Table of Contents

1 Introduction

This project started with a wish to be able to run BPM processes with a load and gather metrics regarding to there performance. This initial project has been delivered to run tests on a few simple process types. There is still a lot of work to be done, but for now you can play a bit with the performance of the provided processes.

Please feel free to extend this framework to your needs and don't forget to push your changes back to this project!

2 Setup and Configuration

2.1 Installation

Download Enterprise Application Platform and BRMS Platform from the Red Hat Customer Portal (https://access.redhat.com/jbossnetwork).

Download Enterprise Application Platform:

1. Under *JBoss Enterprise Platforms*, select the *Application Platform* product.
2. Select version *6.0.1* in the *Version* field.
3. Download *Application Platform 6.0.1* (*jboss-eap-6.0.1.zip*).

Download BRMS Platform:

1. Under *JBoss Enterprise Platforms*, select the *BRMS Platform* product.
2. Select version *5.3.1* in the *Version* field.
3. Download *JBoss BRMS 5.3.1 Deployable for EAP 6* (*brms-p-5.3.1.GA-deployable-ee6.zip*).

Now copy both files to the *jboss-brms-performance-testing-framework/installs* folder. Ensure that these files are executable by running:

```
$ chmod +x <path-to-project>/installs/jboss-eap-6.0.1.zip
$ chmod +x <path-to-project>/installs/brms-p-5.3.1.GA-deployable-ee6.zip
```

Lastly, from the *jboss-brms-performance-testing-framework* folder, run the *init.sh* script:

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

When the script completes you will have a new folder named jboss-eap-6.0 in the *jboss-brms-performance-testing-framework/target* folder. The jboss-eap-6.0 folder is a ready to run Enterprise Application Platform 6.0.1 server runtime with the needed BRMS 5.3.1. parts deployed into it and the following modifications made:

- An *admin* account is enabled (password is *admin*) for the *brms* security domain; this domain is used in both the BRMS interface (Guvnor) and the Business Central interface (jBPM console).
- The data source required for jBPM 5 is added.

Furthermore the performance testing framework is installed and dependencies for the two example processes are added.

2.2 JBoss Developer Studio 5 Configuration

In this section, you will configure JBoss Developer Studio. Specifically you will add the Enterprise Application Platform 6.0.1 server runtime environment.

Important: It is assumed that you already have JBoss Developer Studio installed. The framework has been tested with JBoss Developer Studio 5.0.0.GA.

Launch JBoss Developer Studio:

1. Either select or switch to a new workspace by pointing to the *jboss-brms-performance-testing-framework* folder.
2. If the *Welcome to JBoss Developer Studio* screen appears, dismiss it by click the Workbench

arrow in the upper right.

Install SOA tools for BRMS:

1. Open the JBoss Central view (*Help → JBoss Central*), open the *Software/Update* tab, check the *Business Rules Tooling* box and hit the Install button at the bottom.
2. Follow the installation dialogue by selecting all features, accepting licenses and approving of warnings. Finally restart JBoss Developer Studio to complete the installation.

Detect and add a EAP 6.0.1 runtime environment:

1. Select *Preferences* from the *Window* menu.
2. In the left hand side, expand *JBoss Tools* and then select *JBoss Tool Runtime Detection*
3. Select the Add button and navigate to the *jboss-brms-performance-testing-framework/target* folder, then select *OK*.
4. The EAP 6.0 runtime created earlier should have been found and selected. If so, select *OK*.
5. Select *OK* again to close the Preferences dialog window.

2.3 Configure and start the JBoss Enterprise Application Platform

In this section, you will start the server from within JBoss Developer Studio, but before that you will make sure that the server will get enough memory for the additional BRMS functionality.

