

S H A R E

Technology • Connections • Results

Getting WMQ messages into WebSphere Application Server (including from WebSphere Message Broker)

Session 8714

Simon Gormley – WebSphere MQ JMS Level 3 Service,
IBM Hursley, sgormley@uk.ibm.com

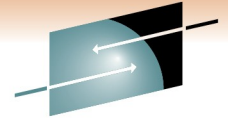
David Coles – WebSphere Message Broker Level 3 Service,
IBM Hursley – dcoles@uk.ibm.com

Thursday 3rd March 2011

SHARE
in Anaheim
2011



Agenda

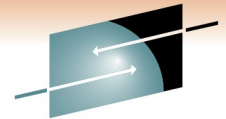


SHARE
Technology • Connections • Results

- **The Scenario**
- Initial Setup
 - Queue Manager
 - Connection Factories
 - Destinations
 - Activation Specifications
- Application Development
 - Message-driven beans (MDBs)
 - Enterprise Java Beans (EJBs)
- Application Deployment
- JMS in WebSphere Message Broker

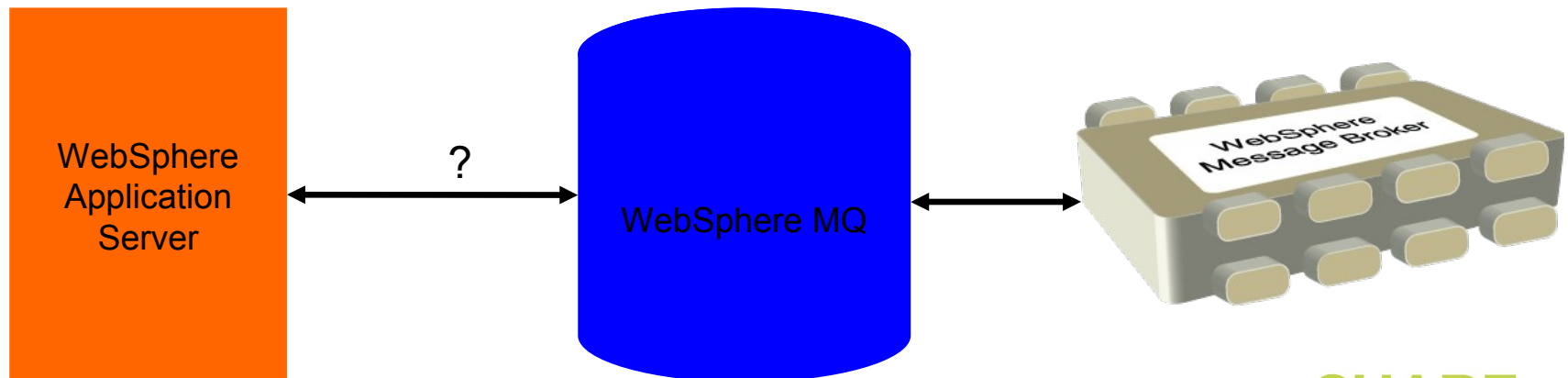
SHARE
in Anaheim
2011

The Scenario



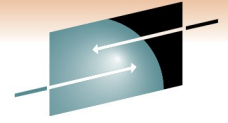
SHARE
Technology • Connections • Results

- Imagine your enterprise is an existing WebSphere MQ user or sends messages across other transports.
- Maybe WebSphere Message Broker is pulling in data from another source
- Another part of the organisation has started using WebSphere Application Server, and now wants to use the data that already exists in other applications.
- How do you do this?



SHARE
in Anaheim
2011

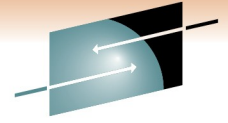
The Scenario



SHARE
Technology • Connections • Results

- WebSphere Application Server is a fully compliant Java Enterprise Edition (JEE) Application Server.
 - Provides integrated support to connect to Java Message Service (JMS) providers.
- WebSphere MQ is a fully compliant JMS provider.
- WebSphere Message Broker can route messages across any JMS 1.1 compliant provider
- Therefore, JMS is the answer!

Agenda



SHARE
Technology • Connections • Results

- The Scenario
- **Initial Setup**
 - **Queue Manager**
 - Connection Factories
 - Destinations
 - Activation Specifications
- Application Development
 - Message-driven beans (MDBs)
 - Enterprise Java Beans (EJBs)
- Application Deployment
- JMS in WebSphere Message Broker

SHARE
in Anaheim
2011

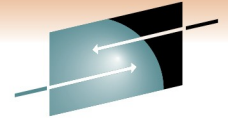
Initial Setup

Queue Manager



- The good news is that there isn't much setup required on the queue manager.
- If the application server is on a different machine to the queue manager, setup as for other client applications :
 - Start a listener
 - Create a SVRCONN channel.
- If the application server is on the same machine as WMQ, you don't need to do anything!

Agenda



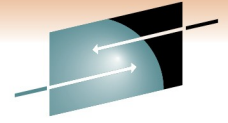
SHARE
Technology • Connections • Results

- The Scenario
- **Initial Setup**
 - Queue Manager
 - **Connection Factories**
 - Destinations
 - Activation Specifications
- Application Development
 - Message-driven beans (MDBs)
 - Enterprise Java Beans (EJBs)
- Application Deployment
- JMS in WebSphere Message Broker

