

# Deploying the ActiveMQ Resource Adapter into IBM WebSphere 7.0

Torsten Mielke

Wednesday, June 27, 2012

## TABLE OF CONTENTS

Introduction.....	2
Part 1 - Getting your hands onto the ActiveMQ Resource Adapter .....	3
Part 2 - Deploying the ActiveMQ Resource Adapter .....	5
Part 3 - Creating an Activation Spec .....	9
Part 4 - Creating an Administered Object .....	11
Part 5 - Creating a Connection Factory .....	14
Part 6 - Deploying a Message Driven Bean that consumes from ActiveMQ.....	16

# DEPLOYING THE ACTIVEMQ RESOURCE ADAPTER INTO WEBSHERE 7.0.

## Introduction

ActiveMQ is a very popular JMS 1.1 compliant Message Broker. It supports many different transports that allow connectivity to pretty much any kind of client application written in any programming language. One common use case for consuming messages from a JMS Broker is via a Message Driven Bean (MDB) deployed into a J2EE Application Server.

Any J2EE compliant Application Server typically provides its own embedded JMS Broker. Message Driven Beans can consume JMS Messages from this JMS Broker instance. However there are scenarios where it is necessary to integrate your MDB with an external JMS Broker such as ActiveMQ.

The [Java Connector Architecture](#) specification (JCA) provides a common model for connecting your J2EE applications to external systems. A so-called JCA Resource Adapter wraps the client libraries of the external system that is to be integrated with in a .rar archive. This archive can then be deployed into the J2EE application server. J2EE applications can finally be configured to use these resource adapters.

ActiveMQ also offers a resource adapter that can be deployed into any J2EE Application Server. This allows J2EE applications and Message Driven Beans to communicate with an external ActiveMQ broker instance in order to consume and produce JMS messages.

Deploying the ActiveMQ resource adapter to a J2EE Application Server can be a tricky configuration exercise, depending on the Application Server used. This white paper presents step-by-step instructions on how to deploy the ActiveMQ 5.5.1 resource adapter into IBM WebSphere 7.0.1. I present a number of screen shots to illustrate the necessary configuration in WebSphere. Previous versions of the ActiveMQ Resource Adapter can be deployed in exactly the same way as well.

From a higher-level perspective the deployment involves these tasks:

*Part 1* explains how to obtain the ActiveMQ resource adapter, as it is not part of an ActiveMQ installation.

*Part 2* outlines the steps for deploying the resource adapter into WebSphere 7.0.

*Part 3* creates a corresponding Activation Spec in WebSphere, which uses this resource adapter.

*Part 4* defines the WebSphere Administered Objects, like the JMS destination.

*Part 5* finally configures the JMS Connection Factory that is used by the Message Driven Bean to connect to the external ActiveMQ broker instance.

*Part 6* deploys a sample Message Driven Bean that can be used for testing the deployment and configuration of the resource adapter. This part is optional.

## Part 1 - Getting your hands onto the ActiveMQ Resource Adapter

The ActiveMQ Resource Adapter is not shipped with your default ActiveMQ installation. You need to obtain it separately and there are essentially two options:

1. Download the precompiled ActiveMQ Resource Adapter from the FuseSource Maven repository, or
2. Build the resource adapter yourself from source code

I will discuss both options next.

### Option 1:

The perhaps easiest way to obtain the ActiveMQ Resource Adapter is to download it from the FuseSource Maven repository. You can get the latest and greatest version from

<http://repo.fusesource.com/nexus/content/groups/public/org/apache/activemq/activemq-rar/>

Select the required version and download the file with the .rar extension.

Unless you run the ActiveMQ broker on the same machine as the WebSphere Application Server, you will need to modify the configuration of this resource adapter. The embedded file META-INF/ra.xml contains that configuration. So if you need to change it, you will need to

- checkout this file  

```
jar xvf activemq-rar-5.5.1.rar META-INF/ra.xml
```
- modify it  

```
vi META-INF/ra.xml
```
- and check it in again  

```
jar uvf activemq-rar-5.5.1.rar META-INF
```

The possible [configuration properties](#) of the resource adapter are documented on the FuseSource web site. Once you have configured the resource adapter according to your environment, you can proceed to part 2.

### Option 2:

If you don't like to fiddle with jar files manually and rather prefer full control over the resource adapter configuration, then you can alternatively check out the source code of ActiveMQ and build the resource adapter yourself from that source. The advantage of this option is that you can modify the configuration of the resource adapter upfront before creating the .rar archive. Also, after making a change to the configuration it is easy to simply rebuild the .rar file.

