IBM Software Group

IBM WebSphere Application Server v6

WebSphere Platform Messaging

JMS and SIB Resource Management

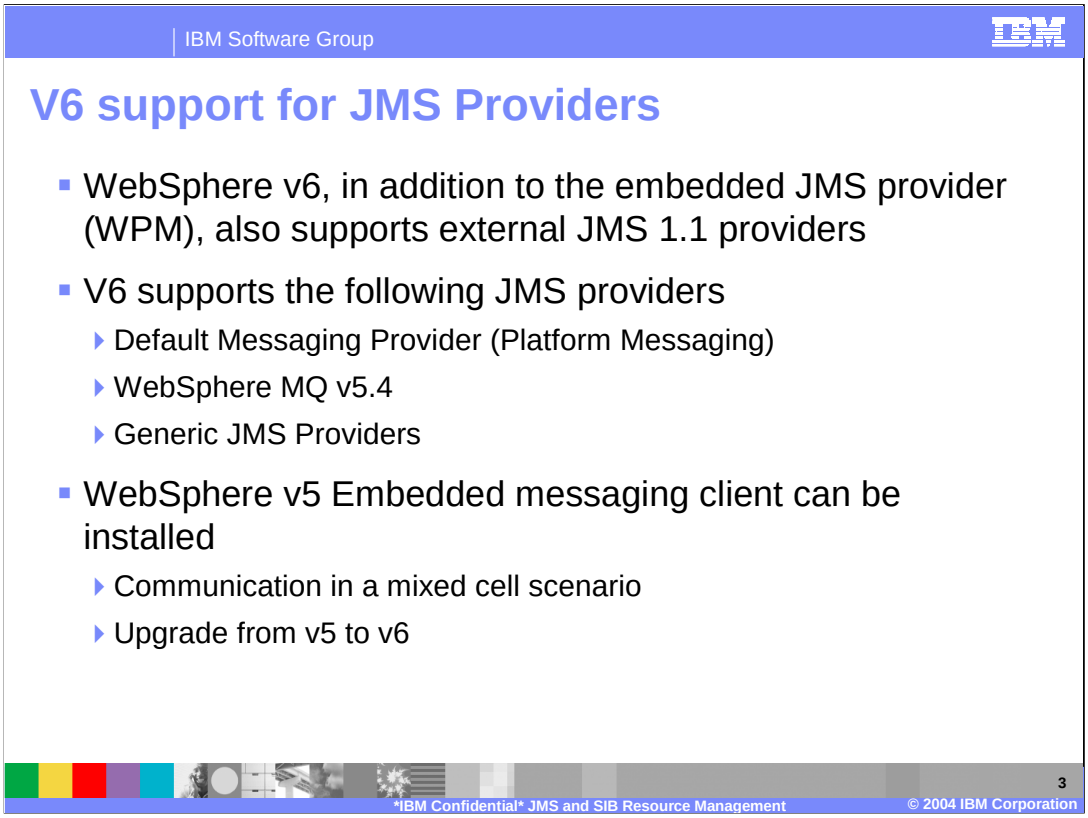
e business on demand.

IBM Confidential

Updated October 12, 2004

Agenda

- Provide an Overview of creating SIB resources using Admin Clients
- Provide an Overview of creating JMS resources using Admin Clients
- Describe WebSphere v6 support of external JMS providers



You can install and use one or more of the following messaging providers:

The default messaging provider. This is installed as part of WebSphere Application Server, administered through the administrative console, and managed as part of the WebSphere Application Server runtime.

WebSphere MQ version 5.4 or later.

Other "generic" messaging providers.

You can also choose to install the embedded messaging client of WebSphere Application Server version 5, which enables JMS 1.0.2 inter-operation with WebSphere Application Server version 5. This also enables you to configure JMS resources for use with WebSphere Application Server version 5.

When upgrading from v5 to v6, the JMS server is uninstalled and an ME is installed.

