

Red Hat JBoss BPM Suite 6

Rewards Demo

(Use case: HR process - Human Task integration with BPM)

Eric D. Schabell

JBoss Technology Evangelist (Integration & BPM)

Table of Contents

Introduction.....	2
Setup and Configuration.....	2
Installation.....	2
JBoss Developer Studio Configuration.....	4
Running the Demo.....	4
JBoss BPM Suite Business Central.....	4
Administration Perspective.....	4
Project Authoring Perspective.....	5
Web Process Designer.....	5
Deploy project.....	6
Data Modeler.....	6
Form Modeler.....	7
Running the demo.....	9

Introduction

This is a JBoss BPM Suite demo application setup guide for the Rewards Demo. The setup will create a fully installed and configured JBoss BPM Suite environment so that you can demo the project in all the available BPM components.

This project demo was created to show the creation and integration of human tasks using the provided tooling into a business process.

The demo centers around a simulated company that provides a process for approving or rejecting awards for its employees. This process is called the Rewards process, anyone in the company can submit a reward for an employee by submitting the initial form when the process is started.

There is a human task to be completed by the manager role (this is the HR teams role), used solely to review the submitted reward for acceptance or rejection. If accepted, there will be a note placed in the employees personnel file (you can view it in the server log), and then an email will be sent to the employee notifying them of the results. This email has been setup to use an implementation that will publish the email to the server log instead of sending it out to an email server.

Setup and Configuration

Installation

You first need to get the project by cloning it from the central location:

```
git clone git://github.com/eschabell/bpms-rewards-demo.git
```

Once downloaded, you will have the following folder structure:

- \bpms-rewards-demo

- \installs – Initially empty, but will contain the JBoss BPM Suite platform downloads.
- \projects – Holds any developer based projects for this demo, can be empty.
- \support – Additional supporting files used by the demo.
- \target – Will be created by running `init.sh`. Contains the fully configured BRMS runtime server.
- \docs – Holds all documentation, presentations, and images to get you started with this demo project.
- `init.sh` – Script to install and configure the run time server environment.

Next, download JBoss BPM Suite project from the Red Hat Customer Portal (<https://access.redhat.com/jbossnetwork>).

Download BRMS Platform:

1. Under *JBoss Enterprise Platforms*, select the *BPM Suite* product.
2. Select version *6.0.2.GA* in the *Version* field.
3. Download *JBoss BPM Suite v6 Deployable for EAP*
(Please note that this is the deployable distribution, not the standalone one.)

Now copy `jboss-bpms-6.0.2.GA-redhat-5-deployable-eap6.zip`, to the `bpms-rewards-demo`'s *installs* folder. Ensure that this file is executable by running:

```
$ chmod +x <path-to-project>/installs/jboss-bpms-6.0.2.GA-redhat-5-deployable-eap6.zip
```

Download EAP 6 Platform:

4. Under *JBoss Enterprise Platforms*, select the *Application Platform* product.
5. Select version *6.1.1* in the *Version* field.
6. Download *JBoss Application Platform 6.1.1*.

Now copy `jboss-eap-6.1.1.zip`, to the `brms-rewards-demo`'s *installs* folder. Ensure that this file is executable by running:

```
$ chmod +x <path-to-project>/installs/jboss-eap-6.1.1.zip
```

Lastly, from the `brms-rewards-demo` folder, run the *init.sh* script:

```
$ ./init.sh
```

When the script completes you will have a new folder named *jboss-eap-6.1*, in the `brms-rewards-demo`'s *target* folder. The folder is a ready to run EAP server with BPM Suite with the following modifications made:

- The *erics*, *mary*, *alan*, *bob* accounts enabled (passwords are same all *bpmsuite*) in the *application-roles.properties* and *application-users.properties* file in *bpms-rewards-demo/target/jboss-eap-6.1/standalone/configuration*
- Adjusted configuration is copied by a new *standalone.xml* to *bpms-rewards-demo/target/jboss-eap-6.1/standalone/configuration*
- The demo project is pre-installed into the *bpms-rewards-demo/target/jboss-eap-6.1/bin/.niogit/**

