCTURE	19
CHAPTER 5. HELLO WORLD PROCESS	21
5.1. CREATING BUSINESS PROCESS	21
5.2. MODELING BUSINESS PROCESS	21
5.3. DEFINING ELEMENT PROPERTIES	22
5.4. BUILDING AND DEPLOYING	23
5.5. INSTANTIATING BUSINESS PROCESS	23
5.6. ABORTING BUSINESS PROCESS	24
CHAPTER 6. HELLO WORLD BUSINESS RULE	26
6.1. CREATING THE BUSINESS RULE	26
6.2. ADDING THE BUSINESS RULE TASK	27
6.3. BUILDING AND DEPLOYING	27
6.4. INSTANTIATING BUSINESS PROCESS	28
CHAPTER 7. BAM	30
7.1. ACCESSING RED HAT JBOSS BPM SUITE DASHBUILDER	30
7.2. MONITORING INSTANCES	30
CHAPTER 8. RED HAT JBOSS DEVELOPER STUDIO	31
8.1. JBOSS CENTRAL	31
8.2. INSTALLING THE JBOSS DEVELOPER STUDIO PLUG-INS	32
8.3. SETTING THE DROOLS RUNTIME	33
8.4. SETTING THE JBOSS BPM SUITE RUNTIME	33
8.5. CONFIGURING THE JBOSS SERVER	33
8.6. IMPORTING PROJECTS FROM A GIT REPOSITORY INTO JBOSS DEVELOPER STUDIO	34
8.7. CREATING A DROOLS PROJECT	37
8.8. CREATING A JBOSS BPM SUITE PROJECT	38
CHAPTER 9. BUSINESS RESOURCE PLANNER	39
9.1. INSTALLING BUSINESS RESOURCE PLANNER	39
9.2. RUNNING THE BUSINESS RESOURCE PLANNER EXAMPLES	39
APPENDIX A. REVISION HISTORY	40

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.

Procedure 1.1. Red Hat JBoss BPM Suite Quick Start

1. Download JBoss BPM Suite installer from access.redhat.com.

2. Run installer by executing the following command:

```
java -jar jboss-bpmsuite-6.1.0.GA-installer.jar
```

3. Step through the GUI installer steps making note of the username/password for accessing the JBoss BPM Suite application after installation.
4. After successful installation, in a command prompt, navigate to the EAP home folder where you have installed JBoss BPM Suite.
5. Start the JBoss BPM Suite 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 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: [Section 2.1.3, “Defining Roles”](#)

Create additional users: [Section 2.1.4, “Creating users”](#)

Create a Hello World Project: [Chapter 4, *Hello World project*](#)

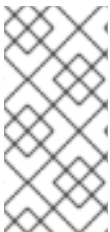
Look up detailed installation instructions for using the installer: *Installation Guide*

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-6.1.0.GA-deployable-eap6.x.zip**: version adapted for deployment on Red Hat JBoss Enterprise Application Platform (EAP 6.4).
 - **jboss-bpmsuite-6.1.0.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 JBoss BPM Suite 6.1 onwards, you must have JBoss EAP 6.4 or better already installed before attempting to install JBoss BPM Suite.

2.1. THE EAP6 BUNDLE INSTALLATION

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



NOTE

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

2.1.1. Downloading the Red Hat JBoss EAP6 Package

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

1. Go to the [Red Hat Customer Portal](#) and log in.
2. Click **Downloads** → **Products Downloads**.
3. In the **Product Downloads** page that opens, click **Red Hat JBoss BPM Suite**.
4. From the **Version** drop-down menu, select version 6.1.0.

- On the **Software Downloads** page that opens, navigate to the **Red Hat JBoss BPM Suite 6.1.0 Deployable for EAP 6** row and click **Download**.

2.1.2. Installing the Red Hat JBoss EAP6 Package

Installation on a fresh JBoss EAP instance

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

- Extract the zip package deployable for JBoss EAP you downloaded from the [Red Hat Customer Portal](#).
- 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 JBoss EAP. This account must not be a superuser account.

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

- 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-bpms-VERSION-TYPE.zip -d SERVER_HOME_PARENT_DIR
```

Example 2.1. The unzip command

```
unzip -u jboss-bpms-6.1.0-redhat-7-deployable-eap6.x.zip -d
/home/john/myServers/
```

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

Installation on an existing JBoss EAP configuration



WARNING

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

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

1. Extract the zip package deployable for 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 JBoss BPM Suite requires JMS, so JMS is added by default into all profiles in **domain.xml** provided by JBoss BPM Suite distribution.)
 - **jboss-eap-6.4/standalone/configuration/*** - (please be aware that JBoss BPM Suite requires JMS, so JMS is added by default into all profiles config files (especially into **standalone.xml** and **standalone-ha.xml**) provided by JBoss BPM Suite distribution.)
 - **jboss-eap-6.4/modules/layers.conf**
 - **jboss-eap-6.4/bin/product.conf**
3. Ensure the target 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 JBoss BPM Suite distribution.
4. Make sure no JBoss EAP module layer is already called 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, refer to [Chapter 2. Business Central configuration](#). For further information about roles and task interactions, refer to [Appendix 11.5 User Task](#).

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.

1. Run

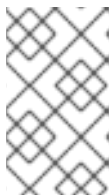
```
./add-user.sh
```

on a Unix system or