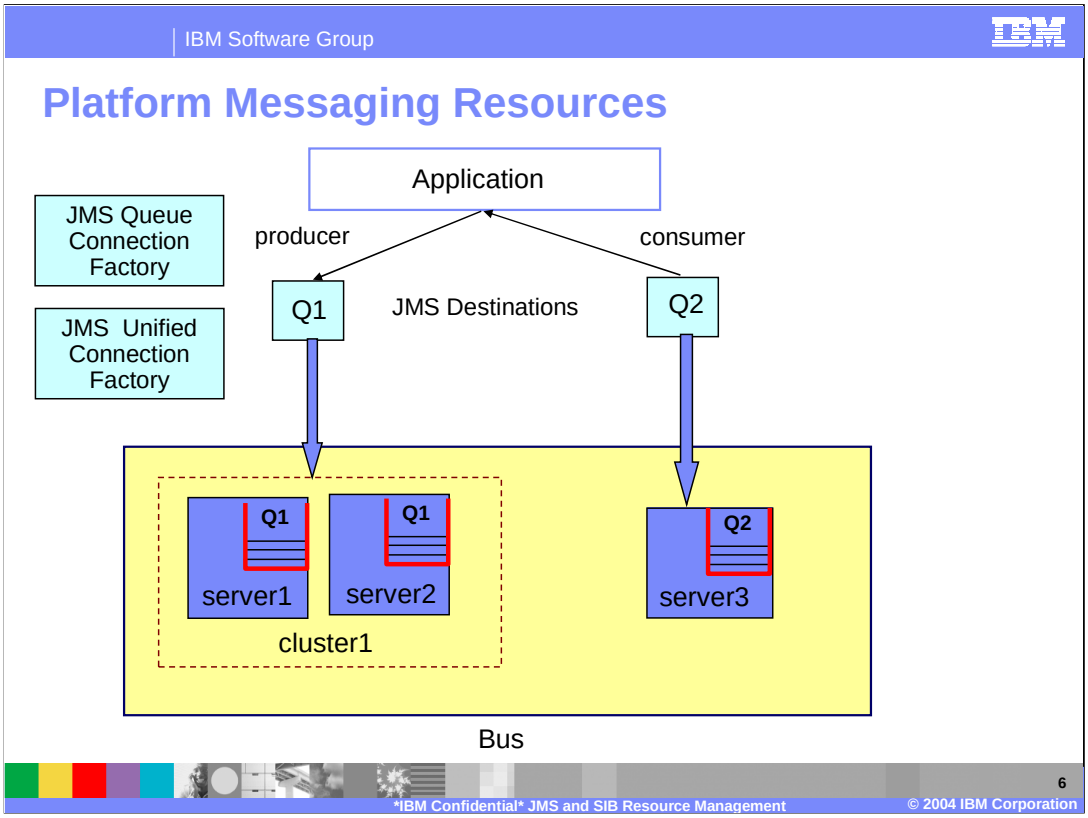
Section

WebSphere Platform Messaging

JMS Administration Overview

Platform Messaging: High level Administration

- Administrative unit is the cell
- Bus level
  - Manage SIBus resources - bus members, messaging engines, destinations, and association with Mediation
- Infrastructure management
  - Define and deploy messaging engines to processes
  - Associate WPM destinations with messaging engines
  - Assign persistent stores to messaging engines
  - Define links – to connect to other SIBus, and to WebSphere MQ
    - Note – no need to define links between messaging engines within a bus



An application that uses point-to-point messaging acts as a producer or consumer of messages with JMS queues.

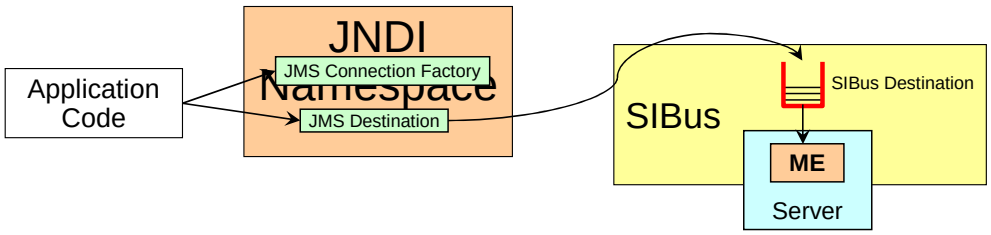
An administrator can define a *JMS queue*, an administrative object that encapsulates the queue name and other configuration properties that the administrator wants to preserve.