1. Select the *Servers* view
   If it is currently not open, select *Show View --> Other...* from the *Window* menu and search for the *Servers* view.
2. You should see the *jboss-eap-6.0* server you created in section 2.2.
3. Double click on *jboss-eap-6.0* to open the server's overview page. Under *General Information*, click the link *Open launch configuration*.
4. On the *Arguments* tab, change the VM arguments from
   '*-Xms256m -Xmx768m -XX:MaxPermSize=256m*'
   to
   '*-Xms1303m -Xmx1303m -XX:MaxPermSize=512m*'
   (leave the rest of the parameters as they are), and close the pop-up with the *OK* button. You can also close the overview page now.
5. Right click on *jboss-eap-6.0* and select *Start* from the pop-up menu. In a few seconds your JBoss Enterprise Application Platform with BRMS will be running.

2.4 Importing Rule Repository

In this section, you will import the BRMS artifacts related to the example processes into the Business Rules Manager (a.k.a. Guvnor).

1. Open up your Web browser of choice and navigate to *http://localhost:8080/jboss-brms/*.
2. Use the default credentials of *admin*/*admin*.
3. Upon logging in, you will see the following prompt**:**

   This looks like a brand new repository.
   Would you like to install a sample repository?

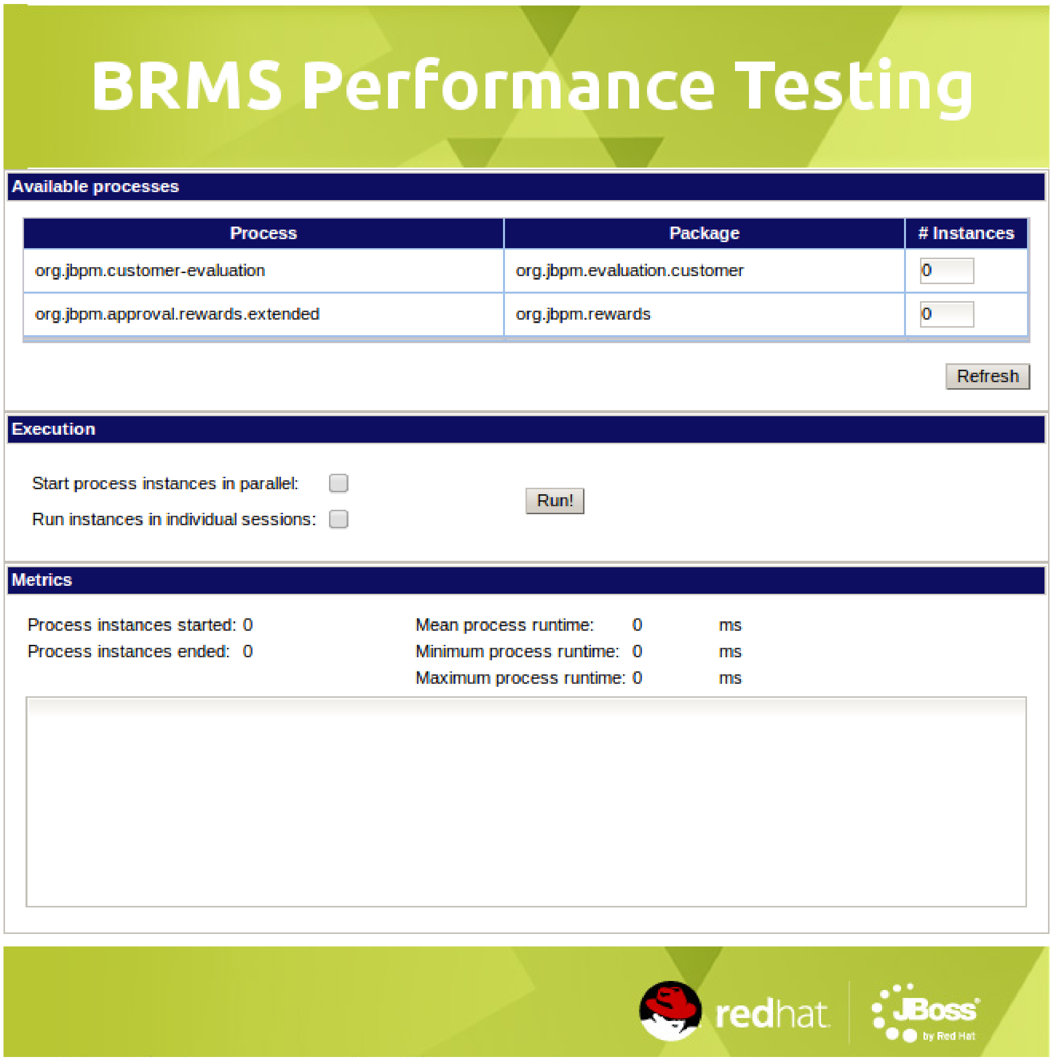
   Important: Please be sure to select ***No thanks.***
4. Select the *Administration* section on the left hand side.
5. From the *Administration* list select *Import Export*. This will open the *Import Export* window.
6. Now select *Browse...* (or *Choose File*) and navigate to *jboss-brms-performance-testing-*