SHARE
in Anaheim
2011

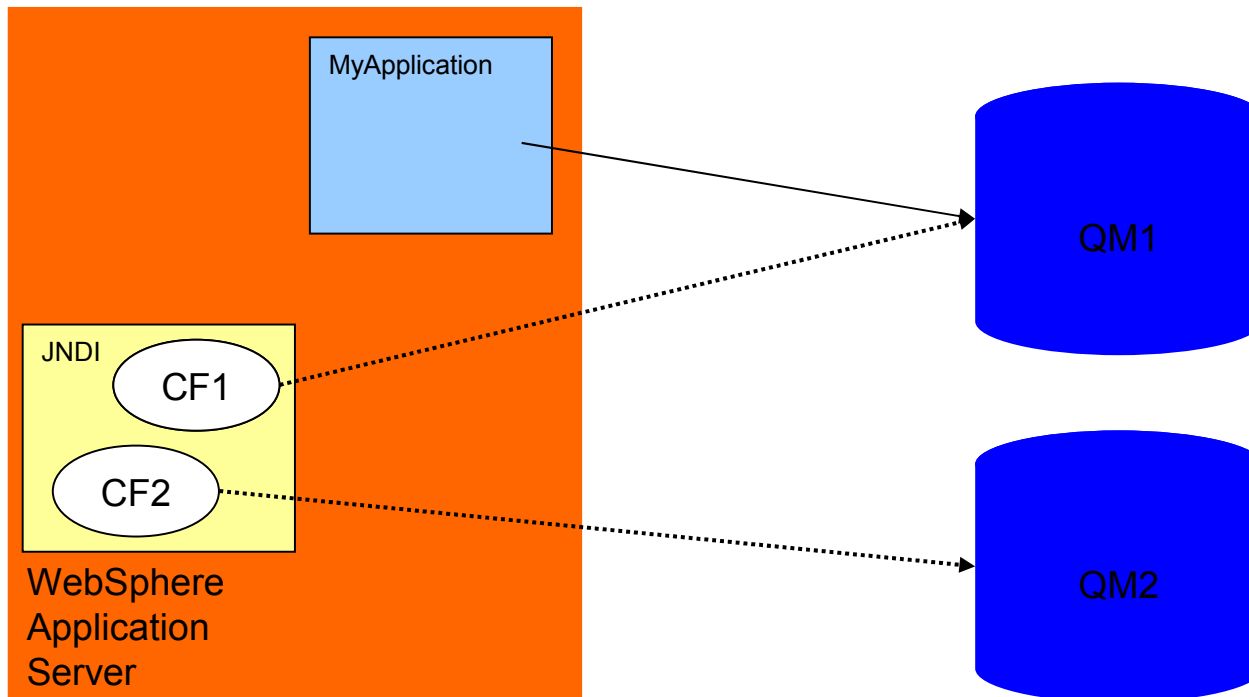
Initial Setup

Connection Factories



SHARE
Technology • Connections • Results

- Contains information about how to connect to a queue manager.

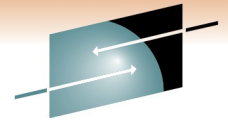


Initial Setup Connection Factories



- Need to be defined using WebSphere Administrative Console.
- A handy wizard helps through the creation process.
 - Also provides the ability to verify the Connection Factory has been defined correctly, by trying to connect to the specified queue manager.

Initial Setup Connection Factories



SHARE
Technology • Connections • Results

Create WebSphere MQ JMS resource

Create WebSphere MQ JMS resource

This wizard creates a WebSphere MQ JMS resource

→ **Step 1: Configure basic attributes**

Step 2: Select connection method

Step 3: Test connection

Step 4: Summary

Configure basic attributes

Configure the basic attributes to use for the new WebSphere MQ JMS resource

* Name

* JNDI name

Description

Initial Setup

Connection Factories



- WebSphere Application Server supports three types:
 - Queue Connection Factory
 - Used by applications that are going to be sending and receiving messages to and from queues (point-to-point messaging).
 - Topic Connection Factory
 - Used by publish/subscribe applications.
 - Unified Connection Factory
 - Can be used by either point-to-point or publish/subscribe applications.

Initial Setup

Connection Factories

- Important properties:
 - Queue Manager Name
 - Transport Type
 - BINDINGS
 - *Used when WebSphere Application Server is on the same machine as the queue manager.*
 - CLIENT
 - *Used when WebSphere Application Server is on a different machine to the queue manager.*
 - BINDINGS_THEN_CLIENT
 - *Special option useful when you are not sure if the application is running on the same machine as the queue manager or not.*
 - Hostname
 - Server Channel Name

Initial Setup

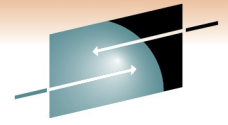
Connection Factories



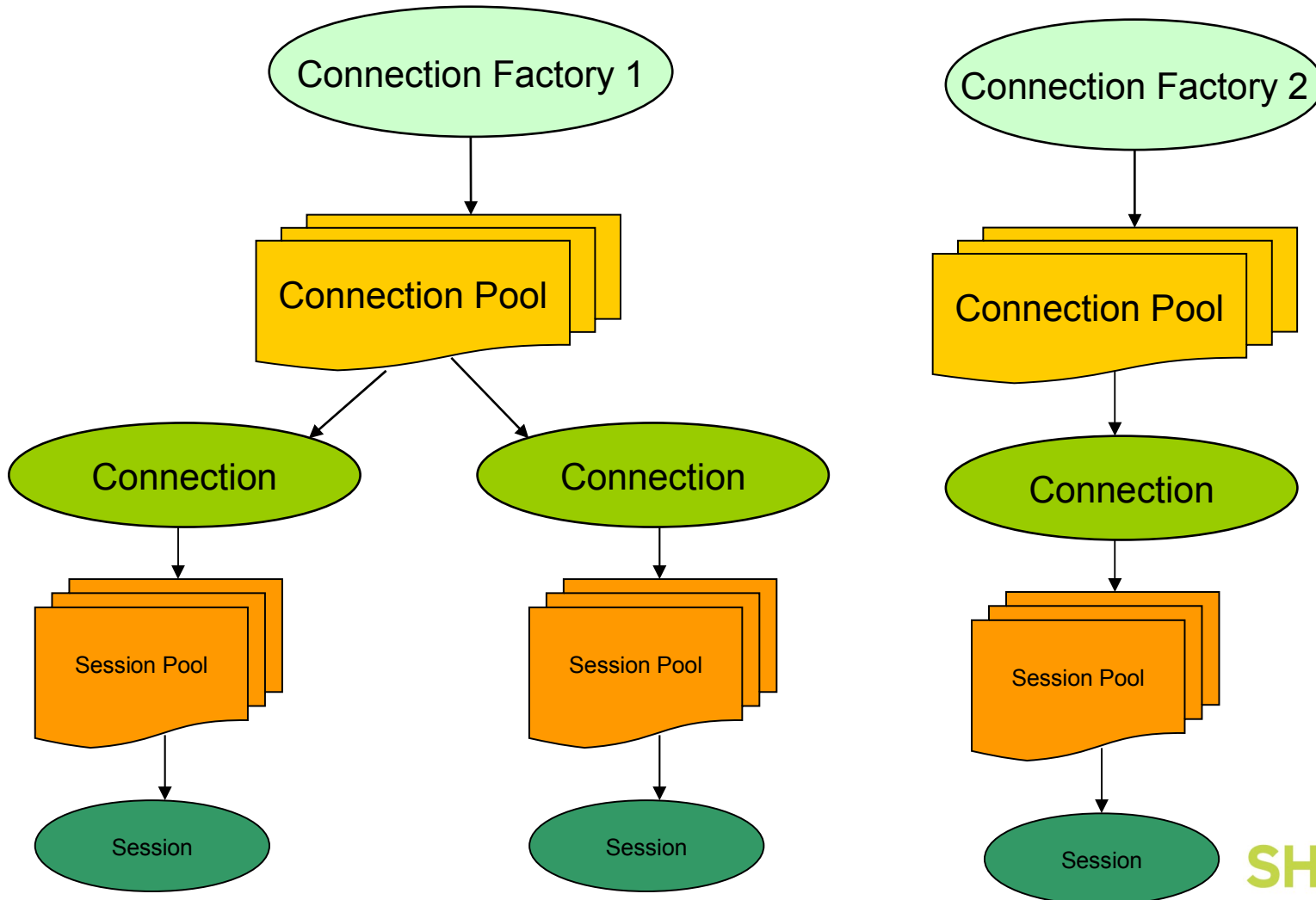
- WebSphere Application Server also provides Connection Pooling.
 - One Connection Pool per Connection Factory.
 - Used when an application creates a JMS Connection from a Connection Factory.
 - By default, only 10 connections can be created to a queue manager from a given Connection Factory.