```
add-user.bat
```

on a Windows system from the **bin** directory.

2. Enter **b** to select an Application User at the type of user prompt and press Enter.
3. Accept the default Realm (ApplicationRealm): by pressing Enter.
4. At the username prompt, enter a user name and confirm. For example: **helloworlduser**.



NOTE

Make sure that the usernames don't conflict with any known groups. For example, if there is a group called **admin**, you should not create a user with the username **admin**.

5. Create the user's password at the password prompt and reenter the password. For example: **HelloWorld@123**.



NOTE

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

6. Enter a comma separate list of roles the user will need at the roles prompt (refer to [Section 2.1.3, “Defining Roles”](#)).
7. Confirm you want to add the user.
8. Enter **yes** at the next prompt (this is to enable clustering in the future if required).

2.2. THE 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-6.1.0.GA-deployable-generic.zip**: contains the **business-central.war**, **dashbuilder.war** and **kie-server.war** web applications.
- **jboss-bpmsuite-6.1.0.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 the Generic Deployable Package

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

1. Go to the [Red Hat Customer Portal](#) and log in.
2. Click on **Downloads**.
3. From the list of products click on **Red Hat JBoss BPM Suite**.
4. From the **Version** drop-down menu, select version 6.1.0 (if not already selected).
5. In the Software Downloads section that comes up, navigate to the **Red Hat JBoss BPM Suite 6.1.0 Deployable for All Supported Containers** row and then click **Download**.
6. Also navigate to the **Red Hat JBoss BPM Suite 6.1.0 Core Engine** files row and click **Download** to download the JBoss BPM Suite Core Engine files.

2.2.2. Installing the generic deployable package

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

- Set up the database driver and the transaction manager (Bitronix) (refer to [Section 2.2.2.1, “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 (refer to [Section 2.2.2.2, “Setting up Business Central for Red Hat JBoss Web Server 2.1 \(Tomcat 7\)”](#)).

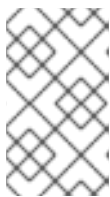
- Set up the Dashbuilder application: set up users and roles and set up persistence (refer to [Section 2.2.2.3, “Setting up Dashbuilder for Red Hat JBoss Web Server 2.0 \(Tomcat 7\)”](#)).

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 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 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/**.



NOTE

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

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

- btm-config.properties

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

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

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

Example 2.2. H2 datasource definition

```
resource.ds1.className=bitronix.tm.resource.jdbc.lrc.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 datasource definition

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

Example 2.4. DB2 Type 4 datasource definition

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

Example 2.5. Oracle datasource definition

```
resource.ds1.className=oracle.jdbc.xa.client.OracleXADataSource
resource.ds1.uniqueName=jdbc/jbpm
resource.ds1.minPoolSize=0
resource.ds1.maxPoolSize=10
resource.ds1.driverProperties.URL=jdbc:oracle:thin:@//localhost
:1521/bpms
resource.ds1.driverProperties.user=dbuser
resource.ds1.driverProperties.password=dbpassword
resource.ds1.allowLocalTransactions=true
```

Example 2.6. Microsoft SQL Server datasource definition

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

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

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

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

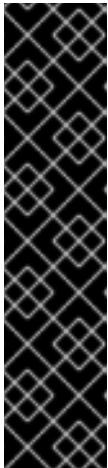
8. Define the btm.root system property and location where bitronix config file is placed:

In **\$TOMCAT_DIR/bin/**, create 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/TransactionSynchronizationRegistr
y"
```

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.



IMPORTANT

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

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

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, refer to the Tomcat 7 documentation).

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



NOTE

Make sure that the usernames don't conflict with any known roles. For example, you should not create a user with the username **admin**.

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

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

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

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

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

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

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

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

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

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

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

**NOTE**

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

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

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

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

Change this value to the datasource defined earlier:

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

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

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