A) You can download the source of the Fuse distributions of ActiveMQ from

<http://repo.fusesource.com/nexus/content/groups/public/org/apache/activemq/apache-activemq/>

(select the -src.zip or -src.tar.gz files). Unzip/untar the downloaded file.

B) Alternatively the trunk version of Apache is available for checkout using this Subversion command  
`svn co https://svn.apache.org/repos/asf/activemq/trunk activemq`

The trunk version is obviously not recommended for being used in production, so you can choose any release version from <https://svn.apache.org/repos/asf/activemq/tags/>

Whatever way you choose to download the sources, you will find the same directory structure once the source are downloaded and extracted on your hard drive.

The subdirectory `activemq-ra` contains the resource adapter specific source code. The directory `activemq-rar` does not contain any source code but is used to package the `.rar` file and includes any client side ActiveMQ runtime libraries when building the `.rar` file. Most importantly it also contains the resource adapter configuration in `src/main/rar/META-INF/ra.xml`.

It is in this file where you can change the `serverUrl` other settings for connecting to your external ActiveMQ broker instance. The possible [configuration properties](#) of the resource adapter are documented on the FuseSource web site.

Once you made any necessary changes to `src/main/rar/META-INF/ra.xml`, you're ready to kick off a build of the resource adapter. The easiest way is to build the entire ActiveMQ distribution.

Simply run a `mvn clean install -Dtest=0` from the top level project directory and after a few minutes the build should be finished. The build time highly depends on how many Maven artifacts you need to download to your local repository.

The resulting resource adapter `.rar` file can be found in the `activemq-rar/target` folder.

This file will get deployed into WebSphere 7.0 in the next part.

## Part 2 - Deploying the ActiveMQ Resource Adapter

In this part we will deploy the ActiveMQ Resource Adapter into IBM Websphere 7.0. If you want to test this deployment first, then its best to do this using your own WebSphere profile.

So let's start with creating a new profile in WebSphere so that we do not to interact with any existing applications.

- Create a new WAS profile to deploy this resource adapter. Using the WebSphere command line tool:
 

```
./manageprofiles.sh -create -templatePath
/opt/java/WebSphere7/profileTemplates/default/ -profileName AMQ-5.5.1-RAR-Test -
profilePath /opt/java/WebSphere7/profiles/AMQ-5.5.1-RAR-Test
```

 Alternatively you can use the GUI application.
- Start up WebSphere using that profile.
- Log into the WebSphere administration console from your browser.
- On the left pane select Resources – Resource Adapters – Resource Adapters. This should bring you to the following screen:

The screenshot displays the IBM WebSphere Administration Console interface. The left-hand navigation pane is expanded to show the path: **Resources** > **Resource Adapters**. The main content area is titled "Resource adapters" and contains the following elements:

- Resource adapters**: A heading followed by a descriptive paragraph about connecting applications to an Enterprise Information System (EIS).
- Scope**: A dropdown menu currently set to "All scopes". Below it, a note explains that the scope specifies the level at which the resource definition is visible.
- Preferences**: A section containing buttons for "Install RAR", "New", "Delete", and "Update RAR", along with icons for refresh, add, and delete.
- Table**: A table with columns "Select", "Name", and "Scope". The table is currently empty, showing "Total 0" at the bottom.

- Next click on the “Install RAR” link.
- Enter full path to the file **activemq-rar-5.5.1.rar**
- 

Resource adapters

[Close page](#)

### Install RAR File

Use this page to install a RAR file in one of two ways. You can either upload a RAR file from the local file system, or specify an existing RAR file on a server. The RAR file must be installed at the node level, and you can select the node below.

**Scope**

\* Node  
linux-3i3bNode03

**Path**

Local file system  
Full path  
/home/tmielke/activemq-rar-5.3.0.8-fuse.rar

Remote file system  
Full path

- Click Next
- Leave the default settings for “General Properties”

Integrated Solutions Console Welcome websphere Help | Logout

View: All tasks

- Welcome
- ▣ Guided Activities
- ▣ Servers
- ▣ Applications
  - New Application
  - ▣ Application Types
- ▣ Services
- ▣ Resources
  - Schedulers
  - Object pool managers
  - ▣ JMS
  - ▣ JDBC
  - ▣ Resource Adapters
    - Resource adapters
    - J2C connection factories
    - J2C activation specifications
    - J2C administered objects
  - ▣ Asynchronous beans
  - ▣ Cache instances
  - ▣ Mail
  - ▣ URL
  - ▣ Resource Environment
- ▣ Security
- ▣ Environment
- ▣ System administration
- ▣ Users and Groups
- ▣ Monitoring and Tuning
- ▣ Troubleshooting
- ▣ Service integration
- ▣ UDDI