Initial Setup

Connection Factories

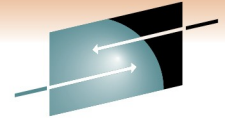


SHARE
Technology • Connections • Results



Initial Setup

Connection Factories



SHARE
Technology • Connections • Results

[Connection factories](#) > connectionFactory

A unified JMS connection factory can be used to create JMS connections to both

Configuration

General Properties

Administration

Scope

Node=L3D4112Node04,Server=server1

Provider

WebSphere MQ messaging provider

* Name

connectionFactory

* JNDI name

jms/connectionFactory

Description

Connection

Queue manager

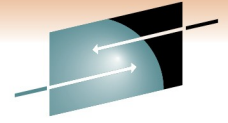
SHARE

Transport

Bindings, then client

SHARE
in Anaheim
2011

Agenda



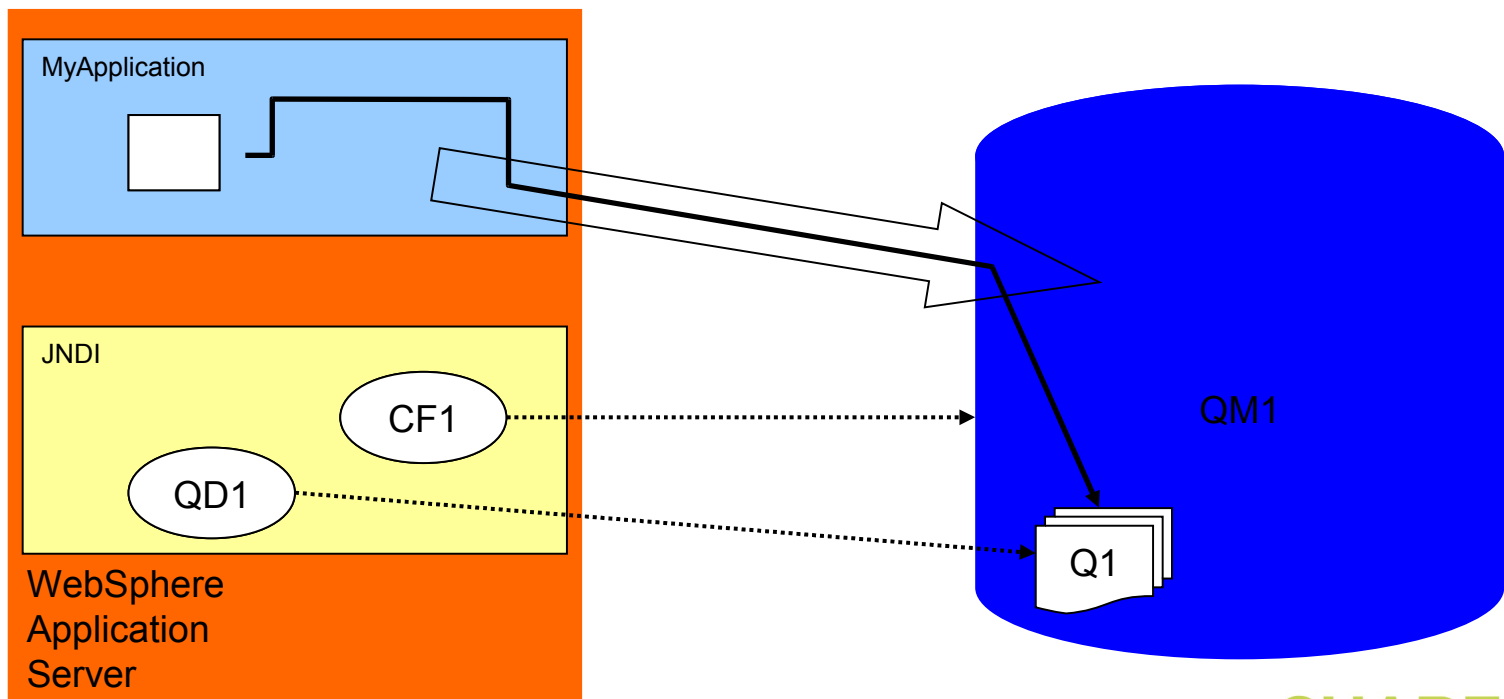
SHARE
Technology • Connections • Results

- The Scenario
- **Initial Setup**
 - Queue Manager
 - Connection Factories
 - **Destinations**
 - Activation Specifications
- Application Development
 - Message-driven beans (MDBs)
 - Enterprise Java Beans (EJBs)
- Application Deployment
- JMS in WebSphere Message Broker

SHARE
in Anaheim
2011

Initial Setup Destinations

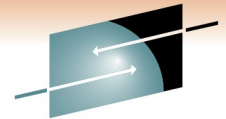
- JMS Destinations map to either queues or topics.



Initial Setup Destinations

- Need to be defined using WebSphere Administrative Console.
- Should have one Destination definition for every queue or topic used by applications running inside of the application server.
- Important properties:
 - Queue Name
 - Topic Name

Initial Setup Destinations



SHARE
Technology • Connections • Results

Queues ?

[Queues](#) > **queueDestination**

Queue destinations provided for point-to-point messaging by the WebSphere MQ messaging provider. Use WebSphere MQ queue destination administrative objects to manage queue destinations for the WebSphere MQ messaging provider.

Configuration

General Properties

Administration

Scope

Provider

* Name

* JNDI name

Description

WebSphere MQ Queue

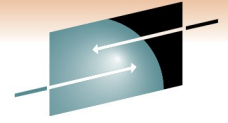
* Queue name

Additional Properties

- [Advanced properties](#)
- [WebSphere MQ Queue Connection Properties](#)
- [Custom properties](#)

SHARE
in Anaheim
2011

Agenda



SHARE
Technology • Connections • Results

- The Scenario
- **Initial Setup**
 - Queue Manager
 - Connection Factories
 - Destinations
 - **Activation Specifications**
- Application Development
 - Message-driven beans (MDBs)
 - Enterprise Java Beans (EJBs)
- Application Deployment
- JMS in WebSphere Message Broker

SHARE
in Anaheim
2011

Initial Setup

Activation Specifications



- Standard mechanism for listening for messages on JMS destinations.
- Contain information to create a connection to a specified queue or topic on a queue manager.
- Based on the J2EE Connector Architecture (JCA) 1.5 standard.
- Provides a common way for all JEE 1.4 compliant application servers to connect to JMS providers.

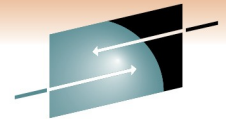
Initial Setup

Activation Specifications

- To create an Activation Specification.
 - Specify the JMS Destination to listen on.
 - Enter details of the queue manager where the Destination resides.
 - Optionally, specify a JMS Message Selector.
 - SQL expression.
 - Only messages that match the Selector will be delivered to applications using this Activation Specification.
 - A handy wizard takes you through all of the necessary steps, and checks it works too!.

