MDB Hello World With WildFly And ActiveMQ RAR Deployed as a Module

# Introduction

This is the 3rd increment of the [MDB Series](http://palashray.com/java/java/mdb/). In [Part 1](http://palashray.com/mdb-hello-world-with-wildfly-and-embedded-artemis-mq/), we spoke about what is an MDB? We then, took a simple design problem: a servlet posts user data to a Message Queue. The Message Broker then sends a notification to a MDB. We used the Artemis MQ, which comes embedded in Wildfly, as our Message Broker. In [Part 2](http://palashray.com/mdb-hello-world-with-wildfly-and-activemq-with-rar-deployment/), we demonstrated how to integrate a MDB with an external [Apache ActiveMQ](https://activemq.apache.org/) broker running in a separate process. We used the ActiveMQ RAR deployed independently to achieve integration between Wildfly and ActiveMQ.

In this increment, we are going to talk about how to deploy the ActiveMQ RAR as a Module in Wildfly.

# Implementation Details

We will use the sources from the [previous series](http://palashray.com/mdb-hello-world-with-wildfly-and-activemq-with-rar-deployment/) as a starting point. The previous source can be found here:

<https://github.com/paawak/blog/tree/master/code/mdb-demo/wildfly/external-activemq/rar-archive-deployment/mdb-activemq-rar-demo-spring>

We would keep all the Java and Resource files as is. The only change would be the standalone.xml.

## The standalone.xml

We need to rename the [standalone-with-external-activemq-rar-deployment.xml](https://github.com/paawak/blog/blob/master/code/mdb-demo/wildfly/external-activemq/rar-archive-deployment/mdb-activemq-rar-demo-spring/src/main/wildfly/standalone-with-external-activemq-rar-deployment.xml) to *standalone-with-activemq-module-deployment-spring.xml*. We would need to tweak this a bit.

### The mdb section

In the *mdb section*, we would need to change the name of the *resource-adapter-name* to *activemq-rar*.

|  |
| --- |
| <**mdb**>  <**resource-adapter-ref** resource-adapter-name="activemq-rar"/>  <**bean-instance-pool-ref** pool-name="mdb-strict-max-pool"/>  </**mdb**> |

### The resource-adapter section

The *resource-adpater section* needs some changes as well. We would need to have the *id* as *activemq-rar.rar*. Also, we would need to add a *module* section, and remove the existing *archive* section as shown below:

|  |
| --- |
| <**resource-adapters**>   <**resource-adapter** id="activemq-rar.rar">   <**module** slot="main" id="org.apache.activemq.ra" />   <**transaction-support**>XATransaction</**transaction-support**> |

Rest of the sections are identical. You can find the differences between the 2 versions here:

<https://github.com/paawak/blog/commit/356922b74e01c245e28186c152c3c56aa10ad4dd#diff-d39b537e052761d25f736f35d27b728a>

The complete *standalone-with-activemq-module-deployment-spring.xml* can be found here:

<https://github.com/paawak/blog/blob/master/code/mdb-demo/wildfly/external-activemq/module-deployment/mdb-activemq-module-demo-spring/src/main/wildfly/standalone-with-activemq-module-deployment-spring.xml>

# Source Code

The complete source can be found here:

<https://github.com/paawak/blog/tree/master/code/mdb-demo/wildfly/external-activemq/module-deployment/mdb-activemq-module-demo-spring>

# Running the Demo

Instructions for building the war, creating guest user and starting ActiveMQ remain the same. The only change is creating a module from the ActiveMQ RAR.

## Creating the ActiveMQ RAR module

### Downloading the RAR

The RAR can be downloaded from here:

<https://repo1.maven.org/maven2/org/apache/activemq/activemq-rar/>

### Creating the module directory structure

Create the module directory structure by running the below command:

mkdir -p WILDFLY\_HOME/modules/system/layers/base/org/apache/activemq/ra/main

### Extract the RAR contents

Copy the RAR file into the *main* directory. The, use the below command to extract the contents of the RAR:

jar -xvf activemq-rar-5.15.4.rar

### Copy the module.xml

Copy the *src/main/wildfly/module.xml* into the *main* directory.

cp src/main/wildfly/module.xml WILDFLY\_HOME/modules/system/layers/base/org/apache/activemq/ra/main/

The module.xml looks like this:

|  |
| --- |
| <?xml version='1.0' encoding='UTF-8'?> <**module** xmlns="urn:jboss:module:1.5"  name="org.apache.activemq.ra">  <**resources**>  <**resource-root** path="." />  <**resource-root** path="activemq-amqp-5.15.4.jar" />  <**resource-root** path="activemq-broker-5.15.4.jar" />  <**resource-root** path="activemq-client-5.15.4.jar" />  <**resource-root** path="activemq-jms-pool-5.15.4.jar" />  <**resource-root** path="activemq-kahadb-store-5.15.4.jar" />  <**resource-root** path="activemq-mqtt-5.15.4.jar" />  <**resource-root**  path="activemq-openwire-legacy-5.15.4.jar" />  <**resource-root** path="activemq-pool-5.15.4.jar" />  <**resource-root** path="activemq-protobuf-1.1.jar" />  <**resource-root** path="activemq-ra-5.15.4.jar" />  <**resource-root** path="activemq-spring-5.15.4.jar" />  <**resource-root** path="commons-net-3.6.jar" />  <**resource-root** path="commons-pool2-2.4.2.jar" />  <**resource-root**  path="geronimo-j2ee-management\_1.1\_spec-1.0.1.jar" />  <**resource-root** path="guava-18.0.jar" />  <**resource-root** path="hawtbuf-1.11.jar" />  <**resource-root** path="hawtdispatch-1.22.jar" />  <**resource-root** path="hawtdispatch-transport-1.22.jar" />  <**resource-root** path="jackson-annotations-2.9.4.jar" />  <**resource-root** path="jackson-core-2.9.4.jar" />  <**resource-root** path="jackson-databind-2.9.4.jar" />  <**resource-root** path="mqtt-client-1.14.jar" />  <**resource-root** path="proton-j-0.27.1.jar" />  <**resource-root** path="spring-aop-4.3.17.RELEASE.jar" />  <**resource-root** path="spring-beans-4.3.17.RELEASE.jar" />  <**resource-root** path="spring-context-4.3.17.RELEASE.jar" />  <**resource-root** path="spring-core-4.3.17.RELEASE.jar" />  <**resource-root**  path="spring-expression-4.3.17.RELEASE.jar" />  <**resource-root** path="xbean-spring-4.2.jar" />  </**resources**>  <**exports**>  <**exclude** path="org/springframework/\*\*" />  <**exclude** path="org/apache/xbean/\*\*" />  <**exclude** path="org/apache/commons/\*\*" />  <**exclude** path="org/aopalliance/\*\*" />  <**exclude** path="org/fusesource/\*\*" />  </**exports**>  <**dependencies**>  <**module** name="javax.api" slot="main" />  <**module** name="javax.jms.api" slot="main" />  <**module** name="javax.management.j2ee.api" slot="main" />  <**module** name="javax.resource.api" slot="main" />  <**module** name="org.slf4j" />  <**module** name="org.slf4j.impl" />  </**dependencies**> </**module**> |

## Starting WildFly

Now we can start Wildfly with our custom configuration as below:

WILDFLY\_HOME/bin/standalone.sh -c standalone-with-activemq-module-deployment-spring.xml

After Wildfly starts successfully, you can access the Author page with the below URL:

[http://localhost:8080/mdb-activemq-module-demo-spring/author.html](#)