**Resource adapters > Resource adapters**

Use this page to manage resource adapters, which provide the fundamental interface for connecting applications to an Enterprise Information System (EIS). The WebSphere(R) Relational Resource Adapter is embedded within the product to provide access to relational databases. To access another type of EIS, use this page to install a standalone resource adapter archive (RAR) file. You can configure multiple resource adapters for each installed RAR file.

Configuration

**General Properties**

\* Scope

Name

Description

Archive path

Class path

Native library path

Isolate this resource provider

OK Reset Cancel

- Click OK and Save the configuration



Cell=linux-3i3bNode03Cell, Profile=AMQ-5.3-RAR-Test

Resource adapters ? -

Messages

- ⚠ Changes have been made to your local configuration. You can:
  - [Save](#) directly to the master configuration.
  - [Review](#) changes before saving or discarding.
- ⚠ The server may need to be restarted for these changes to take effect.

**Resource adapters**

Use this page to manage resource adapters, which provide the fundamental interface for connecting applications to an Enterprise Information System (EIS). The WebSphere(R) Relational Resource Adapter is embedded within the product to provide access to relational databases. To access another type of EIS, use this page to install a standalone resource adapter archive (RAR) file. You can configure multiple resource adapters for each installed RAR file.

Scope: =All scopes

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#).

All scopes ▾

Preferences

Install RAR
New
Delete
Update RAR

☑
📄
↕
🔍

Select	Name ↕	Scope ↕
You can administer the following resources:		
<input type="checkbox"/>	<a href="#">ActiveMQ JMS Resource Adapter</a>	Node=linux-3i3bNode03
Total 1		

## Part 3 - Creating an Activation Spec

With the Resource Adapter being deployed successfully now, its time to create an Activation Spec.

- From the main page select Resources – Resource Adapters – Resource Adapters – ActiveMQ JMS Resource Adapter

Integrated Solutions Console Welcome websphere

Cell=linux-3i3bNode03Cell, Profile=AMQ-5.3-RAR-Test

View: All tasks

Resource adapters

Resource adapters > ActiveMQ JMS Resource Adapter

Use this page to manage resource adapters, which provide the fundamental interface for connecting applications to an Enterprise Information System (EIS). The WebSphere(R) Relational Resource Adapter is embedded within the product to provide access to relational databases. To access another type of EIS, use this page to install a standalone resource adapter archive (RAR) file. You can configure multiple resource adapters for each installed RAR file.

Configuration

**General Properties**

- \* Scope: cells:linux-3i3bNode03Cell:nodes:linux-3i3bNode03
- \* Name: ActiveMQ JMS Resource Adapter
- Description: ActiveMQ inbound and outbound JMS ResourceAdapter
- \* Archive path: \${CONNECTOR\_INSTALL\_ROOT}/activemq-rar-5.3.0.8-fuse.rar
- Class path: \${CONNECTOR\_INSTALL\_ROOT}/activemq-rar-5.3.0.8-fuse.rar

**Additional Properties**

- J2C connection factories
- J2C activation specifications
- J2C administered objects
- Advanced resource adapter properties
- Custom properties
- View Deployment Descriptor

- Click on “J2C activation specification” link
- Select New and fill in these values:
  - Name: **AMQActivationSpec**
  - JNDI Name: **AMQActivationSpec**
  - Message listener type: **javax.jms.MessageListener supported by org.apache.activemq.ra.ActiveMQActivationSpec**
- Click OK and Save the configuration

Configuration \_\_\_\_\_

**General Properties**

\* Scope

\* Provider

\* Name

JNDI name

Description

**Security settings**

Select the authentication values for this resource.