```
./startup.sh
```

- b. Navigate to <http://localhost:8080/business-central> in a web browser.
- c. Login with the username/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 **`$TOMCAT_DIR/conf/tomcat-users.xml`**. Note that Dashbuilder requires users to have the role specified as **admin** and/or **analyst**. If you have already defined these users earlier for Business-Central, you don't need to define them again.
2. Enable single sign-on between Dashbuilder and Business Central by uncommenting the following lines in **`$TOMCAT_DIR/conf/server.xml`** file:

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

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

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

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

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

NOTE

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

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

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

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

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

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

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

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

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

2.3. STARTING THE SERVER

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



NOTE

If you installed 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 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 JBoss BPM Suite application. Use the following guidelines to deploy secure and high performance 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.

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 **helloworld** 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, refer to the *Red Hat JBoss BPM Suite User Guide*).

To create the repository structure, do the following:

1. Open Business Central in your web browser (if running locally <http://localhost:8080/business-central>) and log in as a user with the **admin** role (**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 **Create Repository** dialog box, define the repository properties:
 - Repository Name: **helloworldrepo**
 - Organizational Unit: **helloworld**
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**.

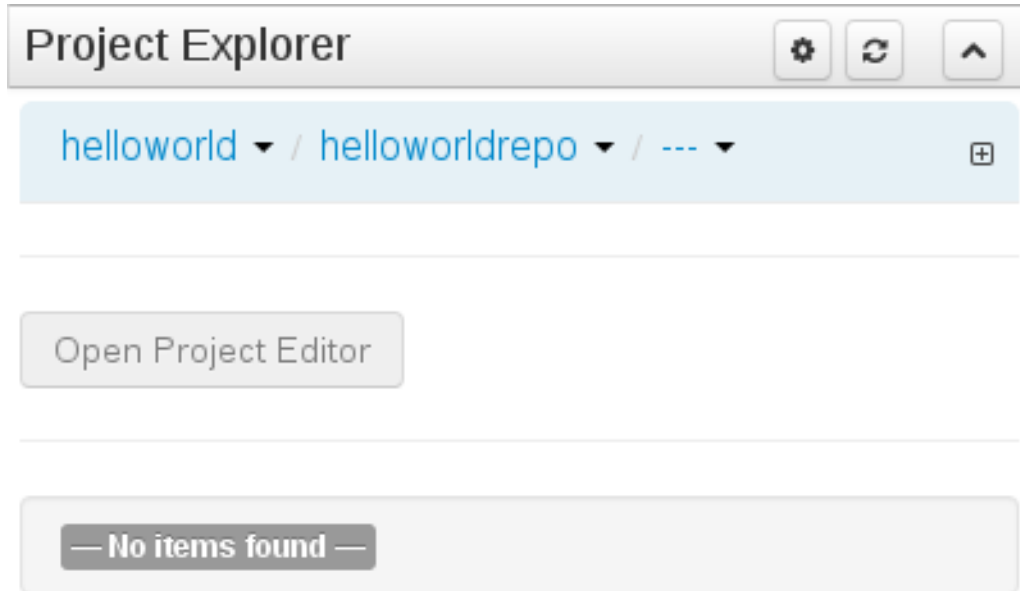


Figure 4.1. Selecting the helloworldrepo repository in the helloworld organizational group in Project Explorer

6. Create a project in the **helloworld** repository:
 - a. On the perspective menu, go to **New Item** → **Project**
 - b. In the displayed **Create new Project** dialog box, define your Project properties:
 - Resource Name (project name): **HelloWorld**
 - c. In the **New Project Wizard** dialog, define the maven properties of the Project:
 - Group ID: **org.bpms**
 - Artifact ID: **HelloWorld**
 - Version ID: **1.0**
 - d. 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.



NOTE



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.
2. A Start Event element appears on the canvas.
3. 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.
4. 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.

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

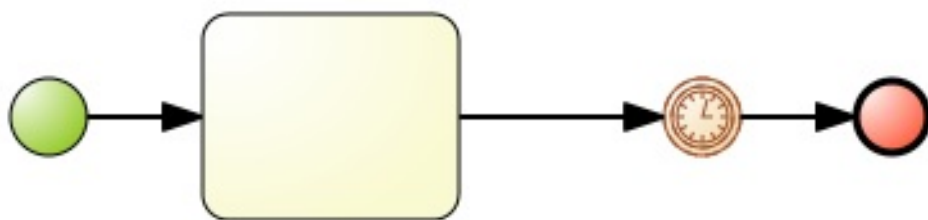



Figure 5.1. Draft of the HelloWorld Process

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 **HelloWorld** 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!");`**

