Red Hat AMQ 7.3

Using the AMQ Core Protocol JMS Client

For Use with AMQ Clients 2.4

Last Updated: 2019-06-18

Red Hat AMQ 7.3 Using the AMQ Core Protocol JMS Client

For Use with AMQ Clients 2.4

Legal Notice

Abstract

This guide describes how to install and configure the client, run hands-on examples, and use your client with other AMQ components.

Table of Contents

CHAPTER 1. OVERVIEW

AMQ Core Protocol JMS is a Java Message Service (JMS) 2.0 client for use in messaging applications that send and receive Artemis Core Protocol messages.

AMQ Core Protocol JMS is part of AMQ Clients, a suite of messaging libraries supporting multiple languages and platforms. For an overview of the clients, see AMQ Clients Overview. For information about this release, see AMQ Clients 2.4 Release Notes .

AMQ Core Protocol JMS is based on the JMS client from Apache ActiveMQ Artemis.

1.1. KEY FEATURES

- JMS 1.1 and 2.0 compatible
- SSL/TLS for secure communication
- Automatic reconnect and failover
- Distributed transactions (XA)
- Pure-Java implementation

1.2. SUPPORTED STANDARDS AND PROTOCOLS

AMQ Core Protocol JMS supports the following industry-recognized standards and network protocols:

- Version 2.0 of the Java Message Service API
- Modern TCP with IPv6

1.3. SUPPORTED CONFIGURATIONS

AMQ Core Protocol JMS supports the following OS and language versions.

- Red Hat Enterprise Linux 6, 7, and 8 with the following JDKs:
  - OpenJDK 8
  - Oracle JDK 8
  - IBM JDK 8
- Microsoft Windows 10 Pro with Oracle JDK 8
- Microsoft Windows Server 2012 R2 and 2016 with Oracle JDK 8

For more information, see Red Hat AMQ Supported Configurations .

1.4. TERMS AND CONCEPTS

This section introduces the core API entities and describes how they operate together.

Table 1.1. API terms

Entity	Description
ConnectionFactory	An entry point for creating connections.
Connection	A channel for communication between two peers on a network. It contains sessions.
Session	A context for producing and consuming messages. It contains message producers and consumers.
MessageProducer	A channel for sending messages to a destination. It has a target destination.
MessageConsumer	A channel for receiving messages from a destination. It has a source destination.
Destination	A named location for messages, either a queue or a topic.
Queue	A stored sequence of messages.
Topic	A stored sequence of messages for multicast distribution.
Message	An application-specific piece of information.

AMQ Core Protocol JMS sends and receives *messages*. Messages are transferred between connected peers using *message producers* and *consumers*. Producers and consumers are established over *sessions*. Sessions are established over *connections*. Connections are created by *connection factories*.

A sending peer creates a producer to send messages. The producer has a *destination* that identifies a target queue or topic at the remote peer. A receiving peer creates a consumer to receive messages. Like the producer, the consumer has a destination that identifies a source queue or topic at the remote peer.

A destination is either a *queue* or a *topic*. In JMS, queues and topics are client-side representations of named broker entities that hold messages.

A queue implements point-to-point semantics. Each message is seen by only one consumer, and the message is removed from the queue after it is read. A topic implements publish-subscribe semantics. Each message is seen by multiple consumers, and the message remains available to other consumers after it is read.

See the JMS tutorial for more information.

1.5. DOCUMENT CONVENTIONS

In this document, all file paths are valid for Linux, UNIX, and similar operating systems (for example, **/home/...**). If you are using Microsoft Windows, you should use the equivalent Microsoft Windows paths (for example, **C:\Users\...**).

CHAPTER 2. INSTALLATION

This chapter guides you through the steps to install AMQ Core Protocol JMS in your environment.

2.1. USING THE RED HAT MAVEN REPOSITORY

The client uses Apache Maven as its build tool. You can configure your Maven environment to download the library from the Red Hat Maven repository.

Procedure

1. Add the Red Hat repository to your Maven settings or POM file. For example configuration files, see Section B.1, “Using the online repository” .

```
<repository>
  <id>red-hat-ga</id>
  <url>https://maven.repository.redhat.com/ga</url>
</repository>
```

2. Add the library dependency to your POM file.

```
<dependency>
  <groupId>org.apache.activemq</groupId>
  <artifactId>artemis-jms-client</artifactId>
  <version>2.7.0.redhat-00056</version>
</dependency>
```

The client is now available in your Maven project.