Authentication alias

\* Message listener type

## Part 4 - Creating an Administered Object

- Select Resources – Resource Adapters – Resource Adapters – ActiveMQ JMS Resource Adapter
- Click on J2C administered objects link

- Select New and fill in these values

Name: **testqueue**  
 JNDI name: **jms/testqueue**  
 Administered object class: **org.apache.activemq.command.ActiveMQQueue**  
**implements javax.jms.queue**

[Resource adapters](#) > [ActiveMQ JMS Resource Adapter](#) > [J2C administered objects](#) > [jms/testqueue](#)

Use this page to configure J2C administered objects (AOs), which are provided by resource adapters. Some messaging styles might need applications to use special administered objects for sending and synchronously receiving messages (through connection objects using messaging style-specific APIs). It is also possible that AOs might be used to perform transformations on an asynchronously received message in a message provider-specific way. You can access administered objects by a component by using either a resource-env-ref or a message-destination-ref (preferred).

Configuration

### General Properties

\* Scope

cells:linux-3i3bNode03Cell:nodes:linux-3i3bNode03

\* Provider

ActiveMQ JMS Resource Adapter

\* Name

testqueue

JNDI Name

jms/testqueue

Description

\* Administered object class

org.apache.activemq.command.ActiveMQQueue implements javax.jms.Queue

Apply

OK

Reset

Cancel

### Additional Properties

- [J2C administered objects custom properties](#)

- Click OK. This brings you back to the list of administered objects
- Click on testqueue again on the “J2C administered objects custom properties link”

**Resource adapters**

Messages

⚠ Changes have been made to your local configuration. You can:

- [Save](#) directly to the master configuration.
- [Review](#) changes before saving or discarding.

⚠ The server may need to be restarted for these changes to take effect.

[Resource adapters](#) > [ActiveMQ JMS Resource Adapter](#) > [J2C administered objects](#) > [jms/testqueue](#) > **Custom properties**

Use this page to specify custom properties that your enterprise information system (EIS) requires for the resource providers and resource factories that you configure. For example, most database vendors require additional custom properties for data sources that access the database.

Preferences

Name	Value	Description	Required
You can administer the following resources:			
<a href="#">compositeDestinations</a>		compositeDestinations	false
<a href="#">PhysicalName</a>		physicalName	false
<a href="#">properties</a>		properties	false
Total 3			

- Select physical name and give it the value **testqueue**.
- Click OK and Save the configuration.

[Resource adapters](#) > [ActiveMQ JMS Resource Adapter](#) > [J2C administered objects](#) > [jms/testqueue](#) > [Custom properties](#) > [PhysicalName](#)

Use this page to specify custom properties that your enterprise information system (EIS) requires for the resource providers and resource factories that you configure. For example, most database vendors require additional custom properties for data sources that access the database.

Configuration

---

**General Properties**

\* Scope  
cells:linux-3i3bNode03Cell:nodes:linux-3i3bNode03

Required

Name  
PhysicalName

Value  
testqueue

Description  
physicalName

Type  
java.lang.String

## Part 5 - Creating a Connection Factory

Finally we need to create a JMS Connection Factory.

- Select Resources – Resource Adapters – Resource Adapters – ActiveMQ JMS Resource Adapter
- Click “J2C connection factories” link
- Select New and enter these values:  
Name: **AMQ-CF**  
JNDI Name: **jms/AMQ-CF**  
Connection factory interface: **javax.jms.ConnectionFactory**
- Leave defaults for all other settings
- Click OK and Save the configuration

**General Properties**

\* Scope

\* Provider

\* Name

JNDI name

Description

\* Connection factory interface

Category

**Security settings**

Select the authentication values for this resource.

Authentication alias for XA recovery

Component-managed authentication alias

Mapping-configuration alias

Container-managed authentication alias

- There is no need to further configure any custom properties of the Connection Factory.

These are all the JMS resources that need to be configured in WebSphere in order to connect to an ActiveMQ broker. I suggest to restart WebSphere after completing all steps of part 5 so that these changes can take effect.



