JBoss Cache Core Edition Tutorial

Manik Surtani

Galder Zamarreño

Release 3.0.0 Naga

October 2008

1. Introduction

JBoss Cache is an in-memory replicated, transactional, and fine-grained cache. This tutorial focuses on the core Cache API. Please refer to the accompanying tutorial for POJO Cache, if it is the POJO Cache API you are interested in.

For details of configuration, usage and APIs, please refer to the *user manuals* [http://labs.jboss.org/jbosscache/docs/index.html].

2. What You Will Learn

- Cache creation and modification
- Replication of state
- Transactions

3. Configuration

First download the JBoss Cache 3.x distribution from *the download page* [http://labs.jboss.org/jbosscache/download/index.html]. You will need the ALL distribution (`jbosscache-core-3.X.Y.GA-all.zip`). Unzip it, and you will get a directory containing the distribution, such as `jbosscache-core-3.X.Y` . For the sake of this tutorial, I will refer to this as `${JBOSSCACHE_HOME}` .

The configuration files are located in `${JBOSSCACHE_HOME}/etc`. You can modify the behavior of the cache by editing the various configuration files.

- `log4j.xml` - Logging output. You can enable logging, specify log levels or change the name and path to the log file.
- `config-samples/total-replication.xml` - Cache configuration file used for this tutorial.

4. Script

The only script needed for this tutorial is the `${JBOSSCACHE_HOME}/tutorial/build.xml` ant script. You also need to have *Apache Ant* [http://ant.apache.org/] installed for running the tutorial GUI.

5. Running The Tutorial GUI

The GUI is run by:

- Changing to the `${JBOSSCACHE_HOME}/tutorial` directory (e.g.,`cd ${JBOSSCACHE_HOME}/tutorial`)
- And then running the ant script (e.g.,`ant run`)

This will cause a GUI window to appear, giving you a tree view of the cache in the top pane and a BeanShell view of the JVM in the lower pane.

The BeanShell view is preset with the following variables:

- `cache` - a reference to the `org.jboss.cache.Cache` interface, used by the GUI instance.
- `root` - a reference to the root `org.jboss.cache.Node` instance for the above cache.
- `transactionManager` - a reference to the registered `javax.transaction.TransactionManager` instance.

The references made available to the BeanShell window point to the same cache instance used by the tree view in the GUI above.

To run the GUI as a replicated tutorial, it is useful to start another command line window and run the ant script again as you did above. Now you will have two cache instances running in two separate GUIs, replicating state to each other.

6. Tutorials

Note that it is recommended that you shut down and restart the GUI for each of the following tutorials, to ensure clean caches every time.

6.1. Caches and Nodes

For this tutorial, start a single instance of the GUI. In this tutorial, we will:

- Create nodes under the root node.
- Remove nodes under the root node, both individually and recursively.
- Add and remove data from nodes.

1. Set up the Fqns you need. In the BeanShell pane, create 3 Fqn variables:

```
childFqn1 = Fqn.fromString("/child1");
childFqn2 = Fqn.fromString("/child2");
childFqn3 = Fqn.fromString("/child2/child3");
```

2. Create child nodes under the root node.

```
child1 = root.addChild(childFqn1);
child2 = root.addChild(childFqn2);
child3 = root.addChild(childFqn3);
```

3. Query the nodes.

```
root.hasChild(childFqn1); // should return true
child2.hasChild(childFqn3.getLastElement()); // should return true
child3.getParent(); // should return child2
child2.getParent(); // should return root
```

4. Put some data in the nodes. By selecting the nodes in the tree view, you should see the contents of each node.

```
child1.put("key1", "value1");
child1.put("key2", "value2");
```

```
child2.put("key3", "value3");
child2.put("key4", "value4");
child3.put("key5", "value5");
child3.put("key6", "value6");
```

5. Query some of the data.

```
child1.getKeys();
child2.getData();
```

6. Remove some data in the nodes.

```
child1.remove("key1");
child2.remove("key3");
child3.clearData();
```

7. Delete nodes

```
root.removeChild(childFqn1); // will also remove any data held under child1
root.removeChild(childFqn2); // will recursively remove child3 as well.
```

In addition to the above, you should refer to the `Cache` and `Node` *API docs* [http://labs.jboss.org/portal/jbosscache/docs/index.html] and try out the APIs in the BeanShell script.

6.2. Replication

For this tutorial, start two instances instance of the GUI. Repeat the exercises in the previous tutorial, only alternating between the two GUI windows when creating/removing nodes or adding/removing data. This demonstrates how the two cache instances in the two GUIs are kept in sync.

6.3. Transactions

For this tutorial, start two instances instance of the GUI. Repeat the exercises in the previous tutorial, only starting transactions before creating/removing nodes or adding/removing data.

This will depict how replication only occurs on transaction boundaries. Try rolling back a few transactions as well, to see how nothing gets replicated in these cases. Below is the sample code for managing transactions:

```
tm = cache.getTransactionManager();
tm.begin();
// do operations here
tm.commit(); // or tm.rollback();
```