2.2. INSTALLING A LOCAL MAVEN REPOSITORY

As an alternative to the online repository, AMQ Core Protocol JMS can be installed to your local filesystem as a file-based Maven repository. Note that AMQ Core Protocol JMS is delivered as part of the AMQ Broker component.

Procedure

1. Use your subscription to download the **AMQ Broker Maven Repository** zip file.

2. Extract the file contents into a directory of your choosing.
   On Linux or UNIX, use the **unzip** command to extract the file contents.

```
unzip amq-broker-<version>-maven-repository.zip
```

   On Windows, right-click on the zip file and select **Extract All**.

3. Configure Maven to use the repository in the **maven-repository** directory inside the extracted install directory. For more information, see Section B.2, “Using a local repository” .

2.3. INSTALLING THE ZIP FILE

AMQ Core Protocol JMS is delivered as part of the AMQ Broker component. The AMQ Broker zip file contains the examples and a distribution of the client libraries for those not using Maven. If you are using Maven and do not require the examples, you do not need to install the zip file.

Procedure

1. Use your subscription to download the **AMQ Broker** zip file.
2. Extract the file contents into a directory of your choosing.
   On Linux or UNIX, use the **unzip** command to extract the file contents.

   ```
   unzip amq-broker-<version>-bin.zip
   ```

   On Windows, right-click on the zip file and select **Extract All**.

   When you extract the contents of the zip file, a directory named **amq-broker-<version>** is created. This is the top-level directory of the installation and is referred to as **<install-dir>** throughout this document.
3. (Optional) If you are not using Maven, add the jar files in the **<install-dir>/lib** directory to your Java classpath.

CHAPTER 3. GETTING STARTED

This chapter guides you through a simple exercise to help you get started using AMQ Core Protocol JMS.

3.1. PREREQUISITES

- The example programs are located in the AMQ Broker zip file. To get started, you must install the zip file.
- To build the example, Maven must be configured to use the Red Hat repository or a local repository.

3.2. PREPARING THE BROKER

The example programs require a running broker with a queue named **exampleQueue**. Follow these steps to define the queue and start the broker:

Procedure

1. Install the broker.
2. Create a broker instance. Enable anonymous access.
3. Start the broker instance and check the console for any critical errors logged during startup.

```
$ <broker-instance-dir>/bin/artemis run
...
14:43:20,158 INFO  [org.apache.activemq.artemis.integration.bootstrap] AMQ101000: Starting ActiveMQ Artemis Server
...
15:01:39,686 INFO  [org.apache.activemq.artemis.core.server] AMQ221020: Started Acceptor at 0.0.0.0:5672 for protocols [AMQP]
...
15:01:39,691 INFO  [org.apache.activemq.artemis.core.server] AMQ221007: Server is now live
```

4. Use the **artemis queue** command to create a queue called **exampleQueue**.

```
<broker-instance-dir>/bin/artemis queue create --name exampleQueue --auto-create-address --anycast
```

You are prompted to answer a series of questions. For yes or no questions, type **N**. Otherwise, press Enter to accept the default value.

3.3. RUNNING YOUR FIRST EXAMPLE

Procedure

1. Use Maven to build the examples by running the following command in the **<install-dir>/examples/features/standard/queue** directory.

```
mvn clean package dependency:copy-dependencies -DincludeScope=runtime -DskipTests
```

The addition of **dependency:copy-dependencies** results in the dependencies being copied into the **target/dependency** directory.

2. Use the **java** command to run the example.
   On Linux or UNIX:

```
java -cp "target/classes:target/dependency/*"
org.apache.activemq.artemis.jms.example.QueueExample
```

   On Windows:

```
java -cp "target\classes;target\dependency\*"
org.apache.activemq.artemis.jms.example.QueueExample
```

The example creates a consumer and producer for a queue named **exampleQueue**. It sends a text message and then receives it back, printing the received message to the console.

Running it on Linux results in the following output.

```
$ java -cp "target/classes:target/dependency/*"
org.apache.activemq.artemis.jms.example.QueueExample
Sent message: This is a text message
Received message: This is a text message
```

The source code for the example is in the **<install-dir>/examples/features/standard/queue/src** directory. Additional examples are available in the **<install-dir>/examples/features/standard** directory.

APPENDIX A. USING YOUR SUBSCRIPTION

AMQ is provided through a software subscription. To manage your subscriptions, access your account at the Red Hat Customer Portal.

