

INSTALLING AMQ BROKER

DOWNLOADING THE AMQ BROKER ARCHIVE

Prerequisites

- You must have a Red Hat subscription.

Procedure

1. In a web browser, navigate to <https://access.redhat.com/downloads/> and log in. The Product Downloads page is displayed.
2. In the JBoss Integration and Automation section, click the Red Hat AMQ Broker link. The Software Downloads page is displayed.
3. Select the desired AMQ Broker version from the Version drop-down menu.
4. On the Releases tab, click the Download link for the specific AMQ Broker file you want to download.

EXTRACTING THE AMQ BROKER ARCHIVE ON LINUX

Procedure

5. Create a new user named **amq-broker** and provide it a password.

```
$ sudo useradd amq-broker  
$ sudo passwd amq-broker
```

6. Create the directory **/opt/redhat/amq-broker** and make the new **amq-broker** user and group the owners of it.

```
$ sudo mkdir /opt/redhat  
$ sudo mkdir /opt/redhat/amq-broker  
$ sudo chown -R amq-broker:amq-broker /opt/redhat/amq-broker
```

7. Change the owner of the archive to the new user.
-

```
$ sudo chown amq-broker:amq-broker amq-broker-7.x.x-bin.zip
```

8. Move the installation archive to the directory you just created.

```
$ sudo mv amq-broker-7.x.x-bin.zip /opt/redhat/amq-broker
```

9. As the new **amq-broker** user, extract the contents by using the **unzip** command.

```
$ su - amq-broker
$ cd /opt/redhat/amq-broker
$ unzip <archive_name>.zip
$ exit
```

A directory named something similar to **apache-artemis-2.33.0.redhat-00010** is created. In the documentation, this location is referred to as **<install_dir>**.

EXTRACTING THE AMQ BROKER ARCHIVE ON WINDOWS SYSTEMS

Procedure

10. Use Windows Explorer to create the directory folder **\redhat\amq-broker** on the desired drive letter.
For example: **C:\redhat\amq-broker**
11. Use Windows Explorer to move the installation archive to the directory you just created.
12. In the **\redhat\amq-broker** directory, right-click the installation archive zip file and select
Extract All.
A directory named something similar to **apache-artemis-2.33.0.redhat-00010** is created. In the documentation, this location is referred to as **<install_dir>**.

CREATING A STANDALONE BROKER

Prerequisites

- AMQ Broker must be installed.

CREATING A BROKER INSTANCE

Procedure

1. Create a directory for the broker instance.

| If you are using... | Do this... |
|--------------------------|---|
| Red Hat Enterprise Linux | <ol style="list-style-type: none">1. Create a new directory to serve as the location for the broker instance. <pre>\$ sudo mkdir /var/opt/amq-broker</pre>2. Assign the user that you created during installation. <pre>\$ sudo chown -R amq-broker:amq-broker /var/opt/amq-broker</pre> |
| Windows | Use Windows Explorer to create a new folder to serve as the location for the broker instance. |

2. Use the **artemis create** command to create the broker.

| If you are using... | Do this... |
|--------------------------|--|
| Red Hat Enterprise Linux | <ol style="list-style-type: none">1. Switch to the user account you created during installation. \$ su - amq-broker2. Change to the directory you just created for the broker instance. \$ cd /var/opt/amq-broker3. From the broker instance's directory, create the broker instance. \$ <install_dir>/bin/artemis create mybroker |
| Windows | <ol style="list-style-type: none">1. Open a command prompt from the directory you just created for the broker instance.2. From the broker instance's directory, create the broker instance. > <install_dir>\bin\artemis.cmd create mybroker |

3. Follow the **artemis create** prompts to configure the broker instance.

Example 4.1. Configuring a broker instance using **artemis create**

```
$ /opt/redhat/amq-broker/bin/artemis create mybroker
```

```
Creating ActiveMQ Artemis instance at: /var/opt/amq-broker/mybroker
```

```
--user: is mandatory with this configuration:  
Please provide the default username:  
admin
```

```
--password: is mandatory with this configuration:  
Please provide the default password:
```

```
--role: is mandatory with this configuration:  
Please provide the default role:
```

amq

```
--allow-anonymous | --require-login: is mandatory with
this configuration: Allow anonymous access? (Y/N):
Y
```

Auto tuning journal ...

done! Your system can make 19.23 writes per millisecond, your journal-buffer-timeout will be 52000

You can now start the broker by executing:

```
"/var/opt/amq-broker/mybroker/bin/artemis" run
```

