



IBM Software Group

IBM® WebSphere® Application Server V6

Java™ Message Service (JMS) Resource Management



@business on demand.

© 2005 IBM Corporation
Updated March 2, 2005

This presentation will focus on managing JMS Resources used by the Default Messaging Provider.

Goals

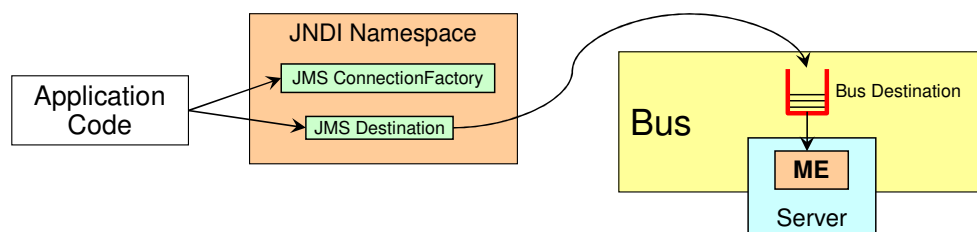
- Provide an overview of creating JMS resources using administrative clients
 - ▶ JMS Connection Factories
 - ▶ JMS Destinations
 - ▶ JMS Activation Specifications



The goal of this presentation is to provide you with an overview of how to create and manage JMS resources, including JMS Connection Factories, Destinations, and Activation Specifications.

Using Bus and JMS Destinations

- To use a JMS Queue or Topic, you need 2 different layers of administratively created objects
 - ▶ Bus Destination – Queue or TopicSpace
 - This destination is on the bus and is invisible to the application
 - ▶ JMS Destination – JMS Queue or JMS Topic
 - This destination acts as a proxy or pointer to the destination on the bus



In order to use a JMS Queue or Topic you need to create two layers of messaging resources. WebSphere Application Server uses its underlying service integration technology to implement JMS messaging.

This means that before you can define a JMS destination, you will have to create a destination on your bus. This Destination is invisible to the application, which will only be aware of the JMS Destination. The JMS Destination exists in the JNDI namespace and acts as a pointer to the underlying Bus Destination. Application code will also need to make use of a JMS ConnectionFactory.

The JMS Destinations and ConnectionFactories documented here are specific to the Service Integration Bus implemented by WebSphere Application Server and as such are referred to as SIBJMSTDestinations and SIBJMSTConnectionFactories in the commands that create them. Other JMS Providers (such as WebSphere MQ) would have their own ConnectionFactories and Destinations.

As with all JMS Applications, binding the ConnectionFactories and Destinations to JNDI is an administrative task, and these resources are used by application code to connect to the underlying destination using the JMS API.

Explanation of Destinations

- Point-to-Point messaging
 - ▶ Create a Queue on the bus
 - ▶ Create JMS Queue

- Publish-Subscribe Messaging
 - ▶ Create a TopicSpace on the bus (or use the default TopicSpace)
 - ▶ Create JMS Topic



There are two types of JMS destinations. Point-to-point messaging is implemented using queues. To use a JMS queue, first create a Queue on the appropriate Service Integration Bus. Then create a JMS Queue that will act as a pointer to this queue.

Publish-and-subscribe messaging uses JMS Topics. To create a JMS Topic, first create a TopicSpace on your Service Integration Bus. Then create a JMS Topic. As with Queues, this Topic will be bound to the namespace and allow applications to access the underlying TopicSpace.

Configuring JMS Resources



- Expand Resources > JMS Providers > Default Messaging

Configuration

☐ Scope: Cell=2kserver6Cell01, Node=2kserver6Node01

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](#).

Cell
2kserver6Cell01

→ Node
2kserver6Node01

Server

General Properties

Scope
cells:2kserver6Cell01:nodes:2kserver6Node01

Name
Default messaging provider

Description
Messaging provider for connection to the service integration bus.

Connection Factories

- [JMS connection factory](#)
- [JMS queue connection factory](#)
- [JMS topic connection factory](#)

Destinations

- [JMS queue](#)
- [JMS topic](#)

Activation Specifications

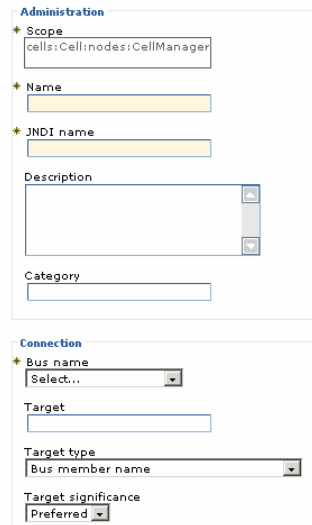
- [JMS activation specification](#)



JMS resources can be created from the Administrative Console by expanding the Resources link, and then the JMS Providers link, in the main menu. By clicking on the Default Messaging link, you will reveal a panel that allows you to create Queues, Topics, ConnectionFactories, and Activation Specifications that utilize the Default Messaging Provider, which is the Service Integration Bus technology. Be sure to select the appropriate scope for the resources you create. To select Cell scope, clear both the “Node” and “Server” fields, and click Apply.