Accessing your account

1. Go to access.redhat.com.
2. If you do not already have an account, create one.
3. Log in to your account.

Activating a subscription

1. Go to access.redhat.com.
2. Navigate to **My Subscriptions**.
3. Navigate to **Activate a subscription** and enter your 16-digit activation number.

Downloading ZIP and TAR files

To access ZIP or TAR files, use the customer portal to find the relevant files for download. If you are using RPM packages, this step is not required.

1. Open a browser and log in to the Red Hat Customer Portal **Product Downloads** page at access.redhat.com/downloads.
2. Locate the **Red Hat AMQ** entries in the **JBOSS INTEGRATION AND AUTOMATION** category.
3. Select the desired AMQ product. The **Software Downloads** page opens.
4. Click the **Download** link for your component.

Registering your system for packages

To install RPM packages on Red Hat Enterprise Linux, your system must be registered. If you are using ZIP or TAR files, this step is not required.

1. Go to access.redhat.com.
2. Navigate to **Registration Assistant**.
3. Select your OS version and continue to the next page.
4. Use the listed command in your system terminal to complete the registration.

To learn more see How to Register and Subscribe a System to the Red Hat Customer Portal .

APPENDIX B. USING RED HAT MAVEN REPOSITORIES

This section describes how to use Red Hat-provided Maven repositories in your software.

B.1. USING THE ONLINE REPOSITORY

Red Hat maintains a central Maven repository for use with your Maven-based projects. For more information, see the repository welcome page.

There are two ways to configure Maven to use the Red Hat repository:

- Add the repository to your Maven settings
- Add the repository to your POM file

Adding the repository to your Maven settings

This method of configuration applies to all Maven projects owned by your user, as long as your POM file does not override the repository configuration and the included profile is enabled.

Procedure

1. Locate the Maven **settings.xml** file. It is usually inside the **.m2** directory in the user home directory. If the file does not exist, use a text editor to create it.
   On Linux or UNIX:

   ```
   /home/<username>/.m2/settings.xml
   ```

   On Windows:

   ```
   C:\Users\<username>\.m2\settings.xml
   ```

2. Add a new profile containing the Red Hat repository to the **profiles** element of the **settings.xml** file, as in the following example:

   Example: A Maven settings.xml file containing the Red Hat repository

   ```
   <settings>
     <profiles>
       <profile>
         <id>red-hat</id>
         <repositories>
           <repository>
             <id>red-hat-ga</id>
             <url>https://maven.repository.redhat.com/ga</url>
           </repository>
         </repositories>
         <pluginRepositories>
           <pluginRepository>
             <id>red-hat-ga</id>
             <url>https://maven.repository.redhat.com/ga</url>
             <releases>
               <enabled>true</enabled>
             </releases>
             <snapshots>
   ```

```
        <enabled>false</enabled>
      </snapshots>
    </pluginRepository>
  </pluginRepositories>
 </profile>
</profiles>
<activeProfiles>
 <activeProfile>red-hat</activeProfile>
</activeProfiles>
</settings>
```

For more information about Maven configuration, see the Maven settings reference.

Adding the repository to your POM file

To configure a repository directly in your project, add a new entry to the **repositories** element of your POM file, as in the following example:

Example: A Maven pom.xml file containing the Red Hat repository

```
<project>
 <modelVersion>4.0.0</modelVersion>

 <groupId>com.example</groupId>
 <artifactId>example-app</artifactId>
 <version>1.0.0</version>

 <repositories>
  <repository>
   <id>red-hat-ga</id>
   <url>https://maven.repository.redhat.com/ga</url>
  </repository>
 </repositories>
</project>
```

For more information about POM file configuration, see the Maven POM reference.

B.2. USING A LOCAL REPOSITORY

Red Hat provides file-based Maven repositories for some of its components. These are delivered as downloadable archives that you can extract to your local filesystem.

To configure Maven to use a locally extracted repository, apply the following XML in your Maven settings or POM file:

```
<repository>
 <id>red-hat-local</id>
 <url>${repository-url}</url>
</repository>
```

${repository-url} must be a file URL containing the local filesystem path of the extracted repository.

Table B.1. Example URLs for local Maven repositories

Operating system	Filesystem path	URL
Linux or UNIX	**/home/alice/maven-repository**	**file:/home/alice/maven-repository**
Windows	**C:\repos\red-hat**	**file:C:\repos\red-hat**

Revised on 2019-06-18 17:14:58 UTC