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).

4. You still need to define properties of the Timer Event to make the process wait on runtime, so you can inspect its execution: click the Timer Element and define the time it will wait for execution:
 - Edit Time Duration: **1m** and press Enter.

The Process will therefore wait for 1 minute on the Timer Event after it executes the Script Task.


5. Give the Timer Event a name via the properties, and then save your Process: open the **Save** menu () and click **Save**.

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



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.

7. Click the () button 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 (refer to the *Red Hat JBoss BPM Suite User Guide* for information on the validation feature).

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, that is, run, 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 the Project Explorer, click the **Open Project Editor** button.
4. Check that the correct Project details are displayed on the **Project Screen** and click the **Build** → **Build & Deploy** in the upper right hand corner of the Project Screen view.

Project: [HelloWorld:org.bpmms:1.0]

Save Delete Rename Copy Build X [dropdown] [dropdown]

Project Settings: Project General Settings

Project General Settings

Project Name: HelloWorld

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

Group artifact version

Group ID: org.bpmms Example: com.myorganization.myprojects ⓘ

Artifact ID: HelloWorld Example: MyProject ⓘ

Version: 1.0 1.0.0 ⓘ

Figure 5.2. Project editor with the HelloWorld project properties.

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 **Hello World** process. You might need to click the **Refresh** button to have the deployment displayed on the list.
3. Click the **Start** (▶) button next to the Process definition entry and in the **Form** dialog box, click the **Submit** button 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; refer to 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, typically the terminal emulator in which the server was started. Then the Process instance waits on the Timer Event. You can see the current execution status in the **Options** → **Process Model**.

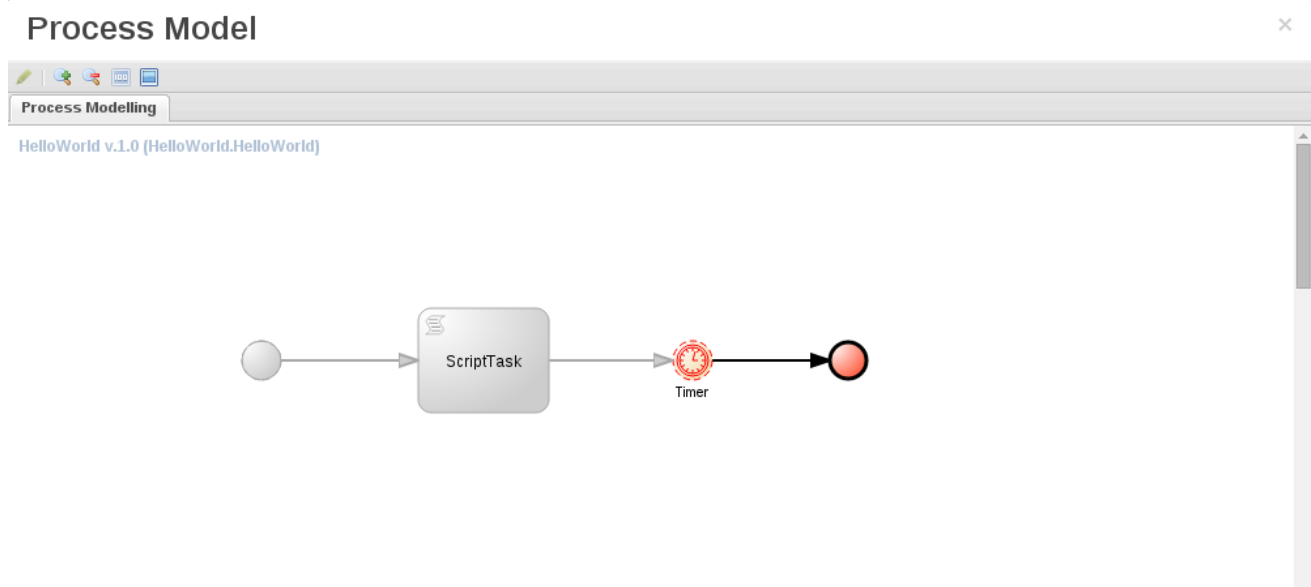


Figure 5.3. HelloWorld realtime execution diagram: the Timer Event being executed.


5.6. ABORTING BUSINESS PROCESS

At the moment, an instance of the **HelloWorld** Process is running on the Execution Server (being executed). You can check in the **Process Instances** view: to display the view, go to **Process Management** → **Process Instances**.

From here you can perform basic administration actions, such as send a Signal to a Process instance (⏏) and abort it (⏏).

Process Instances						
Bulk Actions ▾		Showing ▾ ▼ Active ▼ Completed ▼ Aborted ▼ Related To Me				⌵ ⌵
Select	Id	Name	Description	Initiator	Version	Actions
<input type="checkbox"/>	4	HelloWorld	HelloWorld	helloworlduser	1.0	⊗ ⊗

Figure 5.4. Active Process instances including the HelloWorld Process instance.

Since, the **HelloWorld** Process instance is waiting on the Timer Event, abort it: click the () on the line with the information on the **HelloWorld** Process instance: the instance will disappear from the list of active Process instances and appear in the list of **Aborted** instances.