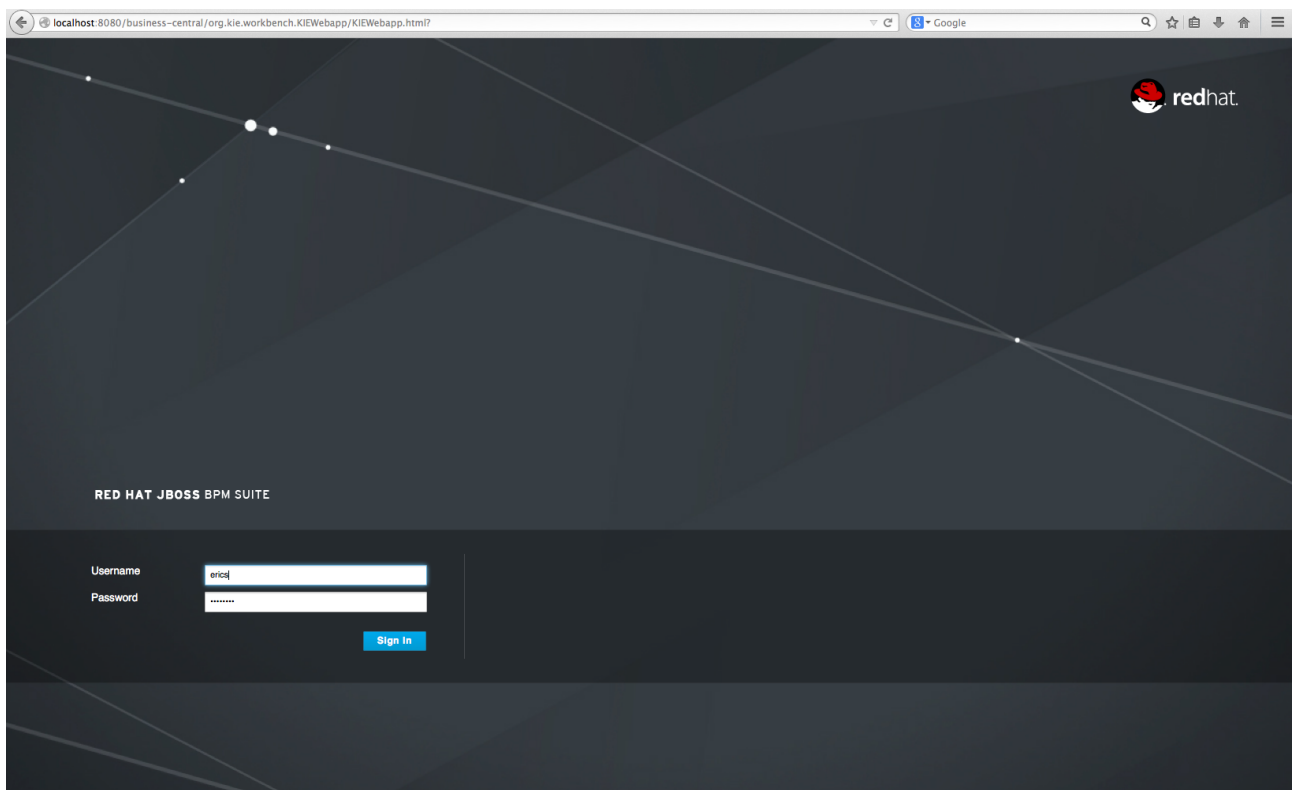
JBoss Developer Studio Configuration

Not used in this demo project (yet).