Or you can run the broker **in** the

background using: `"/var/opt/amq-`

```
broker/mybroker/bin/artemis-service" start
```

STARTING THE BROKER INSTANCE

Procedure

3. Switch to the user account you created during installation.

\$ su - amq-broker

4. Use the **artemis run** command to start the broker instance.

```
$ /var/opt/amq-broker/mybroker/bin/artemis run
```

Λ | V | / \ | _ \ | |
 / \ | \ / | | | | |) | _ | | _ _ _ _
 / \ \ | M | | | | | _ < | ' _ / _ \ | / / _ \ ' _ |
 / _ _ \ | | | | _ | | | |) | | | (_ | < _ / |
 / / \ \ | | _ \ \ \ | _ / | _ \ \ / | _ \ \

Red Hat JBoss AMQ

7.2.1.GA

10:53:43,959 INFO

```
[org.apache.activemq.artemis.integration.bootstrap] AMQ101000:
Starting ActiveMQ Artemis Server
10:53:44,076 INFO [org.apache.activemq.artemis.core.server]
AMQ221000: live Message Broker is starting with configuration
Broker Configuration
(clustered=false,journalDirectory=./data/journal,bindingsDirectory=./data/b
indings,largeMessage sDirectory=./data/large-
messages,pagingDirectory=./data/paging)
10:53:44,099 INFO [org.apache.activemq.artemis.core.server]
AMQ221012: Using AIO Journal
...
```

PRODUCING AND CONSUMING TEST MESSAGES

Procedure

5. Use the **artemis producer** command to produce a few test messages and send them to the broker.

This command sends 100 messages to the **helloworld** address, which is created automatically on the broker. The producer connects to the broker by using the default port 61616, which accepts all supported messaging protocols.

```
$ /opt/redhat/amq-broker/amq-broker-7.2.0/bin/artemis producer --
destination helloworld --message-count 100 --url
tcp://localhost:61616
```

```
Producer ActiveMQQueue[helloworld], thread=0 Started to calculate
elapsed time ...
```

```
Producer ActiveMQQueue[helloworld], thread=0
```

```
Produced: 100 messages Producer
```

```
ActiveMQQueue[helloworld], thread=0 Elapsed time in
second : 1 s
```

```
Producer ActiveMQQueue[helloworld], thread=0 Elapsed time in milli
second : 1289 milli seconds
```

6. Use the web console to see the messages stored in the broker.

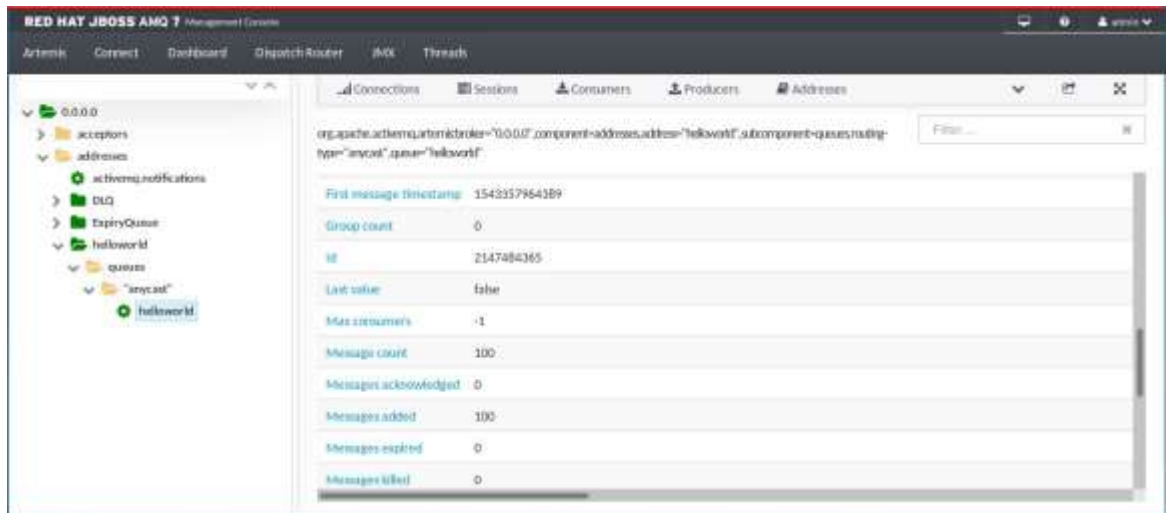
- a. In a web browser, navigate to <http://localhost:8161>.

- b. Log into the console using the default username and default password that you created when you created the broker instance.

