

HornetQ Quickstart Guide

Putting the buzz in messaging

Table of Contents

1. Legal Notice	1
2. About HornetQ	2
3. Getting Started	3
4. Download	4
4.1. Software Download	4
4.2. Project Information	4
5. Installation	5
5.1. Prerequisites	5
5.2. Standalone HornetQ Server	5
5.3. HornetQ In JBoss AS 5	6
6. Starting The Server	7
6.1. Standalone HornetQ	7
6.2. HornetQ In JBoss AS 5	7
7. Running the Examples	8
7.1. The JMS examples	8
7.2. The Java EE Examples	9

1

Legal Notice

Copyright © 2009 Red Hat, Inc. and others.

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.

2

About HornetQ

What is HornetQ?

- HornetQ is an open source project to build a multi-protocol, embeddable, high performance, clustered, asynchronous messaging system.
- HornetQ is an example of Message Oriented Middleware (MoM) For a description of MoMs and other messaging concepts please see the ???.
- For answers to more questions about what HornetQ is and isn't please visit the FAQs wiki page [<http://www.jboss.org/community/wiki/HornetQGeneralFAQs>].

Why use HornetQ? Here are just a few of the reasons:

- 100% open source software. HornetQ is licenced using the Apache Software License v 2.0 to minimise barriers to adoption.
- HornetQ is designed with usability in mind.
- Written in Java. Runs on any platform with a Java 5+ runtime, that's everything from Windows desktops to IBM mainframes.
- Superb performance. Our class beating high performance journal provides persistent messaging performance at rates normally seen for non persistent messaging, our non persistent messaging performance rocks the boat too.
- Full feature set. All the features you'd expect in any serious messaging system, and others you won't find anywhere else.
- Elegant POJO based design with minimal third party dependencies. Run HornetQ stand-alone, run it in integrated in your favourite JEE application server, or run it embedded inside your own product.
- Seamless high availability. We provide a HA solution with automatic client failover so you can guarantee zero message loss or duplication in event of server failure.
- Hugely flexible clustering. Create clusters of servers that know how to load balance messages. Link geographically distributed clusters over unreliable connections to form a global network. Configure routing of messages in a highly flexible way.
- For a full list of features, please see the features wiki page [<http://www.jboss.org/community/wiki/HornetQFeatures>].

3

Getting Started

This short guide explains how to download, install and quickly get started with HornetQ.

After downloading and installing we highly recommend you run the examples to get acquainted with HornetQ. We ship with over 65 examples demonstrating most of the features.

This guide is not intended to be a replacement for the user manual. The user manual goes into much more depth, so please consult that for further information.

4

Download

The official HornetQ project page is <http://hornetq.org/>.

4.1. Software Download

The software can be download from the Download page:<http://hornetq.org/downloads.html> [<http://hornetq.org/downloads.html/>]

4.2. Project Information

- Please take a look at our project wiki [<http://www.jboss.org/community/wiki/HornetQ>]
- If you have any user questions please use our user forum [<http://www.jboss.org/index.html?module=bb&op=viewforum&f=312>]
- If you have development related questions, please use our developer forum [<http://www.jboss.org/index.html?module=bb&op=viewforum&f=313>]
- Pop in and chat to us in our IRC channel [<irc://irc.freenode.net:6667/hornetq>]
- Our project blog [<http://hornetq.blogspot.com/>]
- Follow us on twitter [<http://twitter.com/hornetq>]
- HornetQ Subversion trunk is <http://anonsvn.jboss.org/repos/hornetq/trunk>
- All release tags are availble from <http://anonsvn.jboss.org/repos/hornetq/tags>

5

Installation

This section describes how to install HornetQ.

5.1. Prerequisites

Note

HornetQ only runs on Java 5 or later. We highly recommend to use Java 6.

By default, HornetQ server is run with 1GiB of memory. If your computer has less memory, or you want to run it with more available RAM, modify the value in `bin/run.sh` accordingly.

For persistence, HornetQ uses its own fast journal, which you can configure to use `libaio` (which is the default when running on Linux) or Java NIO. In order to use the `libaio` module on Linux, it's required to install `libaio`.

If you're not running on Linux then you don't need to worry about this.

You can install `libaio` using the following steps:

Using `yum`, (e.g. on Fedora or Red Hat Enterprise Linux):

```
sudo yum install libaio
```

Using `aptitude`, (e.g. on Ubuntu or Debian system):

```
sudo apt-get install libaio
```

Apache Ant [<http://ant.apache.org/>] is required to create the profiles for JBoss AS 5 and run the examples.