Initial Setup

Activation Specifications



SHARE
Technology • Connections • Results

[Activation specifications](#) > **SHARE Activation Spec**

WebSphere MQ Activation Specification

Configuration

General Properties

Administration

Scope

Node=L3D4112Node04,Server=server1

Provider

WebSphere MQ Resource Adapter

* Name

SHARE Activation Spec

* JNDI name

jms/SHARESpec

Description

Connection

Queue manager

SHARE

Transport

Bindings, then client ▼

Additional Properties

- [Advanced properties](#)
- [Broker properties](#)
- [Custom properties](#)
- [Client transport properties](#)

Related Items

- [JAAS - J2C authentication data](#)

SHARE
in Anaheim
2011

Initial Setup

Activation Specifications

- By default, applications that use the Activation Specification will process 10 messages concurrently.
 - To change this, modify the Activation Specification Advanced Property Maximum server sessions.

Activation specifications

[Activation specifications](#) > [SHARE Activation Spec](#) > [Advanced properties](#)

These properties are used to configure advanced functionality when using a JMS connection factory to create connections to WebSphere MQ.

Configuration

General Properties

Message compression

Compress message headers

Compression algorithm for message payloads
NONE

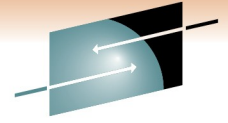
Connection consumer

Retain messages, even if no matching consumer is available

Rescan interval
5000 milliseconds

Maximum server sessions
10

Agenda

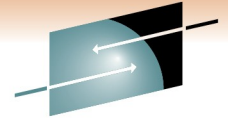


SHARE
Technology • Connections • Results

- The Scenario
- **Initial Setup**
 - Queue Manager
 - Connection Factories
 - Destinations
 - Activation Specifications
- **Application Development**
 - **Message-driven beans (MDBs)**
 - Enterprise Java Beans (EJBs)
- Application Deployment
- JMS in WebSphere Message Broker

SHARE
in Anaheim
2011

Application Development



SHARE

Technology • Connections • Results

- JEE applications are known as Enterprise Applications, and are stored in Enterprise Application Archives (EARs).
- Can consist of multiple parts:
 - Message-driven beans
 - Enterprise Java Beans
 - Servlets
 - Static web pages

Application Development

Message-driven beans



- Message-driven beans (MDBs) are JMS applications that get called when a message arrives on a given destination.
 - Similar to WMQ triggered applications.
- Recommended way of getting WMQ messages into WAS.
- Application developer only has to worry about the business logic required to process the message.
 - Application server handles the actual detection and delivery of the message.

Application Development

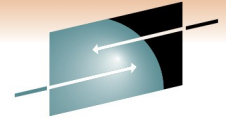
Message-driven beans



- MDBs must implement a method called `onMessage()`.
 - This is called when a message is detected on the specified destination.
 - Message is passed into the method.
 - `onMessage()` simply needs to contain the code to process it.
 - Application Server handles all transaction management.
- IBM Rational tooling provides wizards for creating MDBs.

Application Development

Message-driven beans



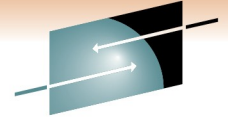
SHARE
Technology • Connections • Results

```
public void onMessage(Message message) {
    try
    {
        System.out.println("In onMessage()");

        if (message instanceof TextMessage)
        {
            TextMessage textMsg = (TextMessage)message;
            System.out.println("Message text is " + textMsg.getText());
        }
    }
    catch (JMSEException ex)
    {
        System.out.println("JMSEException occurred : " + ex);
    }
}
```

SHARE
in Anaheim
2011

Agenda



SHARE
Technology • Connections • Results

- The Scenario
- **Initial Setup**
 - Queue Manager
 - Connection Factories
 - Destinations
 - Activation Specifications
- **Application Development**
 - Message-driven beans (MDBs)
 - **Enterprise Java Beans (EJBs)**
- Application Deployment
- JMS in WebSphere Message Broker

SHARE
in Anaheim
2011

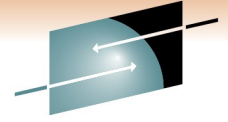
Application Development

Enterprise Java Beans



- Java applications that run inside of WAS.
- EJBs need to create their own connections to WMQ and get (or send) messages themselves.
 - EJBs use the standard JMS API.
 - Can be easier than MDBs when handling responses in request-reply messaging
 - Application server still handles transaction management, based on values specified in the application's deployment descriptor.

Application Development Enterprise Java Beans



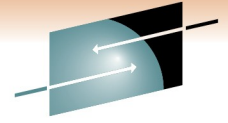
SHARE
Technology • Connections • Results

```
public void receiveMessage()
{
    try
    {
        InitialContext ctx = new InitialContext();
        ConnectionFactory cf = (ConnectionFactory) ctx.lookup("connectionFactory");
        Connection conn = cf.createConnection();
        conn.start();
        Session sess = conn.createSession(true, Session.AUTO_ACKNOWLEDGE);

        Queue d = (Queue) ctx.lookup("queueDestination");
        MessageConsumer consumer = sess.createConsumer(d);
        int timeout = 30000;
        Message msg = consumer.receive(timeout);
        if (msg instanceof TextMessage)
        {
            TextMessage textMessage = (TextMessage) msg;
            System.out.println("Message received:" + textMessage.getText());
        }
        consumer.close();
        sess.close();
        conn.close();
    }
    catch (Exception ex)
    {
        System.out.println("Error getting the connection factory");
        System.out.println("Exception : " + ex);
    }
}
```

SHARE
in Anaheim
2011

Agenda

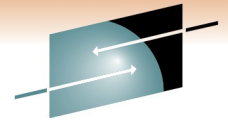


SHARE
Technology • Connections • Results

- The Scenario
- Initial Setup
 - Queue Manager
 - Connection Factories
 - Destinations
 - Activation Specifications
- Application Development
 - Message-driven beans (MDBs)
 - Enterprise Java Beans (EJBs)
- **Application Deployment**
- JMS in WebSphere Message Broker

SHARE
in Anaheim
2011

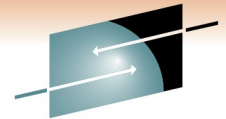
Application Deployment



SHARE
Technology • Connections • Results

- The final step is to deploy the application into the application server.
- As part of the message-driven bean deployment process, you need to tell the application which Activation Specification to use, which determines the queues and queue managers to monitor.

Application Deployment



SHARE

Technology • Connections • Results

Install New Application

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

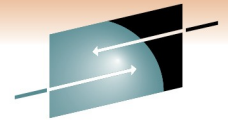
Bind listeners for message-driven beans

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

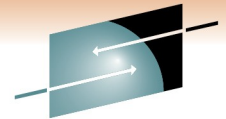
Select	EJB module	EJB	URI	Messaging type	Listener Bindings
<input type="checkbox"/>	SHARE-MDB	TestMDB	SHARE-MDB.jar, META-INF/ejb-jar.xml	javax.jms.MessageListener	<input type="radio"/> Listener port Name <input type="text"/> <input checked="" type="radio"/> Activation Specification Target Resource JNDI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

Application Deployment



SHARE
Technology • Connections • Results

- If the application has been deployed against an Activation Specification, simply start the application to start processing messages.
- If deployed against a Listener Port, a full application server restart might be required!
- And that's it!