The **Attributes** tab is displayed.

- c. On the **Attributes** tab, navigate to addresses → helloworld → queues → "anycast" → helloworld.
- In the previous step, you sent messages to the **helloworld** address. This created a new anycast **helloworld** address with a queue (also named **helloworld**). The **Message count** attribute shows that all 100 messages that were sent to **helloworld** are currently stored in this queue.

Figure 4.1. Message count



Use the **artemis consumer** command to consume 50 of the messages stored on the broker. This command consumes 50 of the messages that you sent to the broker previously.

```
$ /opt/redhat/amq-broker/amq-broker-7.2.0/bin/artemis consumer --  
destination helloworld -- message-count 50 --url  
tcp://localhost:61616
```

```
Consumer:: filter = null
```

```
Consumer ActiveMQQueue[helloworld], thread=0 wait until 50  
messages are consumed Consumer ActiveMQQueue[helloworld],  
thread=0 Consumed: 50 messages
```

```
Consumer ActiveMQQueue[helloworld], thread=0 Consumer thread  
finished
```

7. In the web console, verify that the **Message count** is now 50.
50 of the messages were consumed, which leaves 50 messages stored in the **helloworld** queue.
8. Stop the broker and verify that the 50 remaining messages are still stored in the **helloworld** queue.

- a. In the terminal in which the broker is running, press **Ctrl+C** to stop the broker.
- b. Restart the broker.

```
$ /var/opt/amq-broker/mybroker/bin/artemis run
```

- c. In the web console, navigate back to the **helloworld** queue and verify that there are still 50 messages stored in the queue.
9. Consume the remaining 50 messages.

```
$ /opt/redhat/amq-broker/amq-broker-7.2.0/bin/artemis consumer --  
destination helloworld -- message-count 50 --url  
tcp://localhost:61616
```

```
Consumer:: filter = null
```

```
Consumer ActiveMQQueue[helloworld], thread=0 wait until 50  
messages are consumed Consumer ActiveMQQueue[helloworld],  
thread=0 Consumed: 50 messages
```

```
Consumer ActiveMQQueue[helloworld], thread=0 Consumer thread  
finished
```

10. In the web console, verify that the **Message count** is 0.
All of the messages stored in the **helloworld** queue were consumed,
and the queue is now empty.

STOPPING THE BROKER INSTANCE

Procedure

- Use the **artemis stop** command to stop the broker instance:

```
$ /var/opt/amq-broker/mybroker/bin/artemis stop  
2018-12-03 14:37:30,630 INFO  
[org.apache.activemq.artemis.core.server] AMQ221002: Apache  
ActiveMQ Artemis Message Broker version 2.6.1.amq-720004-redhat-  
1 [b6c244ef-  
f1cb-11e8-a2d7-0800271b03bd] stopped, uptime  
35 minutes Server stopped!
```


RUNNING THE AMQ BROKER EXAMPLES

SETTING UP YOUR MACHINE TO RUN THE AMQ BROKEREXAMPLES

Downloading and installing Maven

Maven is required to run the AMQ Broker examples.

Procedure

1. Go to the [Apache Maven Download page](#) and download the latest distribution for your operating system.
2. Install Maven for your operating system.

Downloading and installing the AMQ Maven repository

After Maven is installed on your machine, you download and install the AMQ Maven repository. This repository is available on the Red Hat Customer Portal.

3. In a web browser, navigate to <https://access.redhat.com/downloads/> and log in. The Product Downloads page is displayed.
 4. In the Integration and Automation section, click the Red Hat AMQ Broker link. The Software Downloads page is displayed.
 5. Select the desired AMQ Broker version from the Version drop-down menu.
 6. On the Releases tab, click the Download link for the AMQ Broker Maven Repository. The AMQ Maven repository file is downloaded as a zip file.
 7. On your machine, unzip the AMQ repository file into a directory of your choosing.
A new directory is created on your machine, which contains the Maven repository in a subdirectory named **maven-repository/**.
-

Configuring the Maven settings file

After downloading and installing the AMQ Maven repository, you must add the repository to the Maven settings file.

Procedure

8. Open the Maven **settings.xml** file.

The **settings.xml** file is typically located in the **\${user.home}/.m2/** directory.

- For Linux, this is **~/.m2/**
- For Windows, this is **\Documents and Settings\.m2** or **\Users\.m2**

If you do not find a **settings.xml** file in **\${user.home}/.m2/**, there is a default version located in the **conf/** directory of your Maven installation. Copy the default **settings.xml** file into the **\${user.home}/.m2/** directory.