The administrator configures the queue onto a *destination* on a *service integration bus*. Such a queue is available, over a long period of time, to all applications with access to the destination. The destination is provided by only one member (an application server or cluster) of the service integration bus. (A destination for a queue is *localized to* either an application server or cluster.) A queue that is localized to a cluster is partitioned across the servers in that cluster.

A *JMS connection factory* creates connections to the messaging engine that localizes the destination. The preferred way for an administrator to define a JMS connection factory for queues is to define a *unified* JMS connection factory. Such unified JMS connection factories support both queues and topics, which enables applications to use the same, common, connection factories. As an alternative to defining unified JMS connection factories, an administrator can define domain-specific *JMS queue connection factories*, as used for administration before JMS 1.1.

Using SIB and JMS Destinations

- To use a JMS Queue or Topic, you need 2 different layers of administratively created objects
  - SIBus Destination – SIBus Queue / SIBus TopicSpace
    - This destination is on the bus and is invisible to the 'user'
  - JMSDestination – JMS Queue/JMS Topic
    - This destination acts as a proxy/pointer to the SIBDestination



In order to use a JMS Queue or Topic you need 2 different layers of administratively created objects.

One of the layers is the **SIBDestination**. This Destination is invisible to the 'user' who should only see the **JMS Destination** (which basically acts as a proxy/pointer to the SIBDestination). Users will also need a **JMS Connection Factory**.

NOTE : The JMS Destinations and Connection Factories documented here are specific to the SIBus implementation of JMS and as such are referred to as SIBJMSDestinations and SIBJMSConnectionFactories in the commands that create them. Other JMS Providers (such as MQ) would have their own ConnectionFactories and Destinations.

So – as with all JMS Applications the System Administrator binds JMS Destinations and JMS Connection Factories into JNDI and the user code does lookups on them and uses them to connect to the underlying JMS Provider through the standard JMS Interface.

Deeper explanation of destinations

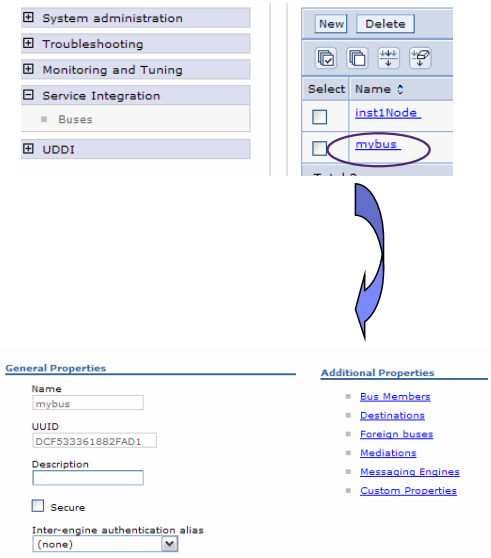
- Point-to-Point messaging
  - Create SIBus Queue
  - Create matching JMS Queue
- Pub-Sub Messaging
  - Create SIBus Topic Space
  - Create JMS topic
    - When a JMS message is sent to a Topic, the SIBus treats it as a message sent to the associated TopicSpace
    - The actual Topic underneath it is a sort of Message Selector on Topic Name