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 JBoss BPM Suite.

You will add a new Task, a Business Rule Task, to the HelloWorld 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 refer to the *Red Hat JBoss BRMS User Guide*

6.1. CREATING THE 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 **HelloWorld** project, do the following:

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



NOTE

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 option is set to **<default>**.
 - 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. A **Save this item** prompt is displayed. Enter the **Check-in comment** and click **Save**.



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.

6.2. ADDING THE 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, click on the **BUSINESS PROCESSES** and select the **HelloWorld** process.
2. In the displayed Business Process designer with the helloworld 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:

- o Name: **BusinessRule**
- o 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.dr1** file, is present in the **helloworldgroup** group.

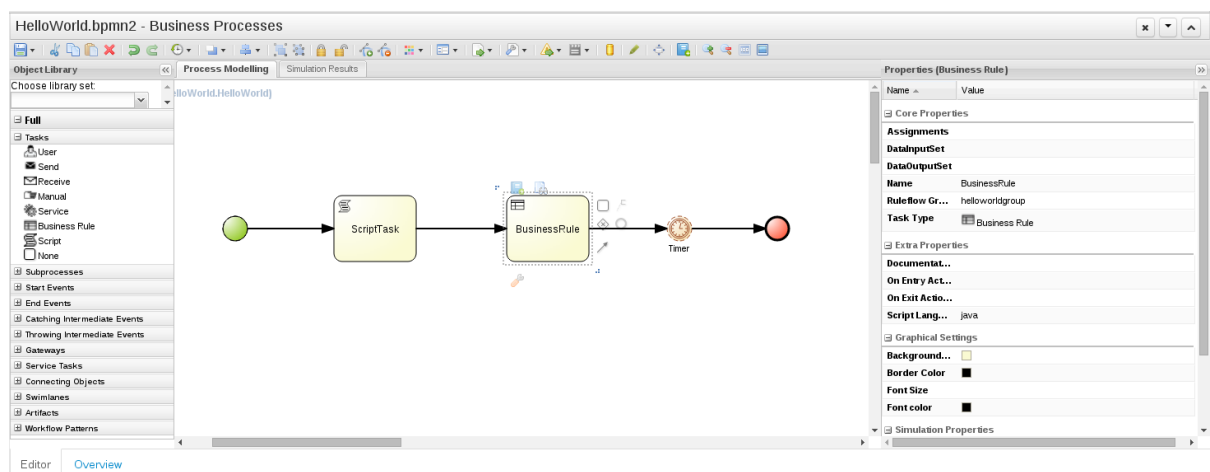


Figure 6.1. The HelloWorld process with the Business Rule Task and its properties.

5. Save the changes by clicking the **Save** button in the upper left corner.

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 HelloWorld project.
3. Open the Project in the Project editor: in the Project Explorer, click the **Open Project Editor** button.
4. Check that the correct Project details are displayed on the **Project Screen**. Change the version number of the Project and click the **Save** button.
5. Click the **Build** → **Build & Deploy** in the upper right hand corner of the Project Screen view.

Project: [HelloWorld:org.bpmms:1.0]

Save Delete Rename Copy Build

Project Settings: Project General Settings

Project General Settings

Project Name: HelloWorld

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

Group artifact version

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

Artifact ID: HelloWorld Example: MyProject

Version: 1.0

Figure 6.2. Project editor with the HelloWorld project properties.

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 that the previous version of the **HelloWorld** deployment is substituted with the new version with the Business Rule Task. By changing the version number of the Project in the Project editor, you can keep both deployments.

6.4. INSTANTIATING BUSINESS PROCESS

To create an instance of the **HelloWorld** 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** (▶) button next to the Process definition entry and in the **Form** dialog box, click the **Submit** button 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; refer to 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. Click on the **Options** →

Process Model to see the current execution status.

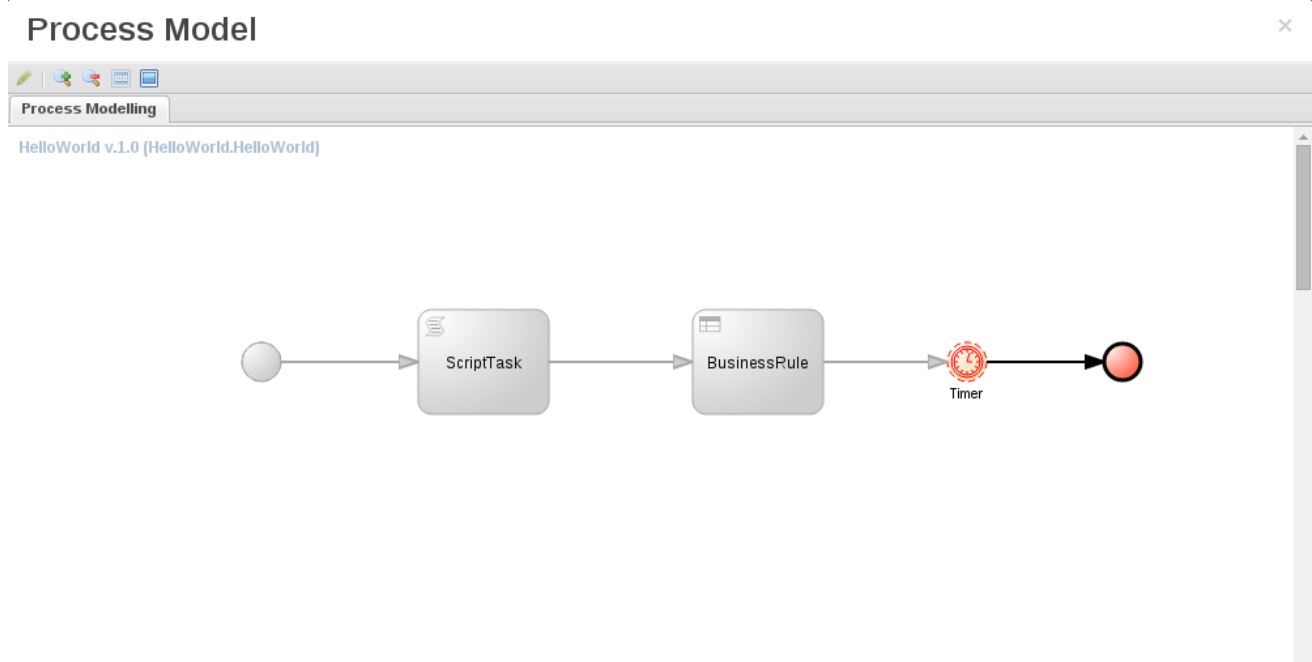


Figure 6.3. HelloWorld realtime execution diagram: the Timer Event being executed.

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**.

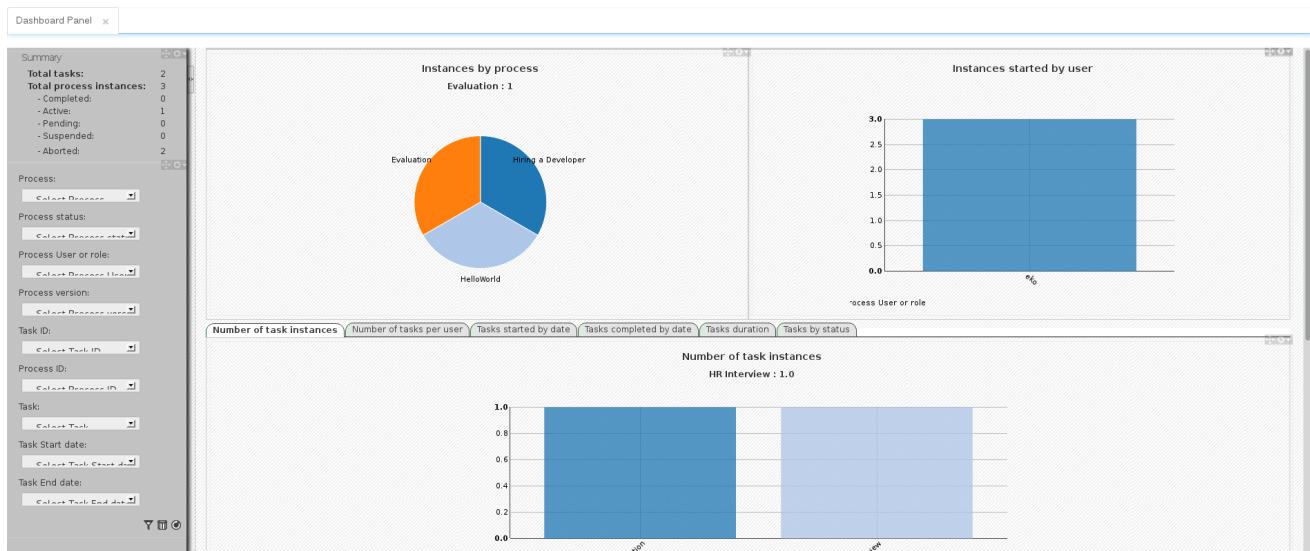


Figure 7.1. Process and Task dashboard

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

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



WARNING

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

8.1. JBOSS CENTRAL

When 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:

```
# Don't use GTK 3
export SWT_GTK3=0
```

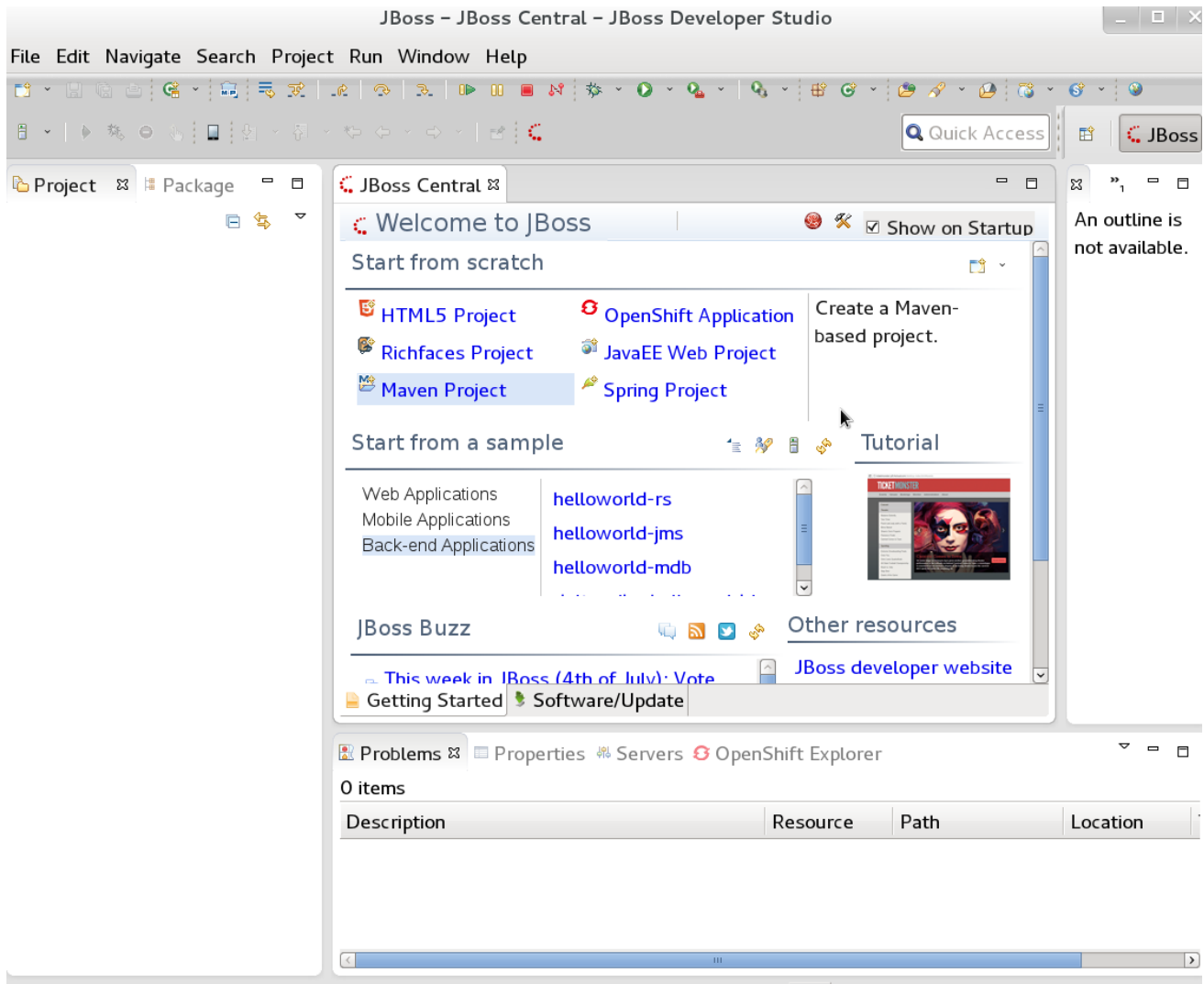


Figure 8.1. JBoss Central

8.2. INSTALLING THE JBOSS DEVELOPER STUDIO PLUG-INS

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

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

1. Start JBoss Developer Studio.
2. Select **Help** → **Install New Software**.
3. Click **Add** to enter the **Add Repository** menu.
4. Give the software site a name next to **Name** field and add the following url in the **Location** field:
<https://devstudio.jboss.com/updates/8.0/integration-stack/>
5. Click **OK**.
6. Select the **JBoss Business Process and Rule Development** feature from the available options and click **Next** and then **Next** again.
7. Read the license and accept it by selecting the appropriate radio button, and click **Finish**.

8. After installation of the plug-ins has completed, restart JBoss Developer Studio.

8.3. SETTING THE 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 8.2. Configure 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 JBoss Developer Studio menu, select **Window** and click **Preferences**.
3. Select **Drools** → **Installed Drools Runtimes**.
4. Click **Add . . .**; provide a name for the new runtime, and click **Browse** to navigate to the directory where you extracted the runtime files in step 1. Click **OK** to register the selected runtime in JBDS.
5. Mark the runtime you have created as the default Drools runtime by clicking on the check box next to it.
6. Click **OK**. If you have existing projects, a dialog box will indicate that you have to restart JBoss Developer Studio to update the Runtime.

8.4. SETTING THE JBOSS BPM SUITE RUNTIME

In order to use the 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 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 8.3. Configure jBPM Runtime

1. From the 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 JBoss Developer Studio to update the Runtime.