Running the Demo

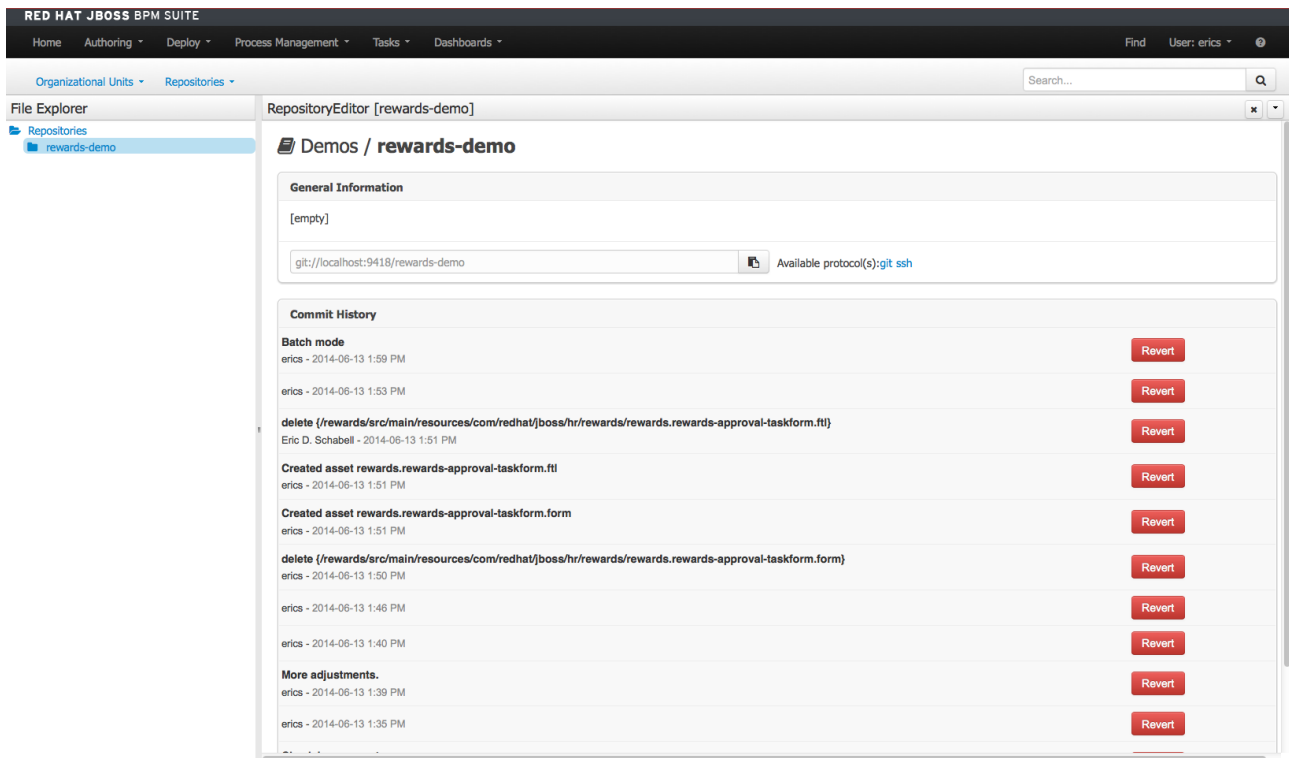
JBoss BPM Suite Business Central

To run this demo project in the Business Central dashboard you will need to install the demo and start the JBoss BPM Suite as instructed above. Once the product has been started you can login to the business-central dashboard at <http://localhost:8080/business-central>, using the admin login (u:erics/p:bpmsuite) will provide complete access to all demo components.