Section

Managing SIB resources

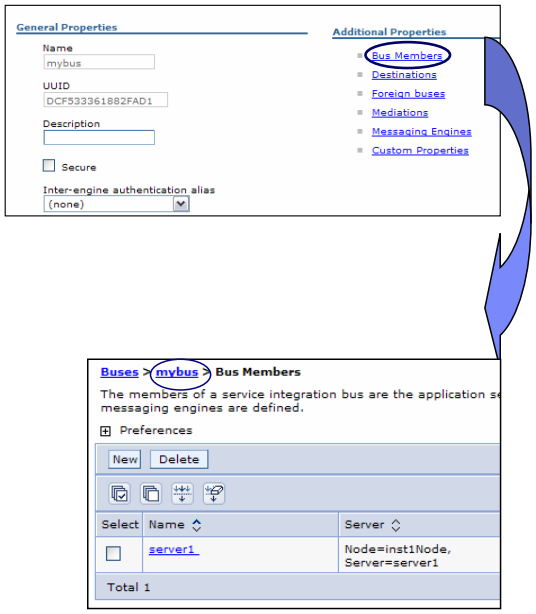
Configuring Buses

- Click Service Integration > Buses
- You can add or delete buses, or select a bus to display or configure its properties
  - A default bus (named as NodeName) is created at install time for samples



Configuring Bus Members

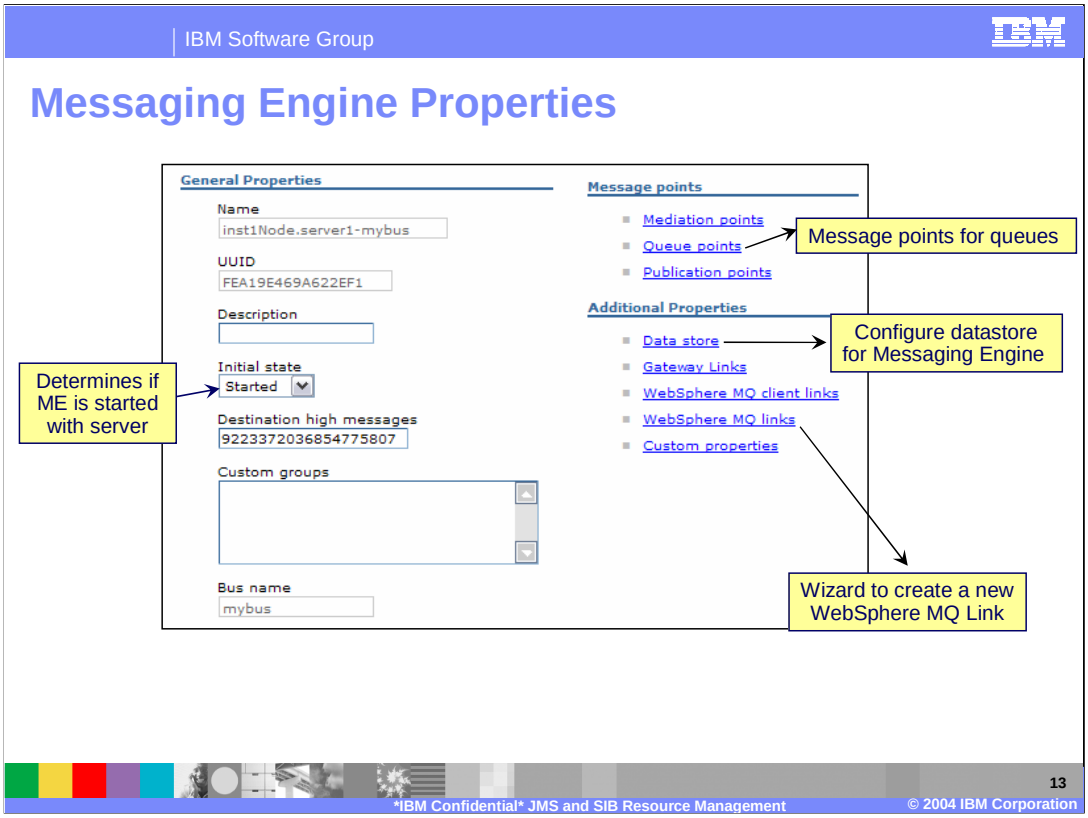
- Under Additional Properties for a bus, click Bus Members
- Bus member is added to the bus and a default messaging engine is created for that member