Creating JMS Connection Factory

- Used to obtain connections for both point-to-point and pub-sub messaging styles
 - ▶ A unified Connection Factory is new in JMS 1.1
 - ▶ Enter Name, Java Naming and Directory Interface (JNDI) name, message reliability, Bus name, remote target type, and so on.
- You can also create a Queue Connection Factory and a Topic Connection Factory separately



Administration

* Scope
cells:Cell:nodes:CellManager

* Name

* JNDI name

Description

Category

Connection

* Bus name
Select...

Target

Target type
Bus member name

Target significance
Preferred

Unlike JMS 1.0, which used different Connection Factory objects for Queues and Topics, JMS 1.1 supports the use of a single unified Connection Factory, called a Connection Factory, for the creation of both destination types.

To create a Connection Factory in the Administrative Console, you need to specify a name, which will be used to refer to the object administratively, as well as a JNDI name, which will be used by your application code to locate the Connection Factory. You can also set the default reliability level for destinations created with this Connection Factory on this page.

If you have older code that still uses Queue Connection Factories or Topic Connection Factories, you can create those as well in a similar fashion. Just choose the appropriate link in the Administrative Console.

Creating JMS Queue

- Represents the Queue that applications interact with
 - ▶ The application-level view of the Service Integration Bus Queue



Administration

* Scope
cells:zkserver6Cell01

* Name

* JNDI name

Description

Connection

Bus name
Select...

* Queue name
Select...

Delivery mode
Application

Time to live
milliseconds

Priority

Advanced

Read ahead
Enabled

The JMS Queue is the object with which applications interact, and which an administrator configures as a JMS resource on the messaging provider. An application that uses JMS point-to-point messaging acts as a producer or consumer of messages from JMS queues, and has no need to know about the underlying service integration resources that support the JMS queue.

The JMS Queue link in the Administrative Console allows you to create a JMS Queue by specifying a JNDI name that will be bound in the Java namespace as well as the name of a Queue defined on the Service Integration Bus, as shown here. Creating this resource enables applications to interact with the Bus-level queue by working with the JMS Queue Java object.

A JMS Topic can be created in a similar fashion. There is a separate link in the Administrative Console, simply use that link to associate a JMS Topic with a Bus-level TopicSpace.

Creating JMS Resources with wsadmin

- Most of the new Service Integration Bus administration commands use the new \$AdminTask wsadmin commands
 - ▶ To list all of the \$AdminTask commands available use:
 - \$AdminTask help -commands
 - ▶ To get detailed information about a specific command:
 - \$AdminTask help <command>
 - ▶ \$scope parameter specifies the level at which the resource is visible, just as when creating a queue using the Administrative Console

- Example (for creating a JMS Queue)

```
set jmsQName "Sample.JMS.Q1"
set jmsQJNDI "Sample/JMS/Q1"
set jmsQDescr "Sample JMS Queue"
set QName "SIB_Q1"
set params [list -name $jmsQName -jndiName $jmsQJNDI -description $jmsQDescr -queueName
             $QName]
$AdminTask createSIBJMSQueue $scope $params
```

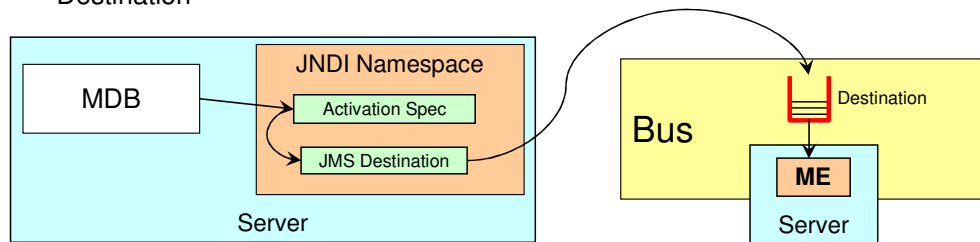


In addition to using the Administrative Console, JMS resources can also be created using the wsadmin command-line client. This functionality is implemented using a new wsadmin object called AdminTask. For example, you can call the AdminTask command named createSIBJMSQueue, and pass it a parameter list that both the JNDI name and the underlying Queue name to create a JMS queue from the command line, as shown here.