5.2. Standalone HornetQ Server

After downloading the distribution, unzip it into your chosen directory. At this point it should be possible to run straight out of the box, the following describes the directory structure:

```
|-- bin
|-- config
| |-- jboss-as
| `-- stand-alone
-- docs
| |-- api
| |-- quickstart-guide
| `-- user-manual
```

```
| -- examples  
| | -- core  
| | -- javaee  
| | -- jms  
| -- lib  
| -- licenses  
| -- schemas
```

- `bin` -- binaries and scripts needed to run JBoss Messaging
- `config` -- configuration files needed to configure HornetQ. This contains configurations to run HornetQ either in stand-alone or inside JBoss AS 5. Please refer to the reference guide for details on configuration.
- `docs` -- guides and javadocs for HornetQ
- `examples` -- JMS and Java EE examples. Please refer to the 'running examples' chapter for details on how to run them.
- `lib` -- jars and libraries needed to run HornetQ
- `licenses` -- licenses for HornetQ
- `schemas` -- XML Schemas used to validate HornetQ configuration files

5.3. HornetQ In JBoss AS 5

HornetQ can be deployed in JBoss AS 5 [<http://www.jboss.org/jbossas/>]. It is not shipped by default with the application server and you need to create new AS 5 profiles to run AS 5 with HornetQ.

To create AS 5 profiles:

1. Download JBoss AS 5
2. Set the environment property `JBOSS_HOME` to point to the directory where you installed JBoss AS 5
3. run `ant` in HornetQ's `config/jboss-as` directory

This will create 2 new profiles in `$JBOSS_HOME/server`:

- `default-with-hornetq` -- it corresponds to AS 5 `default` profile with HornetQ as its JMS provider. In this profile, HornetQ is *non-clustered*
- `all-with-hornetq` -- it corresponds to AS 5 `all` profile with HornetQ as its JMS provider. In this profile, HornetQ is *clustered*

You can then start JBoss AS 5 using one of these profiles:

```
$JBOSS_HOME/bin/run.sh -c default-with-hornetq
```

6

Starting The Server

6.1. Standalone HornetQ

To run a stand-alone server, open up a shell or command prompt and navigate into the `bin` directory. Then execute `./run.sh` (or `run.bat` on Windows) and you should see the following output

```
bin$ ./run.sh

15:05:54,108 INFO  @main [HornetQBootstrapServer] Starting HornetQ server
...
15:06:02,566 INFO  @main [HornetQServerImpl] HornetQ Server version
2.0.0.BETA5 (buzz-buzz, 107) started
```

HornetQ is now running. If any errors are displayed or if the server was not started properly, please refer to the troubleshooting guide for help

Both the run and the stop scripts use the config under `config/stand-alone/non-clustered` by default. The configuration can be changed by running `./run.sh ../config/stand-alone/clustered` or another config of your choosing. This is the same for the stop script and the windows bat files.

6.2. HornetQ In JBoss AS 5

To run HornetQ in JBoss AS 5, you need to create the AS 5 profiles for HornetQ first. Then run JBoss AS 5 with one these profiles. For example, to run JBoss AS 5 with a *non-clustered* HornetQ server, got to `$JBOSS_HOME/bin` directory and type:

```
bin$ ./run.sh -c default-with-hornetq

15:18:35,460 INFO  [ServerImpl] Starting JBoss (Microcontainer)...
15:18:35,462 INFO  [ServerImpl] Release ID: JBoss [The Oracle] 5.1.0.CR1 (build:
SVNTag=JBoss_5_1_0_CR1 date=200904091504)
...
15:19:30,305 INFO  [HornetQServerImpl] HornetQ Server version
2.0.0.BETA5 (buzz-buzz, 107) started
...
15:19:43,601 INFO  [ServerImpl] JBoss (Microcontainer) [5.1.0.CR1 (build:
SVNTag=JBoss_5_1_0_CR1 date=200904091504)] Started in 1m:8s:121ms
```

7

Running the Examples

In the directory `examples` there are 2 sets of examples, these are

- JMS Examples - these demonstrate functionality by sending and consuming JMS messages.
- Java EE Examples - these demonstrate application server integration, e.g. MDBs, EJBs, Servlets, etc.

All the examples are run using ant. You will need ant installed on your system and the ant `bin` directory on your path.