In the **<profiles>** element, add a profile for the AMQ Maven repository.

```
<!-- Configure the JBoss AMQ Maven repository -->
<profile>
  <id>jboss-amq-maven-repository</id>
  <repositories>
    <repository>
      <id>jboss-amq-maven-repository</id>
      <url>file://<JBoss-AMQ-repository path></url>
      <releases>
        <enabled>true</enabled>
      </releases>
      <snapshots>
        <enabled>>false</enabled>
      </snapshots>
    </repository>
  </repositories>
  <pluginRepositories>
    <pluginRepository>
      <id>jboss-amq-maven-repository</id>
      <url>file://<JBoss-AMQ-repository path></url>
      <releases>
        <enabled>true</enabled>
```

```

        </releases>
        <snapshots>
            <enabled>false</enabled>
        </snapshots>
    </pluginRepository>
</pluginRepositories>
>
</profile>

```

Replace **<JBoss-AMQ-repository-path>** with the location of the Maven repository that you installed. Typically, this location ends with **/maven-repository**. For example:

```

        <url>file:///path/to/repo/amq-broker-7.2.0-maven-
            repository/maven-repository</url>

```

In the **<activeProfiles>** element, set the AMQ Maven repository to be active:

```

<activeProfiles>
    <activeProfile>jboss-amq-maven-repository</activeProfile>
    ...
</activeProfiles>

```

9. If you copied the default **settings.xml** from your Maven installation, uncomment the **<active- profiles>** section if it was commented out by default.
10. Save and close **settings.xml**.
11. Remove the cached **\${user.home}/.m2/repository/** directory.
If your Maven repository contains outdated artifacts, you may encounter one of the following Maven error messages when you build or deploy your project:
 - **Missing artifact <artifact-name>**
 - **[ERROR] Failed to execute goal on project <project-name>; Could not resolve dependencies for <project-name>**

RUNNING AN AMQ BROKER EXAMPLE PROGRAM

Example programs for AMQ Broker demonstrate basic and advanced features of the product. You use Maven to run these programs.

Prerequisites

- Your machine is set up to run the AMQ Broker examples.
- You downloaded the [AMQ Broker example programs](#).

Procedure

1. Navigate to the directory of the example that you want to run. The following example assumed that you downloaded the examples to a directory called **amq-broker-examples**.

```
$ cd amq-broker-examples/examples/features/standard/queue
```

2. Use the **mvn clean verify** command to run the example program. Maven starts the broker and runs the example program. The first time you run the example program, Maven downloads any missing dependencies, which may take a while to run.

In this case, the **queue** example program is run, which creates a producer, sends a test message, and then creates a consumer that receives the message:

```
$ mvn clean verify
[INFO] Scanning for
projects...[INFO]
[INFO] -----< org.apache.activemq.examples.broker:queue >-----
-----
[INFO] Building ActiveMQ Artemis JMS Queue Example
2.6.1.amq-720004-redhat-1 [INFO].....[ jar ]...
...
server-out:2018-12-05 16:37:57,023 INFO
[org.apache.activemq.artemis.core.server] AMQ221001: Apache
ActiveMQ Artemis Message Broker version 2.6.1.amq-720004-
redhat- 1 [0.0.0.0, nodeId=06f529d3-f8d6-11e8-9bea-
0800271b03bd]
[INFO] Server
started[INFO]
```

```
[INFO] --- artemis-maven-plugin:2.6.1.amq-720004-redhat-1:runClient
(runClient) @ queue --
-
Sent message: This is a text
message Received message: This
is a text message[INFO]
[INFO] --- artemis-maven-plugin:2.6.1.amq-720004-redhat-1:cli (stop) @
queue ---
server-out:2018-12-05 16:37:59,519 INFO
[org.apache.activemq.artemis.core.server] AMQ221002: Apache
ActiveMQ Artemis Message Broker version 2.6.1.amq-720004-
redhat- 1 [06f529d3-f8d6-11e8-9bea-0800271b03bd] stopped,
uptime 3.734 seconds
server-out:Server stopped!
[INFO] .....
[INFO] BUILD SUCCESS
[INFO] .....
[INFO] Total time: 48.681 s
[INFO] Finished at: 2018 12-05T16:37:59-05:00
[INFO] .....
```



NOTE

Some of the example programs use UDP clustering, and may not work in your environment by default. To run these examples successfully, redirect traffic directed to 224.0.0.0 to the loopback interface:

```
$ sudo route add -net 224.0.0.0 netmask 240.0.0.0 dev lo
```