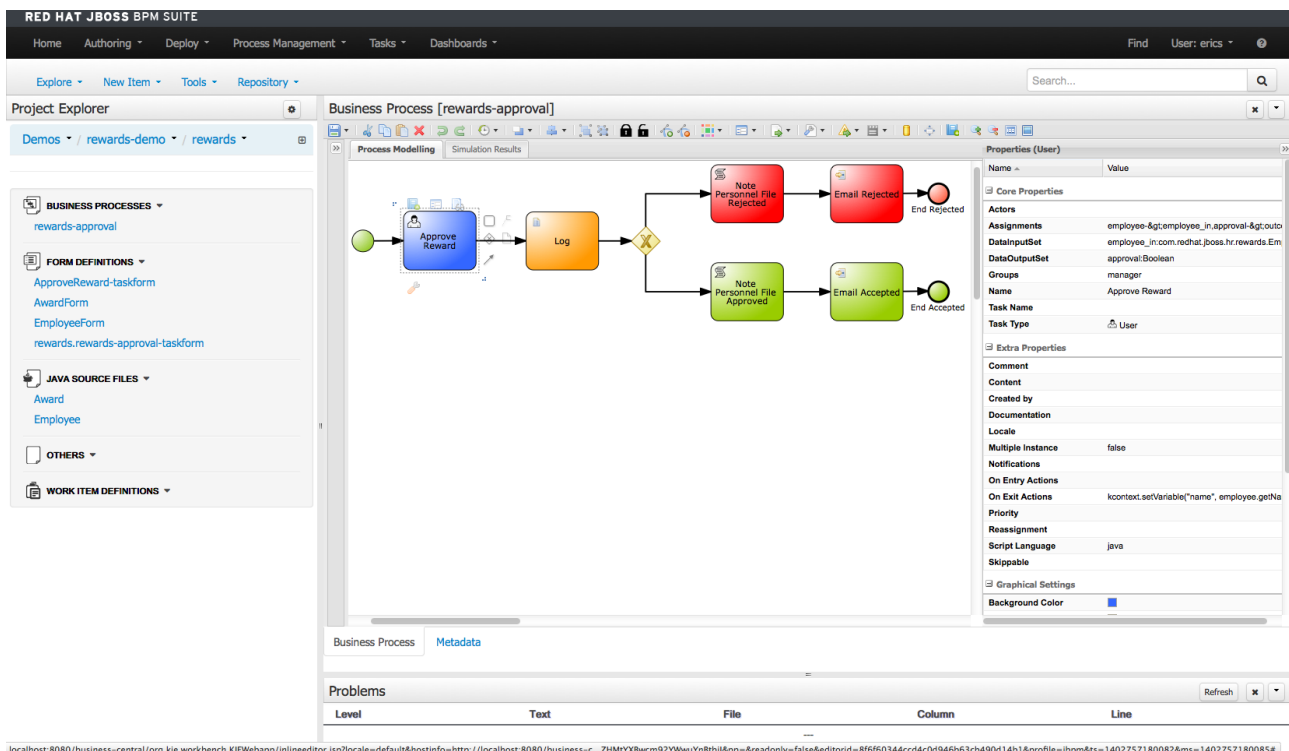
Administration Perspective

To view this perspective once you login you open by clicking on *Authoring* → *Administration*.



Project Authoring Perspective

This is the perspective that will display all the projects assets and allow you to edit or view them in each assets respective designer. This view is available by clicking *Authoring* → *Project Authoring*.



Web Process Designer

The web process designer is used to create BPMN2 processes and to review existing process assets.

When you initially open this demo you should be presented with the process designer open.

1. View process: *Authoring* → *Project Authoring* → *Project Explorer (right panel)* → *Business Processes* → *rewards-approval* and wait for web designer to appear.

Deploy project

To build the project for deployment:

1. Build project: *Authoring* → *Project Authoring* → *Tools* → *Project Editor* → *Build & Deploy (button)*, the project will be deployed and can be found under *Process Management*.

The screenshot shows the Red Hat JBoss BPM Suite interface. The top navigation bar includes 'Home', 'Authoring', 'Deploy', 'Process Management', 'Tools', and 'Dashboards'. The 'Project Explorer' on the left shows a tree structure with 'Business Processes' (rewards-approval), 'Form Definitions' (ApproveReward-taskform, AwardForm, EmployeeForm, rewards.rewards-approval-taskform), 'Java Source Files' (Award, Employee), and 'Others'. The main area displays the 'Project General Settings' for the project '[rewards:com.redhat.jboss.hr:1.0]'. The settings include: Project Name (rewards), Project Description (a text area with placeholder text), Group artifact version (com.redhat.jboss.hr), Artifact ID (rewards), and Version ID (1.0). A 'Problems' panel is visible at the bottom.