## Part 6 - Deploying a Message Driven Bean that consumes from ActiveMQ

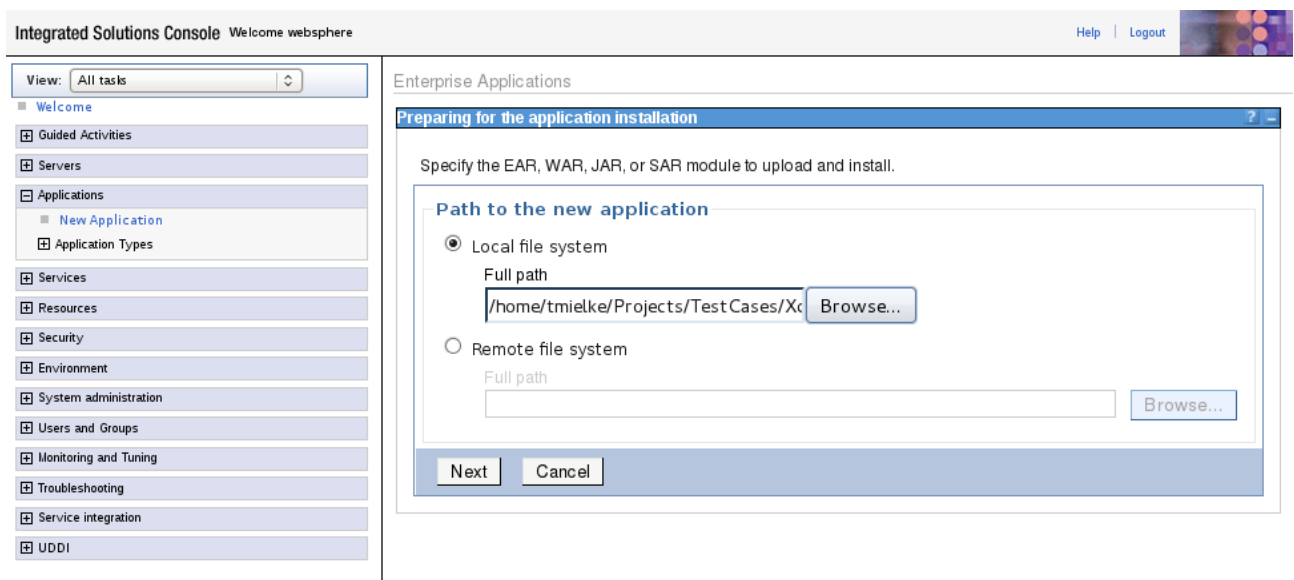
Your ActiveMQ Resource Adapter is now successfully deployed and ready to be used. When deploying a J2EE application that uses a message driven bean, the idea is to bind this MDB to the Activation Specification with the name “AMQActivationSpec” that we created in part 3. That’s about all that’s needed.

Lets go through these steps one by one. As a demo MDB application you can download and deploy the SimpleMDB demo from this location:

[http://www.swiftmq.com/developers/howto\\_jmsintravm\\_was6/simplemdb/index.html](http://www.swiftmq.com/developers/howto_jmsintravm_was6/simplemdb/index.html)

Once downloaded, unzip the file and verify that it unzipped the file simplemdb/simplemdb.ear. Then follow these steps to deploy the Message Driven Bean demo.

- In the WebSphere Administration Console select Applications – New Application – New Enterprise Application.
- Browse to the extracted demo directory on your local file system and select simplemdb.ear file.



- Click Next.
- Choose the Fast Path installation option and click Next.

- In “Step 1: Select installation options” leave all the default settings and click Next.

Specify options for installing enterprise applications and modules.

→ **Step 1: Select installation options**

[Step 2: Map modules to servers](#)

✦ [Step 3: Bind listeners for message-driven beans](#)

[Step 4: Summary](#)

### Select installation options

Specify the various options that are available to prepare and install your application.

Precompile JavaServer Pages files

Directory to install application

Distribute application

Use Binary Configuration

Deploy enterprise beans

Application name

Create MBeans for resources

Override class reloading settings for Web and EJB modules

Reload interval in seconds

Deploy Web services

Validate Input off/warn/fail

Process embedded configuration

---

**File Permission**

Allow all files to be read but not written to

Allow executables to execute

Allow HTML and image files to be read by everyone

---

Application Build ID

Allow dispatching includes to remote resources

Allow servicing includes from remote resources

- In “Step 2: Map modules to server” you can also go with the default settings.

Cell=linux-3i3bNode01 Cell, Profile=ActiveMQ-RAR-Test

The screenshot shows the 'Map modules to servers' step of the 'Install New Application' wizard. The left sidebar shows four steps: Step 1 (Select installation options), Step 2 (Map modules to servers - highlighted), Step 3 (Bind listeners for message-driven beans), and Step 4 (Summary). The main area contains instructions and a table for mapping modules to servers.

Specify options for installing enterprise applications and modules.

**Map modules to servers**

Specify targets such as application servers or clusters of application servers where you want to install the modules that are contained in your application. Modules can be installed on the same application server or dispersed among several application servers. Also, specify the Web servers as targets that serve as routers for requests to this application. The plug-in configuration file (plugin-ctg.xml) for each Web server is generated, based on the applications that are routed through.