7.1. The JMS examples

The JMS Examples all follow the same format. Each examples is contained in its own directory which contains the following.

- `build.xml`

This is the ant build file used to run the example

- `src` directory

This contains the source code for the example

- `server0` configuration directory

This contains the configuration files needed to run the server for the example. There may be multiple configuration directories `server0`, `server1` etc for clustered examples etc.

Each example will start one or more stand-alone servers and stop them after the example has completed.

As a quick start we'll run the queue example. For all other examples refer to the main user manual.

Firstly open a Shell or a Command prompt and navigate to the `examples/jms/queue` directory.

Type the command `ant` and you should see the following output:

```
Buildfile: build.xml

run:

init:

compile:
    [echo] src.example.dir=/home/andy/projects/hornetq/trunk/examples/jms/queue/src
```

```

[javac] Compiling 5 source files to /home/andy/projects/hornetq/trunk/examples/jms/queue
/build/classes

runExample:
  [java] 10:41:04,149 INFO @main [JMSEExample] hornetq.example.runServer is true
  [java] 10:41:04,149 INFO @main [JMSEExample] starting server with config 'server0'
logServerOutput true
  [java] 10:41:04,149 INFO @main [JMSEExample] and vm args: -Xms512M,-Xmx512M,-XX:+U
seParallelGC,-XX:+AggressiveOpts,-XX:+UseFastAccessorMethods,-Djava.util.logging.config.
file=/home/andy/projects/hornetq/trunk/examples/jms/common/../../../../src/config/stand-alone/no
n-clustered/logging.properties
  [java] org.hornetq.jms.example.SpawnedJMSServer out:10:41:08,437 INFO @main [Journal
StorageManager] AIO journal selected
  [java] org.hornetq.jms.example.SpawnedJMSServer out:10:41:08,437 WARN @main [Journal
StorageManager] AIO wasn't located on this platform, it will fall back to using pure Jav
a NIO. If your platform is Linux, install LibAIO to enable the AIO journal
  [java] org.hornetq.jms.example.SpawnedJMSServer out:10:41:08,437 WARN @main [Securit
yStoreImpl] It has been detected that the cluster admin password which is used to replic
ate management operation from one node to the other has not had its password changed fro
m the installation default. Please see the HornetQ user guide for instructions o
n how to do this.
  [java] org.hornetq.jms.example.SpawnedJMSServer out:10:41:10,941 INFO @main [HornetQCo
nnectionFactory] read only is false
  [java] org.hornetq.jms.example.SpawnedJMSServer out:10:41:10,941 INFO @main [HornetQCo
nnectionFactory] read only is false
  [java] org.hornetq.jms.example.SpawnedJMSServer out:10:41:10,941 INFO @main [HornetQCo
nnectionFactory] read only is false
  [java] org.hornetq.jms.example.SpawnedJMSServer out:10:41:10,991 WARN @main [JMSServ
erManagerImpl] Binding for java:/ConnectionFactory already exists
  [java] org.hornetq.jms.example.SpawnedJMSServer out:10:41:10,991 WARN @main [JMSServ
erManagerImpl] Binding for java:/XACConnectionFactory already exists
  [java] org.hornetq.jms.example.SpawnedJMSServer out:10:41:11,241 INFO @main [Messagi
ngServerImpl] HornetQ Server version 2.0.0.BETA5 (buzz-buzz, 107) started
  [java] org.hornetq.jms.example.SpawnedJMSServer out:10:41:11,241 INFO @main [HornetQBoot
strapServer] HornetQ server started
  [java] org.hornetq.jms.example.SpawnedJMSServer out:STARTED::
  [java] 10:41:11,276 INFO @main [JMSEExample] using server0/client-jndi.properties f
or jndi
  [java] Sent message: This is a text message
  [java] Received message: This is a text message
  [java]
  [java] #####
  [java] ### SUCCESS! ###
  [java] #####

BUILD SUCCESSFUL
Total time: 13 seconds

```

Congratulations! You have successfully run your first HornetQ example. Try some of the others.

7.2. The Java EE Examples

The Java EE Examples are examples that require an Application Server to run. They include MDB, Servlet, EJB examples etc. For this you will need The JBoss Application Server installed. How to do this is explained in the previous chapters.

We'll use the MDB example for the purposes of this guide. For the other examples refer to the user guide. Before going any further ensure that the Jboss Application Server is running.