SHARE
Technology • Connections • Results

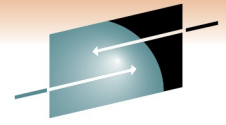
Getting WMQ messages into WebSphere Application Server (**including from WebSphere Message Broker**)

Simon Gormley – WebSphere MQ JMS Level 3 Service,
IBM Hursley, sgormley@uk.ibm.com

David Coles – WebSphere Message Broker Level 3 Service,
IBM Hursley – dcoles@uk.ibm.com

Thursday 3rd March 2011





JMS in Message Broker

- Support for any JMS 1.1 compliant provider not just MQ
- 6 JMS specific nodes for JMS
 - JMSInput
 - JMSOutput / JMSReply
 - JMSMQTransform
 - MQJMSTransform
 - JMSHeader
- Also support for JMS on SOAP nodes for SOAP over JMS
- Point to Point + Publish Subscribe



JMS Input



JMS Output



JMS Reply



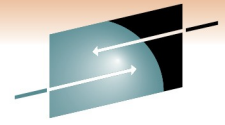
JMS Header



JMS MQ Transform

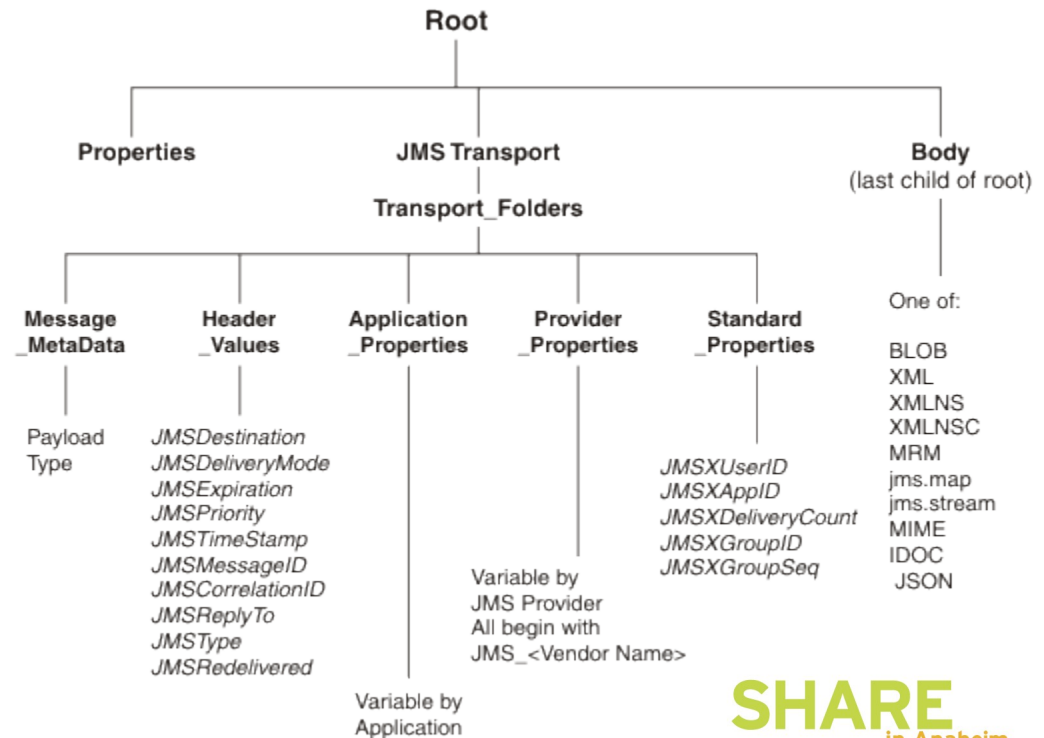


MQ JMS Transform

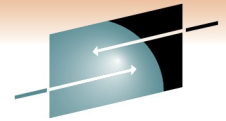


JMS Message Structure

- JMS Messages are not like bit stream wire format messages
- JMS Messages are Java objects
- Message Broker translates the Java object into our logical message tree
- The payload is stored in the body in the same way as with wire format messages



Configuring JMS Input Node



SHARE
Technology • Connections • Results



JMS Input

- Point 2 Point or Publish Subscribe message consumer
- Responsible for creating JMS tree from JMS input message
- Hands payload to appropriate broker parser
- Connection Properties
 - Initial Context factory
 - Location JNDI bindings
 - Points to JNDI administered objects
 - LDAP required userName and password
 - ConnectionFactory identified
- Backout Destination & Threshold – must be configured
- Message selectors available if filtering required

Properties Problems Deployment Log Console

JMS Input Node Properties - JMS Input

Description

Basic

Source queue

JMS Connection

Subscription topic

Input Message Parsing

Parser Options

Message Selectors

Advanced

Validation

Monitoring

Durable subscription ID

Properties Problems Deployment Log Console

JMS Input Node Properties - JMS Input

Description

Basic

JMS provider name* WebSphere MQ

JMS Connection

Initial context factory* com.sun.jndi.fscontext.ReffFSContextFactory

Input Message Parsing

Location JNDI bindings* c:\myBindings

Parser Options

Connection factory name* queueFac1

Message Selectors

Advanced

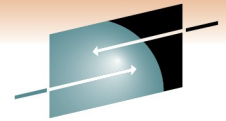
Backout destination

Validation

Backout threshold 0

Monitoring

Configuring the JMS Output/Reply Node



SHARE
Technology • Connections • Results



JMS Output



JMS Reply

- Point to Point or Publish Subscribe message producer
- Responsible for creating JMS output message from JMS tree
- JMS Connection values similar to JMSInput
 - No backout details
 - If you have a JMSInput node in the same the JMSOutput/Reply node does not have to use the same provider
- JMSReply node just sends its message to the destination in the JMSReplyTo header
- JMSOutput node can either send to a queue or a publish to a topic
 - Reply to destination settable for request messages

Properties Problems Deployment Log Console

JMS Output Node Properties - JMS Output

Description

Basic

Destination queue

JMS Connection

Publication topic

Reply to destination

e.g: jndi.MyQueue or jndi.MyTopic

Send to destination list in local environment

Advanced

Validation

Monitoring

Properties Problems Deployment Log Console

JMS Output Node Properties - JMS Output

Description

Basic

New correlation ID

JMS Connection

Transaction mode None

Advanced

Delivery mode Automatic

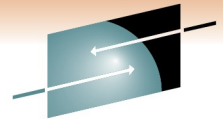
Validation

Message expiration (ms) 0

Message priority 4

Monitoring

Message type BytesMessage



SHARE

Technology • Connections • Results



JMS Header

JMSHeader node

- Code-free way to create/update/delete JMS headers
- Subset of common values can be changed
- Application properties can be added or deleted
 - Static or dynamic values using XPath

Properties Problems Deployment Log Console

JMS Header Node Properties - JMS Header

Description

JMS Transport JMS Header Options Carry forward header Add header Modify header Delete header