Level	Text	File	Column	Line
-------	------	------	--------	------

Data Modeler

To view the project domain model, open the data modeler:

1. Data modeler: *Authoring* → *Project Authoring* → *Tools* → *Data Modeler*; you can now view the data model.

The screenshot shows the Red Hat JBoss BPM Suite Data Modeler interface. The left sidebar shows a tree structure with 'rewards' and a 'Create' button. The main area displays the 'Award' domain model. The 'Create new field' section includes fields for '*Id' (Insert a valid Java identifier), 'Label' (Insert a label), and '*Type' (a dropdown menu). A 'Create' button is also present. Below this, the 'Award (com.redhat.jboss.hr.rewards.Award)' section shows a table with the following data:

Position	Identifier	Label	Type	
0	amount	Award amount	Integer	x
1	explanation	Explanation	String	x

The screenshot shows the 'Data modeler' application window. On the left, a sidebar lists 'rewards' with a 'Create' button and a list of identifiers: 'Award' and 'Employee'. The main area is titled 'Employee' and contains a 'Create new field' section with input fields for '*Id', 'Label', and '*Type', and a 'Create' button. Below this is a table for 'Employee (com.redhat.jboss.hr.rewards.Employee)' with columns: Position, Identifier, Label, Type, and a delete icon. The table has three rows: Position 0 with Identifier 'name' and Label 'Name' (Type String); Position 1 with Identifier 'dept' and Label 'Department' (Type String); and Position 2 with Identifier 'award' and Label 'Award' (Type Award). On the right, a 'Field' configuration panel shows details for the 'award' field: Identifier 'award', Label 'Award', Description (empty), Type 'Award (com.redhat.jboss.h...', Equals (checkbox), and Position '2'.

Position	Identifier	Label	Type
0	name	Name	String
1	dept	Department	String
2	award	Award	Award

If you are interested in viewing the source files which are Java objects generated by this modeler, click on the *Java Source Files* → *Award* or *Java Source Files* → *Employee*.

Form Modeler

Once the process was fully designed, the forms needed for user input to start the process and allow the approval task to be completed were generated using the process designer. These were modified by using the *Form Modeler*.

To view the forms found in this project, click on them to open the form modeler:

1. Form modeler: *Authoring* → *Project Authoring* → *Form Definitions* → {*rewards.rewards-approval-taskform*|*ApprovalReward-taskform*|*AwardForm*|*EmployeeForm*}, you can now view the selected form.
 - *Rewards.rewards-approval-taskform* – the form used to submit data when starting a new process, always generated with the project name and the process id.
 - *ApprovalReward-taskform* – the form used for the *Approval Reward* user task.
 - *AwardForm* – a sub-form created to encompass the embedding of an Award object into an employee form.
 - *EmployeeForm* – a sub-form to create an employee with the award form embedded. This is submitted to the process in the *Rewards.rewards-approval-taskform*.

Form Modeler [rewards.rewards-approval-taskform]

Form data origin Add fields by origin Add fields by type

HTML label	▶
Separator	▶
Simple subform	▶
Multiple subform	▶
Short text	▶
Long text	▶
Character	▶
Single precision decimal	▶
Double precision decimal	▶
BigDecimal	▶
BigInteger	▶
Byte	▶

Employee Reward Submission Form:

- name (employee)
- dept (employee)
- award (employee)
- amount (award)
- explanation (award)

Form Modeler [ApproveReward-taskform]

Form data origin Add fields by origin Add fields by type

HTML label	▶
Separator	▶
Simple subform	▶
Multiple subform	▶
Short text	▶
Long text	▶
Character	▶
Single precision decimal	▶
Double precision decimal	▶
BigDecimal	▶
BigInteger	▶

Approve Reward Form:

- name (employee)
- dept (employee)
- award (employee)
- amount (award)
- explanation (award)
- ☐ • *approval (outcome)