Clusters and servers:

Select	Module	URI	Server
<input type="checkbox"/>	SimpleMessageJAR	simplemdb-ejb.jar,META-INF/ejb-jar.xml	WebSphere:cell=linux-3i3bNode01 Cell,node=linux-3i3bNode01,server=server1

Buttons: Previous, Next, Cancel

- Next is “Step 3: Bind listeners for message driven beans”. Select the “Activation Specification” radio button and supply the Target Resource JNDI Name “AMQActivationSpec”. This is the name of the Activation Spec that we created in step 3.

Cell=linux-3i3bNode01 Cell, Profile=ActiveMQ-RAR-Test

The screenshot shows the 'Bind listeners for message-driven beans' step of the 'Install New Application' wizard. The left sidebar shows four steps: Step 1 (Select installation options), Step 2 (Map modules to servers), Step 3 (Bind listeners for message-driven beans - highlighted), and Step 4 (Summary). The main area contains instructions and a table for binding listeners.

Specify options for installing enterprise applications and modules.

**Bind listeners for message-driven beans**

Each message-driven enterprise bean in your application or module must be bound to a listener port name or to an activation specification JNDI name. When a message-driven enterprise bean is bound to an activation specification JNDI name you can also specify the destination JNDI name and authentication alias.

Apply Multiple Mappings

Select	EJB module	EJB	URI	Messaging type	Listener Bindings
<input checked="" type="checkbox"/>	SimpleMessageJAR	SimpleMessageEJB	simplemdb-ejb.jar,META-INF/ejb-jar.xml	javax.jms.MessageListener	<input type="radio"/> Listener port Name <input type="text"/> <input checked="" type="radio"/> Activation Specification Target Resource JNDI Name <input type="text" value="AMQActivationSpec"/> Destination JNDI name <input type="text"/> ActivationSpec authentication alias <input type="text"/>

Buttons: Previous, Next, Cancel

- Click Next and Finish.

- The MDB gets deployed now.  
Once the deployment has completed click Save and start the SimpleMessageApp application.

If all things are configured correctly, then the following logging output should appear in SystemOut.log of WebSphere:

```
[13/06/12 13:37:57:897 CEST] 00000016 ActiveMQEndpo I org.apache.activemq.ra.ActiveMQEndpointWorker
start Starting
[13/06/12 13:37:57:907 CEST] 00000016 ActivationSpe I J2CA0523I: The Message Endpoint for
ActivationSpec AMQActivationSpec (org.apache.activemq.ra.ActiveMQActivationSpec) and MDB Application
SimpleMessageApp#simplemdb-ejb.jar#SimpleMessageEJB is activated.
[13/06/12 13:37:57:909 CEST] 0000001e ActiveMQEndpo I org.apache.activemq.ra.ActiveMQEndpointWorker$1
run Establishing connection to broker [tcp://192.168.178.31:61616]
[13/06/12 13:37:57:933 CEST] 00000016 EJBContainerI I WSVR0057I: EJB jar started: simplemdb-ejb.jar
[13/06/12 13:37:57:952 CEST] 00000016 ApplicationMg A WSVR0221I: Application started:
SimpleMessageApp
[13/06/12 13:37:57:956 CEST] 00000016 CompositionUn A WSVR0191I: Composition unit
WebSphere:cuname=SimpleMessageApp in BLA WebSphere:blaname=SimpleMessageApp started.
[13/06/12 13:37:58:583 CEST] 0000001e ActiveMQEndpo I org.apache.activemq.ra.ActiveMQEndpointWorker$1
run Successfully established connection to broker [tcp://192.168.178.31:61616]
```

This logging output indicates that your MDB successfully connected to the ActiveMQ broker.

Additional documentation on the ActiveMQ Resource Adapter can be found in the documentation section of the FuseSource.com website, like an [introduction](#) to the resource adapter and the resource adapter [configuration properties](#).

### Contact FuseSource

FuseSource, 14 Oak Park, Bedford, MA 01730 USA  
On the Web: [fusesource.com](http://fusesource.com) Email: [info@fusesource.com](mailto:info@fusesource.com)  
Toll Free: 888.280.5380 Tel: 781.382.4002 Outside US: +44 (0) 1 344 386 344

### FuseSource

FuseSource is the leader in open source integration solutions, and offers enterprise-class products, services and subscriptions used by Fortune 500 companies worldwide. The cost savings and flexibility of open source software combined with FuseSource's experience in building mission-critical applications make it possible to integrate today what was never before possible.

© 2012 FuseSource Corp. All rights reserved.