8.5. CONFIGURING THE JBOSS SERVER

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

Procedure 8.4. Configure the Server

1. Open the view by selecting **Window** → **Open Perspective** → **Other** and select and click **OK**.
2. Add the server view by selecting **Window** → **Show View** → **Other...** and select **Server** → **Servers**.
3. Open the server menu by right clicking the Servers panel and select **New** → **Server**.
4. Define the server by selecting **JBoss Enterprise Middleware** → **JBoss Enterprise Application Platform 6.4+** and clicking **Next**.
5. Set the home directory by clicking the **Browse** button. Navigate to and select the installation directory for JBoss EAP 6.4 which has 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 A GIT REPOSITORY INTO JBOSS DEVELOPER STUDIO

You can configure 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 8.5. Cloning a Remote Git Repository

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

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

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

Clone Git Repository

Source Git Repository

Enter the location of the source repository.

Location

URI:

Host:

Repository path:

Connection

Protocol: ▼

Port:

Authentication

User:

Password:

Store in Secure Store ☐

Figure 8.2. Git Repository Details

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 8.6. Importing a Local Git Repository

1. Start the Red Hat JBoss BRMS/BPM Suite server (whichever is applicable) by selecting the server from the server tab and click the start icon.
2. In JBoss Developer Studio, select **File** → **Import...** and navigate to the Git folder. Open the Git folder to select **Projects from Git** and click **Next**.
3. Select the repository source as **Existing local repository** and click **Next**.

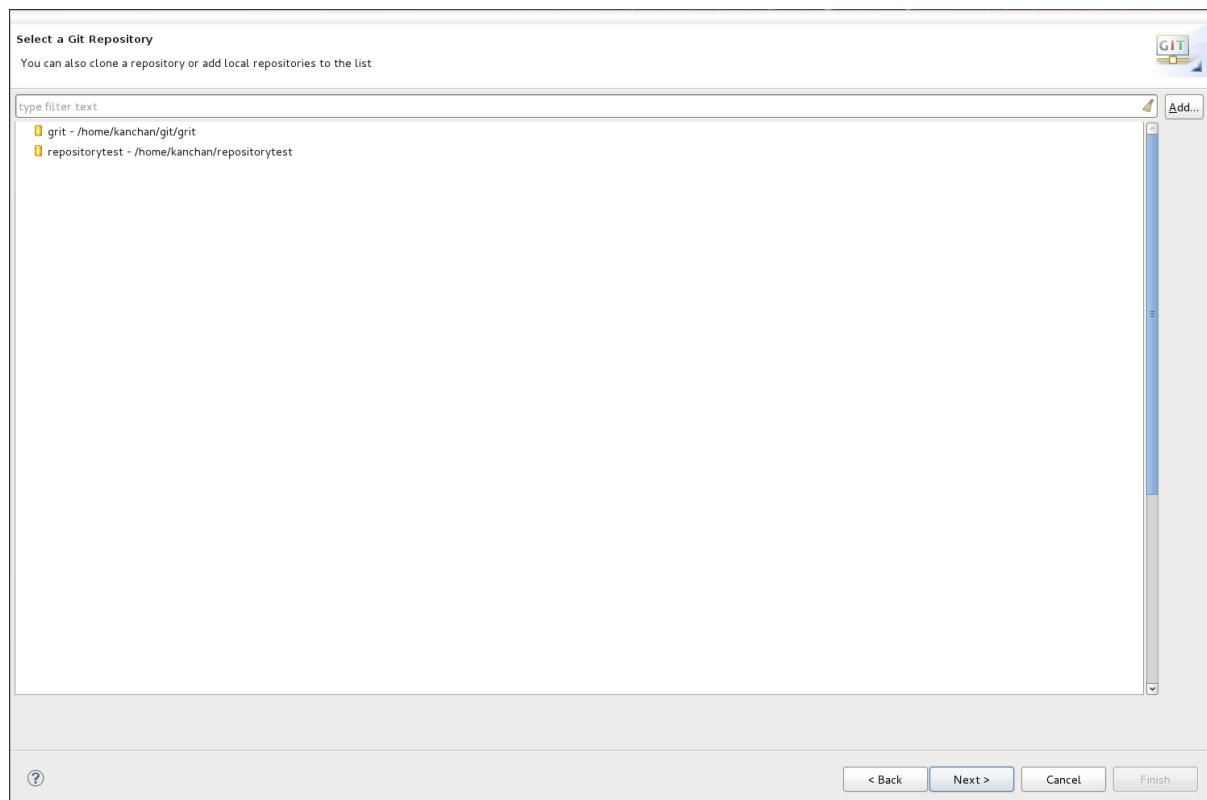


Figure 8.3. Git Repository Details

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

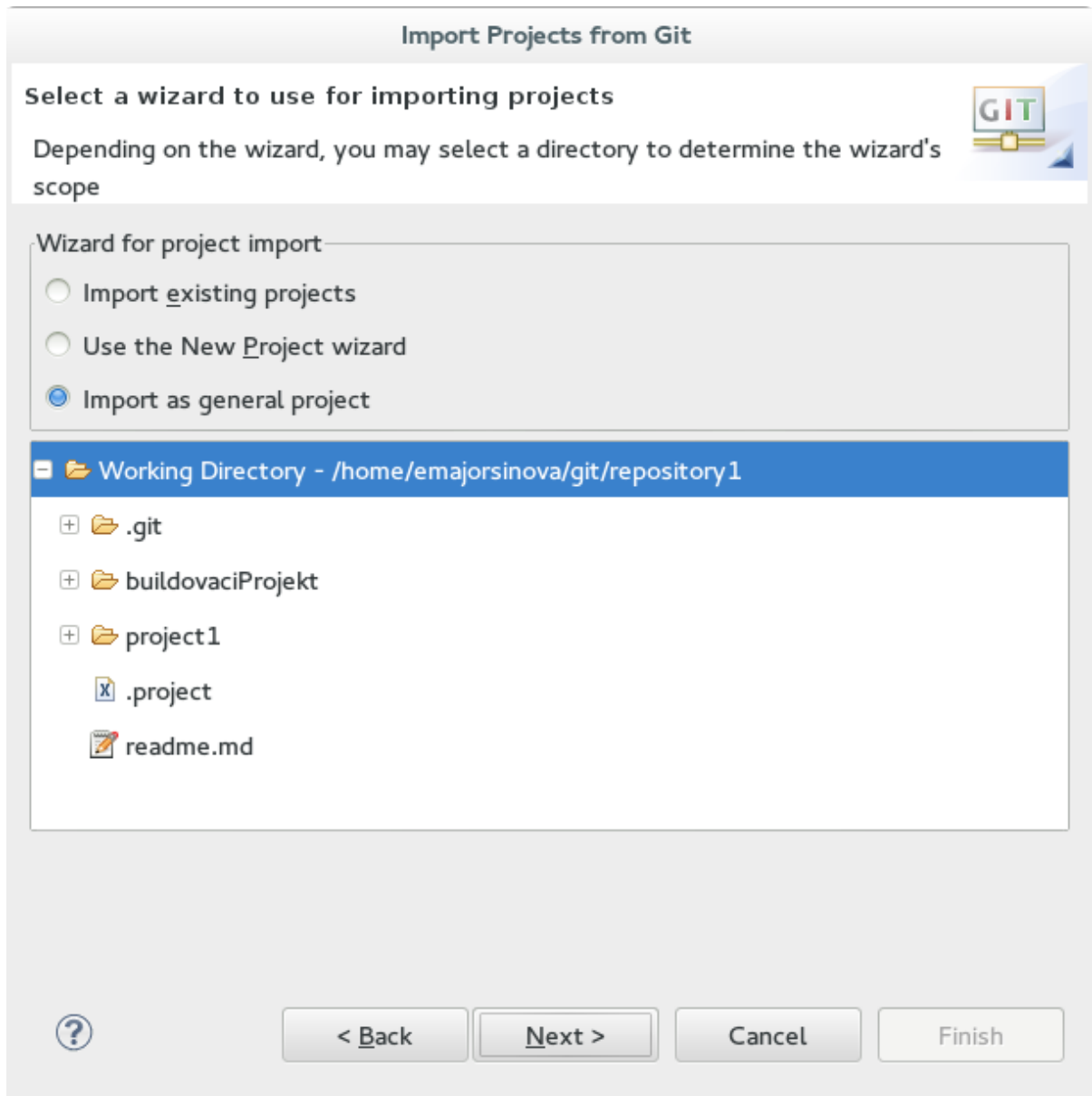