Header Values

Application Properties

Properties Problems Deployment Log Console

JMS Header Node Properties - JMS Header

Description

JMS Transport JMS Delivery Mode

Header Values JMS Message Expiration(ms)

Application Properties JMS Message Priority

Monitoring JMS Correlation Identifier

JMS Reply To

Properties Problems Deployment Log Console

JMS Header Node Properties - JMS Header

Clear incoming values

Application Properties

Name	Type	Value
SaleType	XPath	\$Root/Properties/Topic

Configuration steps for any JMS provider



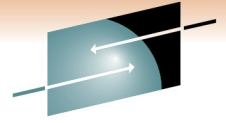
- Ensure JMS Provider jars available to the broker
 - Either place them in the shared-classes directory under the broker's workpath
 - Linux/Unix default: /var/mqsi/shared-classes
 - Windows default: C:\Documents and Settings\All Users\Application Data\IBM\MQSI\shared-classes
 - z/OS: <broker component directory>/shared-classes
 - Or use the jarsURL property on the JMSPROvider configurable service
- Add any JMS Provider native libraries to Broker's LIBPATH
- You can secure your JMS connections and JNDI lookups
 - JMS connection:
 - `mqsisetdbparms <broker Name> -n jms::<Connection Factory Name> -u myuserid -p secret`
 - JNDI lookup
 - `mqsisetdbparms <broker Name> -n jndi::<Initial Context Factory> -u myuserid -p secret`
- Use JMSPROvider configurable service to override the JMS connection properties on the node
 - Also useful to enable provider specific JMS options as not all JMS 1.1 compliant supplier implement the spec in the same way
 - JBOSS uses asynchronous exception handling
 - BEAWeblogic requires extra parameters for XA

WMB JMS Extras



- SOAP over JMS
 - SOAP over Java Message Service 1.0 is a specification that describes how SOAP can bind to a messaging system that supports the Java Message Service
 - Use WSDL definitions with JMS Bindings

```
<soap:address location="jms:jndi:REPLYTOQ2?
jndiConnectionFactoryName=QCF&
jndiInitialContextFactory=com.sun.jndi.fscontext.RefFSContextFactory&
jndiURL=file:/C:/mqsi6/webservices/SOAP/JMS/JNDI&
targetService=SOAPJMSGenMessageSetSOAP_JMS_Service&
timeToLive=30000"/>
```
 - Supported across all WMB SOAP Nodes
- Coordinated transaction support
 - XA on distributed
 - Only if the JMS Provider supports the XA/Open interface
 - RRS on z/OS when using the MQ Java Client and connecting in Bindings mode

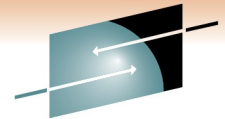


SHARE
Technology • Connections • Results

Any questions?

- If you have any questions, or ideas for future topics, feel free to email us at sgormley@uk.ibm.com or dcoles@uk.ibm.com

SHARE
in Anaheim
2011



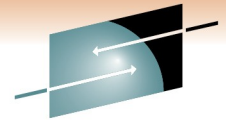
Demo Slides

- With an existing MQ Queue manager, and WebSphere Application Server instance running, we start by defining a JMS queue resource via the WebSphere Administration Console

The screenshot displays the WebSphere Administration Console interface. The left-hand navigation pane shows a tree view of resources, with 'JMS' expanded to show 'Queues'. The main content area is titled 'Queues' and contains the following information:

- Queues**: A JMS queue is used as a destination for point-to-point messaging.
- Scope**: Cell=L3D4112Node04Cell, Node=L3D4112Node04, Server=server1
- Scope description**: 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.](#)
- Scope selector**: A dropdown menu showing 'Node=L3D4112Node04, Server=server1'.
- Preferences**: A section with 'New' and 'Delete' buttons, and a table of queue resources.

Select	Name	JNDI name	Provider	Description	Scope
None					
Total 0					



Demo Slides

- Click on New, then select the WMQ JMS provider, and click OK
- Fill in the destination details, and click OK.

Queues

Queues > Select JMS resource provider

Scope:

Select the provider with which to create the Queue. The following providers support the selected resource type and are available at the selected scope.

Default messaging provider
 V5 default messaging provider
 WebSphere MQ messaging provider

Queues > WebSphere MQ messaging provider > New

Queue destinations provided for point-to-point messaging by the WebSphere MQ messaging provider. Use WebSphere MQ queue destination administrative objects to manage queue destinations for the WebSphere MQ messaging provider.

Configuration

General Properties

Administration

Scope:

Provider:

* Name:

* JNDI name:

Description:

WebSphere MQ Queue

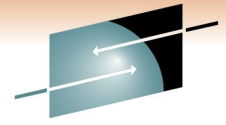
* Queue name:

Queue manager or Queue sharing group name:

The additional properties will not be available until the general properties for this item are applied or saved.

Additional Properties

- Advanced properties
- WebSphere MQ Queue Connection Properties
- Custom properties



Demo Slides

- Create a new WMQ JMS Activation Specification. Click on JMS, Activation Specifications, then New, and select the WMQ JMS Provider. Fill in the name details, and click Next.
- Select the JMS destination to use, for example, the destination just created. The message selector (filter) can also be specified here.

This wizard creates a WebSphere MQ JMS resource

→ Step 1: Configure basic attributes

Step 2: Select connection method

Step 3: Test connection

Step 4: Summary

Configure basic attributes

Configure the basic attributes to use for the new WebSphere MQ JMS resource

* Name: SHARE AS

* JNDI name: jms/SHAREAS

Description:

Next Cancel

Specify MDB destination data

Specify MDB destination data

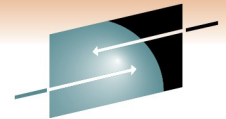
Enter information about the destination from which messages will be delivered to Message Driven Beans that are associated with the new activation specification.

* Destination JNDI name: jms/SHAREAS

Message selector:

Destination type: Queue

Previous Next Cancel



Demo Slides

- Select the connection method, for example, entering all the information to the wizard.

This wizard creates a WebSphere MQ JMS resource

Step 1: Configure basic attributes
→ Step 2: Select connection method
Step 3: Test connection
Step 4: Summary

Select connection method

Decide what information to enter to determine how to connect to WebSphere MQ

Enter all the required information into this wizard
 Use a client channel definition table

Previous Next Cancel

- Enter the Queue Manager name

Custom WebSphere MQ connection

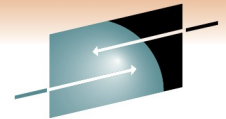
Step 1: Configure basic attributes
Step 2: Select connection method
→ Step 2.1: Supply queue connection details
Step 2.2: Enter connection details
Step 3: Test connection
Step 4: Summary

Supply queue connection details

Enter details about the queue manager or queue sharing group that you wish to connect to.