Like the other Java EE examples the MDB example is a JEE Application which first needs deploying in the JBoss

Application Server. The first thing we need to do is set the `JBOSS_HOME` environment property to the location of the JBoss Application Server, in a Linux shell this would be something like:

```
export JBOSS_HOME=/home/jbossas5.1/build/output/jboss-5.1.0.GA
```

Once set we can then deploy the application, from inside the `examples/javaee/mdb` directory run the command:

```
ant deploy
```

This will make a copy of the messaging profile in the JBoss Application server and start the server.

In the shell window you should see something like the following output:

```
Buildfile: build.xml

validate-jboss:

deploy:

deploy-resources:
  [copy] Copying 1 file to /home/andy/projects/jbossas5.1/build/output/jboss-5.1.0.GA
  /server/default-with-hornetq/deploy/hornetq.sar
  [copy] Copying 1 file to /home/andy/projects/jbossas5.1/build/output/jboss-5.1.0.GA
  /server/default-with-hornetq/deploy/hornetq.sar

init:

compile:

jar:
  [jar] Building jar: /home/andy/projects/hornetq/trunk/examples/javaee/mdb/build/mdb-exa
  mple.jar

ear:
  [jar] Building jar: /home/andy/projects/hornetq/trunk/examples/javaee/mdb/build/mdb-exa
  mple.ear

deploy-ear:
  [copy] Copying 1 file to /home/andy/projects/jbossas5.1/build/output/jboss-5.1.0.GA

  .....

[exec] 16:01:04,607 INFO [JBossASKernel]          Class:javax.jms.MessageListener
[exec] 16:01:04,607 INFO [JBossASKernel] Added bean(jboss.j2ee:ear=mdb-example.ear,
jar=mdb-example.jar,name=MessageMDBExample,service=EJB3) to KernelDeployment of:
mdb-example.jar
[exec] 16:01:04,702 INFO [EJBContainer] STARTED EJB: org.hornetq.javaee.example.server
.MDBExample ejbName: MessageMDBExample
[exec] 16:01:09,496 INFO [TomcatDeployment] deploy, ctxPath=/mdb-example
[exec] 16:01:09,812 WARNING [config] Unable to process deployment descriptor for cont
ext '/mdb-example'
[exec] 16:01:09,812 INFO [config] Initializing Mojarra (1.2_12-b01-FCS) for context
'/mdb-example'
[exec] 16:01:09,958 INFO [Http11Protocol] Starting Coyote HTTP/1.1 on http-127.0.0.1-
8080
[exec] 16:01:09,990 INFO [AjpProtocol] Starting Coyote AJP/1.3 on ajp-127.0.0.1-8009
[exec] 16:01:10,001 INFO [ServerImpl] JBoss (Microcontainer) [5.1.0.GA (build: SVNTag
=JBoss_5_1_0_GA date=200905151005)] Started in 6m:21s:797ms
```

We can now run the example, do this by running the following command from within the `mdb` directory.

```
ant
```

You should see some output similar to the following:

```
Buildfile: build.xml

run:

init:

compile:

runExample:
    [java] Sent message: This is a text message

BUILD SUCCESSFUL
Total time: 7 seconds
```

You may also see some output from the JBoss Application Server, in this instance it is:

```
16:27:54,703 INFO [STDOUT] message This is a text message received
```

The application can then be undeployed by running the following command from within the `mdb` directory.

```
ant undeploy
```

This should so some output similar to the following:

```
Buildfile: build.xml

validate-jboss:

undeploy:

undeploy-misc:
    [delete] Deleting: /home/andy/projects/jbossas5.1/build/output/jboss-5.1.0.GA
    /server/default-with-hornetq/deploy/mdb-example.ear
    [delete] Deleting: /home/andy/projects/jbossas5.1/build/output/jboss-5.1.0.GA
    /server/default-with-hornetq/deploy/hornetq.sar/hornetq-queues.xml
    [delete] Deleting: /home/andy/projects/jbossas5.1/build/output/jboss-5.1.0.GA
    /server/default-with-hornetq/deploy/hornetq.sar/hornetq-jms.xml

BUILD SUCCESSFUL
Total time: 1 second
```

You can verify that the application has been undeployed by checking the output from the JBoss Application Server, in this instance it will be something like:

```
16:36:45,277 INFO [EJBContainer] STOPPED EJB: org.jboss.javaee.example.server.MDBExample
    ejbName: MessageMDBExample
```

Congratulations! you have successfully deployed and run a Java EE example.