framework/support folder and select the ***repository_export_CustomerEval_RewardsExtended.zip*** file.

7. Lastly, select the *Import* button. Select *OK* to confirm that you want to import the artifacts.

3 Running tests in the framework

Go to *http://localhost:8080/brms-testing*. You will see the following page:



BRMS Performance Testing

Available processes

Process	Package	# Instances
org.jbpm.customer-evaluation	org.jbpm.evaluation.customer	0
org.jbpm.approval.rewards.extended	org.jbpm.rewards	0

Execution

Start process instances in parallel: □

Run instances in individual sessions: □

Metrics

Process instances started: 0
Process instances ended: 0

Mean process runtime: 0 ms
Minimum process runtime: 0 ms
Maximum process runtime: 0 ms

3.1 Available processes

In the top panel, the processes available from Guvnor are listed. The zipped repository that was imported in 2.4 contains two example processes:

- The process from the customer evaluation demo, which contains a rule node;
- The (extended) process from the rewards demo, which contains a Human Task node.

The process ID and the name of the containing package are shown, and the last column contains an editable number that is 0 when the page is first loaded. This is the number of instances that will be started for each specific process in the next test run.

Note: When the page is already loaded and new processes become available in the Guvnor repository, those are not included automatically. Press the **Refresh**-button to re-load all available processes at any time – but beware that this will also rest the number of instances to 0 for each process entry.

3.2 Execution

The middle panel contains a couple of options for the test run:

- "***Start process instances in parallel***": when this option is chosen, instances are started as simultaneously *as possible*.
  This does not mean they all run at the same time, the configuration of the application server restricts the number of threads running at a time. The instances are run through stateless session beans (EJBs), so the configured pool size also restricts the effective parallelism.
  When this option is not selected (the default), instances are started consecutively, each starting after the previous one had ended.
- "***Run instances in individual sessions***": when this box is checked all instances will be run in their own, freshly created stateful knowledge session.
  When this option is not selected (the default), all instances are started in the same single knowledge session.

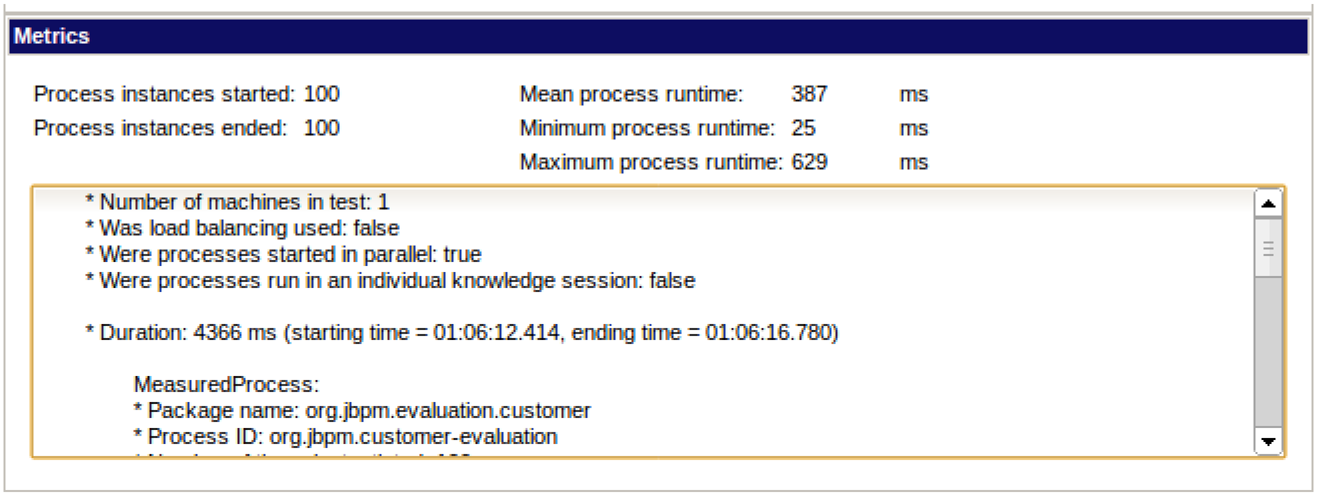
The **Run!**-button can be clicked to start a test run, that will start the indicated number of instances for each process, under the selected circumstances.