Queue manager or queue sharing group name
SHARE

Previous Next Cancel

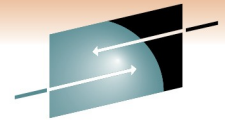


Demo Slides

- Enter the connection details
- Test the connection, which should return successfully!
- Click Next, Finish and then Save to store the information

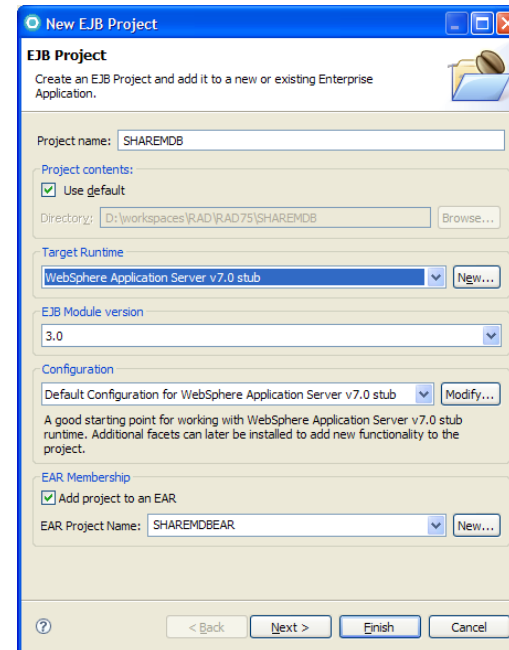
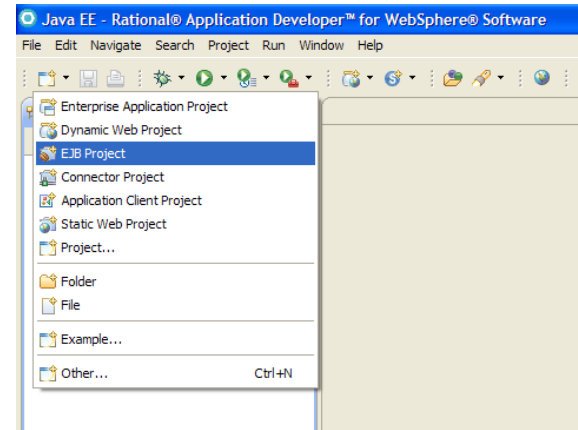
The screenshot shows a dialog box titled "Custom WebSphere MQ connection". On the left, a vertical pane lists four steps: "Step 1: Configure basic attributes", "Step 2: Select connection method", "Step 2.1: Supply queue connection details", and "Step 2.2: Enter connection details" (which is highlighted with a yellow arrow). The main area is titled "Enter connection details" and contains the following fields: "Transport" (a dropdown menu set to "Bindings, then client"), "Hostname" (a text box containing "localhost"), "Port" (a text box containing "1414"), and "Server connection channel" (a text box containing "SYSTEM.DEF.SVR.CONN"). At the bottom, there are "Previous", "Next", and "Cancel" buttons.

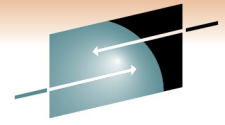
The screenshot shows a dialog box titled "Create WebSphere MQ JMS resource". On the left, a vertical pane lists four steps: "Step 1: Configure basic attributes", "Step 2: Select connection method", "Step 2.1: Supply queue connection details", and "Step 2.2: Enter connection details". "Step 3: Test connection" is highlighted with a yellow arrow. The main area is titled "Test connection" and contains the text: "To test establishing a connection using the information provided select the 'Test Connection' button. It may take several seconds to perform this test. If you wish to skip this test, select the 'Next' button." Below the text is a "Test connection" button. At the bottom, there are "Previous", "Next", and "Cancel" buttons.



Demo Slides

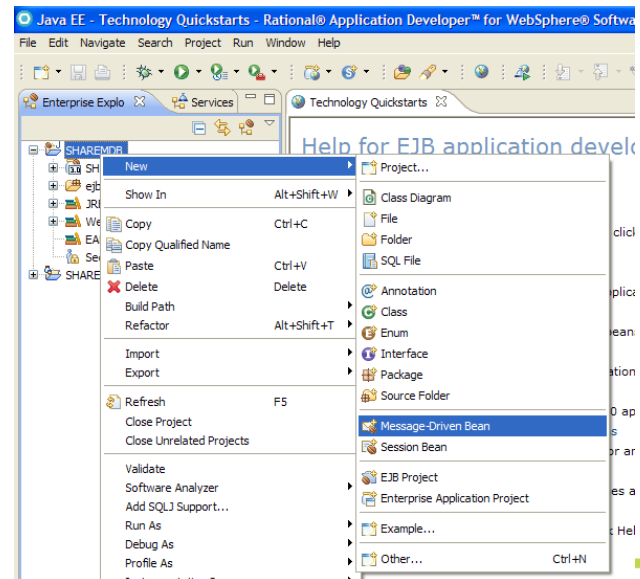
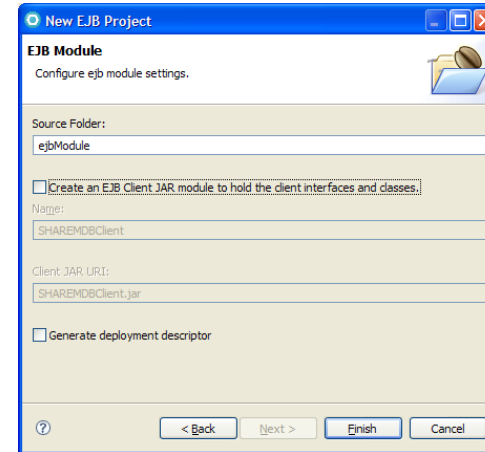
- In Rational Application Developer, create a new EJB Project
- Specify a project name, the target application server, and ensure a EAR project is also specified/created.

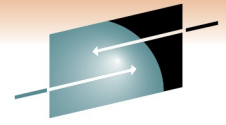




Demo Slides

- There's no need to create a EJB Client. Click Finish.
- Right click on the EJB project, and create a new Message-Driven Bean.





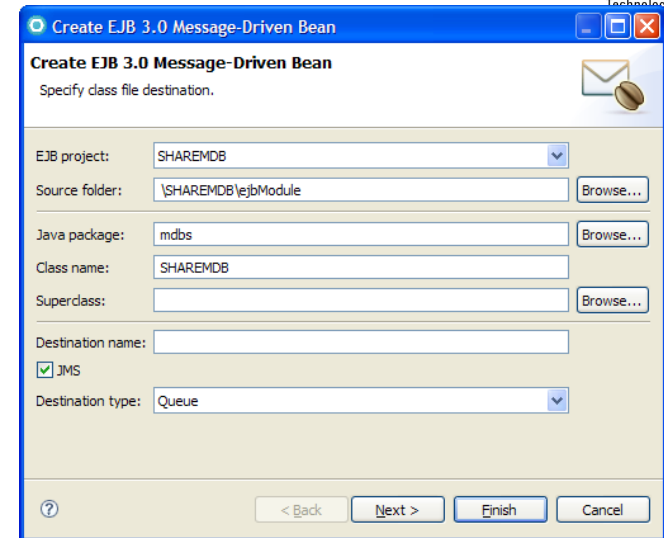
SHARE

Technology • Connections • Results

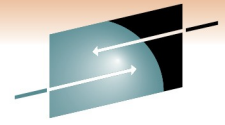
Demo Slides

- Enter the MDB Java package name and class name to be created. Click Finish.
- In the MDB, enter the business logic code into the onMessage method. (This code would require the following Java import statements:

```
import javax.jms.JMSEException;  
import javax.jms.TextMessage;
```

