

JBoss Messaging 2.0 Quickstart Guide

Setting the Standard for High Performance Messaging

Table of Contents

1. About JBoss Messaging	1
2. Getting Started	2
3. Download	3
3.1. Software Download	3
3.2. Project Information	3
4. Installation	4
4.1. Prerequisites	4
4.2. Standalone JBoss Messaging Server	4
4.3. JBoss Messaging In JBoss AS 5	5
5. Starting The Server	6
5.1. Standalone JBoss Messaging	6
5.2. JBoss Messaging In JBoss AS 5	6
6. Running the Examples	7
6.1. The JMS examples	7
6.2. The Java EE Examples	8

1

About JBoss Messaging

The goal of JBoss Messaging is simple and uncompromising: to bring unrivaled levels of performance and reliability to messaging, and to be the fastest, best featured, and most scalable multi-protocol messaging system.

Why use JBoss Messaging? Here are a few of the reasons:

- 100% open source software. We use the Apache Software License version 2.0 to limit restrictions on use.
- Written in Java. Runs on any platform with a J2SE 5.0 JDK, 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.
- 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 JBoss Messaging stand-alone, run it in integrated in your favourite Java EE Application Server or IoC container, or run it embedded inside your own product.
- Seamless high availability. We provide server replication and completely transparent client failover so you don't have to worry about coding your client specially for an HA environment.
- 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.

2

Getting Started

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

After downloading and installing we highly recommend you run the examples to get acquainted with JBoss Messaging. We ship with over 50 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.

3

Download

The official JBoss Messaging project page is <http://www.jboss.org/jbossmessaging/>.

3.1. Software Download

The download location is the JBoss Labs Messaging Project download zone: <http://www.jboss.org/jbossmessaging/downloads/>

3.2. Project Information

If you have any user questions please use our user forum [<http://www.jboss.org/index.html?module=bb&op=viewforum&f=238>]

If you have development related questions, please use our development forum [<http://www.jboss.org/index.html?module=bb&op=viewforum&f=153>]

Pop in and chat to us in our IRC channel [<irc://irc.freenode.net:6667/jbossmessaging>]

JBoss Messaging Subversion TRUNK is here [<http://anonsvn.jboss.org/repos/messaging/trunk>]

All our release tags are here [<http://anonsvn.jboss.org/repos/messaging/tags>]

4

Installation

This section describes how to install JBoss Messaging.

4.1. Prerequisites

Note

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

By default, JBoss Messaging server is run with 1GB of memory. If your computer has less memory, modify the value in `bin/run.sh` accordingly.

For persistence, JBoss Messaging uses its own fast journal file, 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`.

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.

4.2. Standalone JBoss Messaging 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 JBoss Messaging. This contains configurations to run JBoss Messaging either in stand-alone or inside JBoss AS 5. Please refer to the reference guide for details on configuration.
- `docs` -- guides and javadocs for JBoss Messaging
- `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 JBoss Messaging
- `licenses` -- licenses for JBoss Messaging
- `schemas` -- XML Schemas used to validate JBoss Messaging configuration files

4.3. JBoss Messaging In JBoss AS 5

JBoss Messaging 2.0 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 JBoss Messaging.

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 JBoss Messaging's `config/jboss-as` directory

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

- `default-with-jbm2` -- it corresponds to AS 5 `default` profile with JBoss Messaging 2 as its JMS provider. In this profile, JBoss Messaging is *non-clustered*
- `all-with-jbm2` -- it corresponds to AS 5 `all` profile with JBoss Messaging 2 as its JMS provider. In this profile, JBoss Messaging is *clustered*

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

```
$JBOSS_HOME/bin/run.sh -c default-with-jbm2
```

5

Starting The Server

5.1. Standalone JBoss Messaging

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 [JBMBootstrapServer] Starting JBoss Messaging server
...
15:06:02,566 INFO  @main [MessagingServerImpl] JBoss Messaging Server version
2.0.0.BETA1-SNAPSHOT (Stilton, 101) started
```

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

By default, JBoss Messaging is run in a *non-clustered* configuration (based on the configuration in `config/stand-alone/non-clustered` directory).

5.2. JBoss Messaging In JBoss AS 5

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

```
bin$ ./run.sh -c default-with-jbm2

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  [MessagingServerImpl] JBoss Messaging Server version
2.0.0.BETA1-SNAPSHOT (Stilton, 101) 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
```

6

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.

6.1. 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/JBMTTrunk/examples/jms/queue/src
```

```

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

runExample:
 [java] 10:41:04,149 INFO @main [JMSEExample] jbm.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/JBMTrunk/examples/jms/common/../../../../src/config/stand-alone/no
n-clustered/logging.properties
 [java] org.jboss.jms.example.SpawnedJMSServer out:10:41:08,437 INFO @main [Journal
StorageManager] AIO journal selected
 [java] org.jboss.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.jboss.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 JBoss Messaging user guide for instructions o
n how to do this.
 [java] org.jboss.jms.example.SpawnedJMSServer out:10:41:10,941 INFO @main [JBossCo
nnectionFactory] read only is false
 [java] org.jboss.jms.example.SpawnedJMSServer out:10:41:10,941 INFO @main [JBossCo
nnectionFactory] read only is false
 [java] org.jboss.jms.example.SpawnedJMSServer out:10:41:10,941 INFO @main [JBossCo
nnectionFactory] read only is false
 [java] org.jboss.jms.example.SpawnedJMSServer out:10:41:10,991 WARN @main [JMSServ
erManagerImpl] Binding for java:/ConnectionFactory already exists
 [java] org.jboss.jms.example.SpawnedJMSServer out:10:41:10,991 WARN @main [JMSServ
erManagerImpl] Binding for java:/XAConnectionFactory already exists
 [java] org.jboss.jms.example.SpawnedJMSServer out:10:41:11,241 INFO @main [Messagi
ngServerImpl] JBoss Messaging Server version 2.0.0.BETA1-SNAPSHOT (Stilton, 101) started
 [java] org.jboss.jms.example.SpawnedJMSServer out:10:41:11,241 INFO @main [JBMBoot
strapServer] JBoss Messaging server started
 [java] org.jboss.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 Jboss Messaging example. Try some of the others.

6.2. 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 J2EE 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-jbm2/deploy/messaging.sar
  [copy] Copying 1 file to /home/andy/projects/jbossas5.1/build/output/jboss-5.1.0.GA
  /server/default-with-jbm2/deploy/messaging.sar

init:

compile:

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

ear:
  [jar] Building jar: /home/andy/projects/JBMTrunk/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.jboss.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-jbm2/deploy/mdb-example.ear
    [delete] Deleting: /home/andy/projects/jbossas5.1/build/output/jboss-5.1.0.GA
    /server/default-with-jbm2/deploy/messaging.sar/jbm-queues.xml
    [delete] Deleting: /home/andy/projects/jbossas5.1/build/output/jboss-5.1.0.GA
    /server/default-with-jbm2/deploy/messaging.sar/jbm-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.