Figure 8.4. Wizard for Project Import

8.7. CREATING A DROOLS PROJECT

Procedure 8.7. Creating a New Red Hat JBoss Developer Studio Project

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

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

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



NOTE

JBoss Developer Studio provides the option to add a sample HelloWorld Rule file to the project. Accept this default by clicking **Next** to test the sample project in the following steps.

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

4. Select code compatible with **Drools 6.0.x**. Enter a **GroupID**, an **ArtifactID**, and **Version**, and click **Finish**.
5. 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 A JBOSS BPM SUITE PROJECT

Procedure 8.8. Creating a New JBoss BPM Suite project in Red Hat JBoss Developer Studio

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

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

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



NOTE

JBoss Developer Studio provides the option to add a sample HelloWorld Rule file to the project. Accept this default by clicking **Next** to test the sample project in the following steps.

3. Select the jBPM runtime (or use the default).
4. Select generate code compatible with **jBPM 6 or above**, and click **Finish**.
5. 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 TM 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 Portal](#) and log in with your user credentials.
2. Select **Downloads** → **Product Downloads**.
3. In the **Product Downloads** page that opens, click **Red Hat JBoss BPM Suite**.
4. From the **Version** drop-down menu, select version 6.1.0.
5. Select **Red Hat JBoss BPM Suite 6.1.0 Business Resource Planner** and then click **Download**.

9.2. RUNNING THE 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

Revision 1.0.0-12	Thu Dec 17 2015	Vidya Iyengar
Build includes various enhancements and fixes		