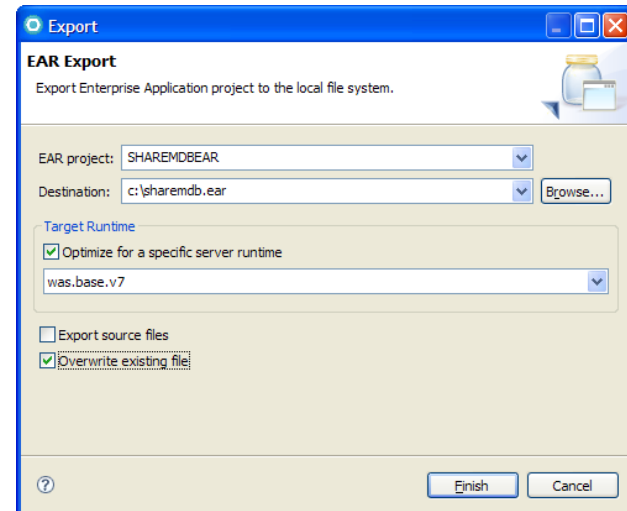
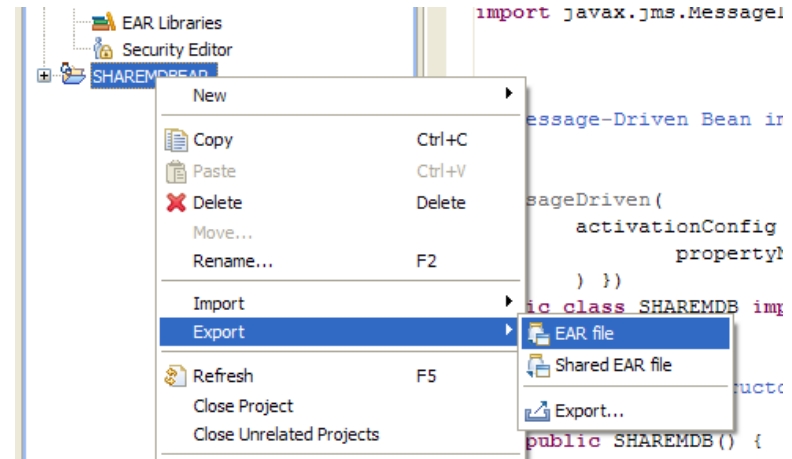


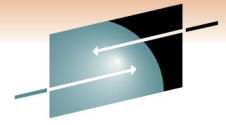
```
public void onMessage(Message message) {  
    if (message instanceof TextMessage) {  
        try {  
            System.out.println("Received text message: "  
                + ((TextMessage) message).getText());  
        } catch (JMSEException e) {  
            e.printStackTrace();  
        }  
    } else {  
        System.out.println("Received message: " + message);  
    }  
}
```



Demo Slides

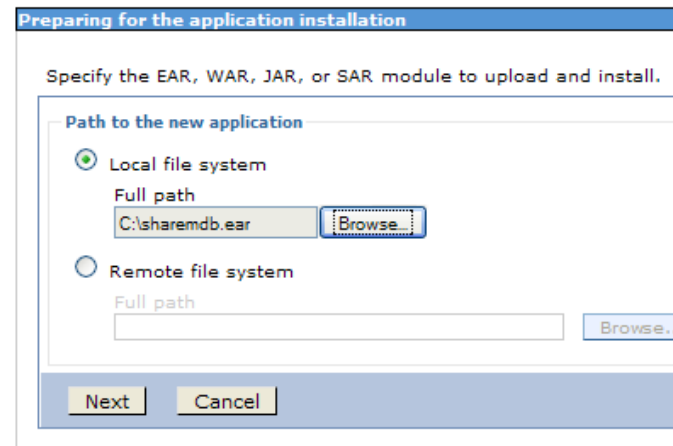
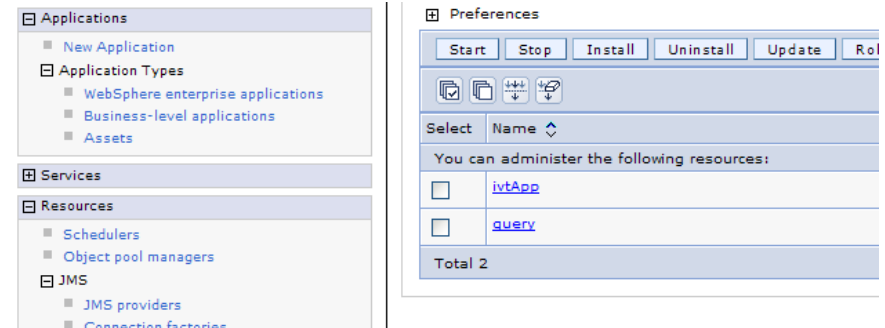
- Right click on the EAR project, and export the EAR file, ensuring the file ends with “.ear”.

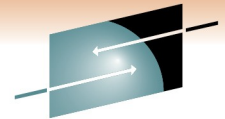




Demo Slides

- In the WAS Administration Console, click Applications, Application Types, WebSphere enterprise applications, and then the “Install” button
- Locate the newly created .ear file, and click Next. Then choose “Fast Path”





Demo Slides

- It is possible to skip over Steps 1 and 2, and click on Step 3.
- Enter the Activation Specification created earlier, and click Next, then Finish.
- Then save the application to the master configuration.

Install New Application

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

Precompile JavaServer Pages files

Directory to install application

Distribute application

Use Binary Configuration

Deploy enterprise beans

Install New Application

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

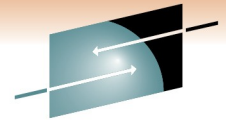
Bind listeners for message-driven beans

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 type="checkbox"/>	SHAREMDB.jar	SHAREMDB	SHAREMDB.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 jms/SHAREAS Destination JNDI name <input type="text"/> ActivationSpec authentication alias <input type="text"/>

Previous Next Cancel



Demo Slides

- The application is now deployed! Select it, and click the Start button to get it running.
- You can now put a test message to the queue, and for this application, see the result in the Application Server SystemOut.log.

The screenshot shows the 'Enterprise Applications' management page. At the top, there are buttons for 'Start', 'Stop', 'Install', 'Uninstall', 'Update', 'Rollout Update', and 'Remove File'. Below these are icons for selection and actions. A table lists three applications: 'ivtApp', 'query', and 'sharemdb'. The 'sharemdb' application is selected and has a red 'X' icon in the status column, while the others have green right-pointing arrows. A 'Total 3' summary is shown at the bottom of the table.

Select	Name	Application Status
<input type="checkbox"/>	ivtApp	➔
<input type="checkbox"/>	query	➔
<input checked="" type="checkbox"/>	sharemdb	✖

Total 3