

# **JBoss Enterprise Application Platform 5 JBoss Cache Tutorial**

**for Use with JBoss Enterprise Application Platform 5**



**Manik Surtani, Galder Zamarreño**

# **JBoss Enterprise Application Platform 5 JBoss Cache Tutorial for Use with JBoss Enterprise Application Platform 5 Edition 5.1.1**

Author Manik Surtani, Galder Zamarreño

Copyright © 2010 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at <http://creativecommons.org/licenses/by-sa/3.0/>. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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, MetaMatrix, 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.

All other trademarks are the property of their respective owners.

This book is a TreeCache and JBossCache Tutorial for use with JBoss Enterprise Application Platform 5 and its patch releases.

---

<b>1. Introduction</b>	<b>1</b>
1.1. What You Will Learn .....	1
<b>2. Configuration</b>	<b>3</b>
<b>3. Script</b>	<b>5</b>
<b>4. Running The Tutorial GUI</b>	<b>7</b>
<b>5. Tutorials</b>	<b>9</b>
5.1. Caches and Nodes .....	9
5.2. Replication .....	10
5.3. Transactions .....	10
<b>A. Revision History</b>	<b>11</b>



# 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 about configuration, usage and APIs, please refer to the JBoss Cache User Guide for JBoss Enterprise Application Platform 5.0, available from [http://www.redhat.com/docs/en-US/JBoss\\_Enterprise\\_Application\\_Platform/](http://www.redhat.com/docs/en-US/JBoss_Enterprise_Application_Platform/).

## 1.1. What You Will Learn

- Cache creation and modification
- Replication of state
- Transactions

---

# Configuration

First download the JBoss Cache 3.x distribution from [the download page](#)<sup>1</sup>. 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.

---

<sup>1</sup> <http://labs.jboss.org/jbosscache/download/index.html>



## 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/)<sup>1</sup> installed for running the tutorial GUI.

---

<sup>1</sup> <http://ant.apache.org/>



## 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.



# Tutorials



## Important

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

## 5.1. 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
```

## 5.2. Replication

For this tutorial, start two instances 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.

## 5.3. Transactions

For this tutorial, start two instances 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();
```

---

# Appendix A. Revision History

**Revision**                      **Mon Jul 18 2011**                      **Jared Morgan** [jmorgan@redhat.com](mailto:jmorgan@redhat.com)  
**5.1.1-100**

Incorporated changes for JBoss Enterprise Application Platform 5.1.1 GA. For information about documentation changes to this guide, refer to *Release Notes 5.1.1*.

**Revision**                      **Wed Sep 15 2010**                      **Laura Bailey** [lbailey@redhat.com](mailto:lbailey@redhat.com)  
**5.1.0-100**

Changed version number in line with new versioning requirements.  
Revised for JBoss Enterprise Application Platform 5.1.0.GA.