3.2.1 Metrics

The framework keeps track of all process instances it runs, and shows that info in the lower panel.

During the test run, an animation will be visible next to the **Run!**-button, indicating the synchronous part of the request to the back end is still ongoing. Once this animation is gone, any asynchronously started instances may still be running, so the panel indicates how many instances are started and ended. As long as these numbers increase and are not equal, the test run is still not finished.

Once finished, the statistical metrics (mean, minimum and maximum runtimes) are shown, as well as a detailed output for each process, instance, rule node and/or Human Task node:



Metrics

```
Process instances started: 100          Mean process runtime:     387     ms
Process instances ended:   100          Minimum process runtime:  25      ms
                                        Maximum process runtime:  629     ms
```

```
    * Number of machines in test: 1
    * Was load balancing used: false
    * Were processes started in parallel: true
    * Were processes run in an individual knowledge session: false

    * Duration: 4366 ms (starting time = 01:06:12.414, ending time = 01:06:16.780)

        MeasuredProcess:
        * Package name: org.jbpm.evaluation.customer
        * Process ID: org.jbpm.customer-evaluation
```

4 Adding processes to the framework

4.1 Add the process to Guvnor

As indicated above, processes available in the Guvnor repository will be available to the test framework as well.

On how to create processes and how to add them to the repository, please refer to the BRMS product documentation.

4.2 Provide the dependencies to the framework

4.2.1 Process model classes

Any classes used for the internal model of a process (e.g. data containers) should be made available to the framework by including them in a jar file that is put into the lib directory of the exploded testing framework ear: *jboss-brms-performance-testing-framework/target/jboss-eap-6.0/standalone/deployments/brms-testing.ear/lib.*

For the example processes (specifically the customer evaluation process) some model classes are included in the **brms-testing-examples.jar** file that is copied there by the **init.sh** script.

4.2.2 Data

Some processes can run without any external data: no input is required and no intermediate data is needed either. Those can be added to a test run without further due; however such (practical) processes are rare.

Most processes do require some form of data, in order to make decisions, have input for external services, etc., in order to run through an entire meaningful scenario. For these processes the framework offers a mechanism to provide this data programmatically.

For each process, one data provider class can be created, by implementing the ProcessInstanceDataProvider interface (or extending the AbstractDataProvider class, which provides default implementations of that interface). Any implementation should be put into the org.jboss.brms.test.service.data package, and made available in a jar file in the same lib directory as mentioned in the previous section.

In the brms-testing-examples project two such implementations are provided:

- CustomerEvaluationData: This class provides the data that is required to start the customer evaluation process in such a way that the happy flow scenario is taken. Besides the mandatory process ID, only the process start parameters are provided here.
- RewardsData: This implementation has no start input for the rewards process, but provides the work item handlers for a couple of node types *and* performs the interaction with the TaskService to deal with the Human Task node.

Both implementations are also included in the **brms-testing-examples.jar** file that is copied there by the **init.sh** script.