To learn more about the new AdminTask object, type "\$AdminTask help" in wsadmin. Similar syntax, shown here, can be used to get help on any specific AdminTask command.

EJB 2.1 Message Driven Beans

- Message Driven Beans (MDBs) run inside WebSphere Application Server, and get messages from Queues and Topics by way of Activation Specifications
- EJB 2.0 uses Listener Ports to connect an MDB to a Destination
 - ▶ Listener Ports are also used if WebSphere MQ is the JMS provider
- The Bus uses J2EE Connector Architecture (JCA) connectors to connect the MDB to the Destination
 - ▶ Create an Activation Specification object and bind it into the namespace
 - ▶ When the MDB is started the Activation Specification connects it to a Destination



Message Driven Beans run inside an Application Server and receive messages from Queues and Topics. This is how an Application running inside the server can listen to a JMS destination. In previous WebSphere Application Server releases, an MDB connected to a destination using a ListenerPort object. The new service integration technologies use JCA connectors to connect the MDB to a destination, and as such you now create an Activation Specification object and bind it into the namespace. When the MDB is started it uses the Activation Specification to connect it to the Destination (Queue or Topic). During this startup, any properties set in the MDB's <activation-config-properties> are used to override any properties that currently exist for the Activation Specification.

In WebSphere Application Server V6, the JCA 1.5 message inflow contract is used for message delivery to all EJB2.1 MDBs. The Service Integration technology and other third party JMS resource adapters implement this contract natively. For providers that only support ASF an 'ASF resource adapter' is provided that maps between ASF and JCA message inflow. The properties on the Activation Specification are the union of those from the MDB and those that would have been defined on the ListenerPort.

MDB Administration with wsadmin

- The Activation Specification contains the JNDI name of a JMS Destination to which an MDB will listen
- Creating Activation Specification using wsadmin
 - ▶ set params [list -name \$activationSpecName -jndiName \$activationSpecJNDIName -busName \$BusName -destJndiName \$destinationJNDIName -destinationType \$destinationType]
 - ▶ \$AdminTask createSIBJMSActivationspec \$scope \$params



To create an ActivationSpec using wsadmin, use the AdminTask command createSIBJMSActivationSpec. It takes a list of parameters as shown here that provides information about the ActivationSpec itself as well as the JMS destination to which it points.

Creating Activation Specification

- Resources > Default Messaging Provider > JMS activation specification
- When installing the application, bind the MDB to the JNDI name of the Activation Specification

General Properties

Administration

* Scope

* Name

* JNDI name

Destination

* Destination type
Queue

* Destination JNDI name

Message selector

Bus name
Select...

Acknowledge mode
Auto-acknowledge



An Activation Specification can also be created using the Administrative Console, using the [JMS Activation Specification](#) link provided on the configuration page for the Messaging Provider. Again, you will provide a JNDI name for the ActivationSpec as well as specify the destination to which it points.

At deployment time, you will bind your Message-Driven Bean to the JNDI name of the Activation Specification.

Support for EJB 2.0 MDBs

- Existing EJB 2.0 MDBs can be deployed against a Listener Port
 - ▶ Same as WebSphere Application Server V5
- EJB 2.0 MDBs can also use Activation Specifications
- New EJB 2.1 MDBs should use Activation Specifications



WebSphere Application Server V6 also supports deploying EJB 2.0 MDBs. For compatibility, they can be configured to use Listener Ports, as they did in version 5, or they can use the newer Activation Specifications if you want to modify your application code.

When creating EJB 2.1 MDBs, you should always use Activation Specifications.

Summary

- JMS resource management can be performed using the Administrative Console or wsadmin
 - ▶ JMS Connection Factories
 - ▶ JMS Destinations
 - ▶ JMS Activation Specifications



In summary, this presentation has focused on creating JMS resources using the Administrative Console and wsadmin. The JMS resources that were covered include JMS Connection Factories, JMS Destinations, and JMS Activation Specifications. For information on creating the underlying resources on a Service Integration Bus, refer to the presentation titled “Service Integration Bus – Resource Management”.

Trademarks, Copyrights, and Disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	CICS	IMS	MQSeries	Tivoli
IBM (logo)	Cloudscape	Informix	OS/390	WebSphere
e(logo)/business	DB2	iSeries	OS/400	xSeries
AIX	DB2 Universal Database	Lotus	pSeries	zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product and service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2004. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.