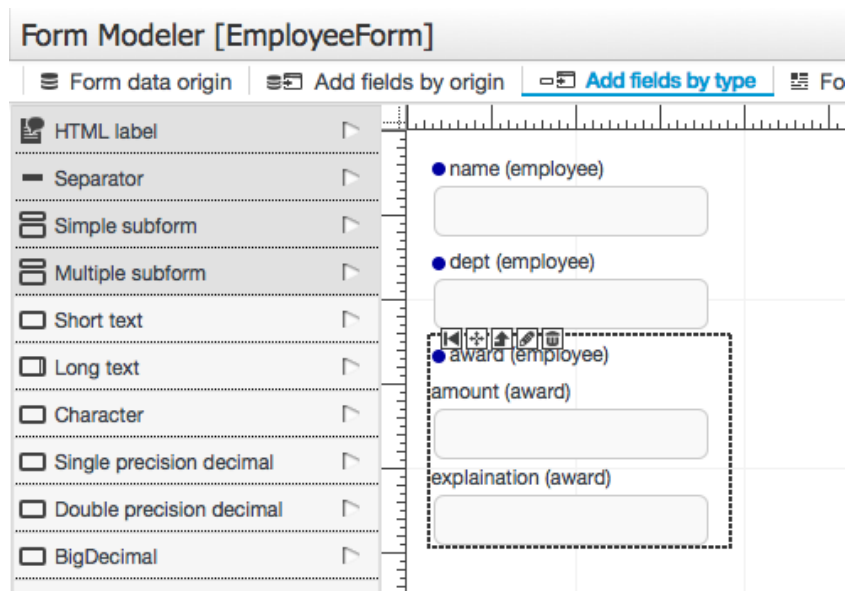
Form Modeler [AwardForm]

Form data origin Add fields by origin Add fields by type Form properties

HTML label	▶
Separator	▶
Simple subform	▶
Multiple subform	▶
Short text	▶
Long text	▶
Character	▶

Award Form:

- amount (award)
- explanation (award)



Running the demo

To run the project demo you have already read the above sections in this manual so you are logged into the JBoss BPM Suite dashboard and have viewed the project artifacts. You have also seen how to build and deploy this project.

To build the project for deployment:

1. Build project: *Authoring* → *Project Authoring* → *Tools* → *Project Editor* → *Build & Deploy (button)*, the project will be deployed and can be found under *Process Management*.
2. Start an instance of the process: *Process Management* → *Process Definitions* where you should now see the *rewards-approval* process definition deployed.
3. Click on the *magnifying glass icon* on the right to view the details of the process definition.
4. You can start a new process instance by clicking on the *New Instance* button.
5. The initial form will pop up and ask for information about the rewards submission, submit for example a name (string), a department that the employee works for (string), an amount for the reward (integer), and an explanation for the reward (string).
6. Submit the form to start the process.
7. You will be provided a look at the *Process Instance* window showing the process has started and is waiting in the *Approve Reward*.
8. To view the task, claim, and complete it you will need to open the *Tasks* → *Tasks List* view at the top menu bar.
9. You will see the entry *Approve Reward* with an unlocked lock icon and magnifying glass icon on the right. If not, make sure you are logged in with a user that has the *manager* role, which you can check by opening the *User* menu in the top right of the dashboard.
10. Claim the task by clicking on the open unlocked lock icon which will open up the *Details* of the task and add a closed lock icon, a check-mark icon, and a magnifying glass icon are now available in the *Actions* section of the task.
11. You can start working on the task by clicking on the check box icon, which opens the *Approve Reward* task form, you can view but not edit the data (read-only) in this form, but are responsible to approve (check the box) or reject (leave the check box empty) the award

and *Complete* the task.

12. The *Tasks List* you are viewing will not be empty as the task has completed and been removed from the view.
13. Go back to *Process Management* → *Process Instances*. Note that the *Active* view will no longer contain the process instance we were working with as it has completed.
14. Select *Completed* to find our completed *rewards-approval* process, where we can view the details by clicking on the magnifying glass icon.
15. The *Instance Details* show the path taken by this instance and other details like process variables and process model can be found under *Views* → *Process Variables* and *Views* → *Process Model*.
16. Finally, you can view the Business Activity Monitoring (BAM) reports on our process and task execution by opening *Dashboards* → *Process & Task Dashboard*. Feel free to explore the provided data as you see fit.

Workshop – building the demo

If you are interested in building the demo yourself, feel free to follow the workshop provided in the docs directory of this project. It is a web based workshop you can follow via your browser, just open the bpms-rewards-demo/docs/bpmsuite-rewards-workshop/index.html file.

Red Hat JBoss BPM Suite

Online Workshop

This workshop will teach you how to build a business process for a company that wants to automate its human resource (HR) employee awards program.