Configuring Messaging Engines

- Listing Messaging Engines
  - Click the name of the bus
  - Under Additional Properties, Click Messaging Engines

- Creating a new Messaging Engine in a SIBus
  - New MEs can be created in clusters for scalability reasons
- Removing a Messaging Engine
  - Before deleting, stop the Messaging Engine.
  - Also, change localization of destinations to another messaging engine



Gateway Links

Links between a messaging engine and another messaging engine in a foreign bus, to enable communication between the two buses.

WebSphere MQ links

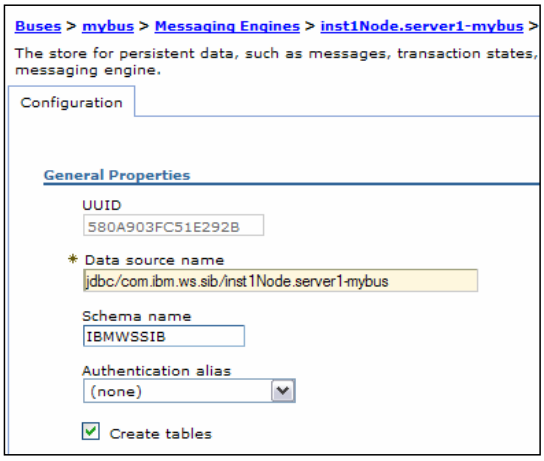
Links between the messaging engine and WebSphere MQ networks. Each WebSphere MQ link connects the messaging engine as a queue manager to WebSphere MQ, thereby providing a bridge between the bus and a WebSphere MQ network.

WebSphere MQ client links

Links between the messaging engine and WebSphere MQ clients. Each WebSphere MQ client link presents the messaging engine, and thereby the bus, as a WebSphere MQ queue manager to which WebSphere MQ clients can attach. This enables WebSphere Application Server version 5 JMS clients to use messaging resources on the bus.

Setting up the Messaging Store for a ME

- Create the database
- Create the tables
  - Administrator can create the tables
    - Use the SIBDDLGenerator command to get the DDL
    - sibDDLGenerator -system db2 -version 8.1 -schema SIB -user test > DDLforSIB.txt
  - Tables can be created when the Message Store is configured
- Create the datasource
- Configure the Message Store
  - Buses -> BusName -> Messaging Engines -> ME Name -> Data Store

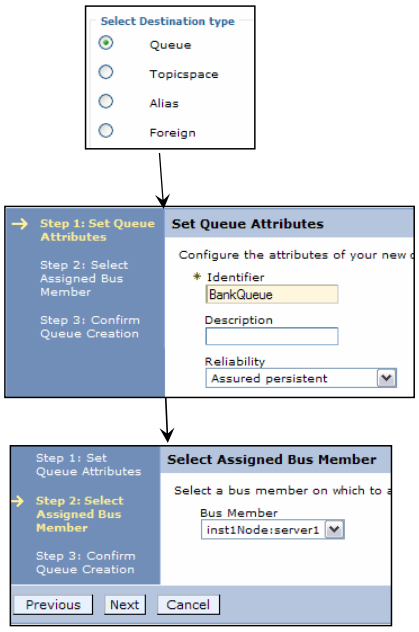


A messaging engine needs DBMS resources, such as database tables, which it can create when starting. If your installation has a policy that only a database administrator has the authority to create database tables, use the sibDDLGenerator command to enable your database administrator to create the DBMS resources that the messaging engine needs. The sibDDLGenerator command generates the DDL statements that your database administrator can save and then process to create the required DBMS resources

Creating Bus Destinations

- Click Service integration > Buses
- Under Additional Properties of a bus, click Destinations. Click New.
  - Wizard to create SIBus Destinations
  - Select Queue or TopicSpace

A Queue Point will be created on the ME on "server1" for Queue "BankQueue"

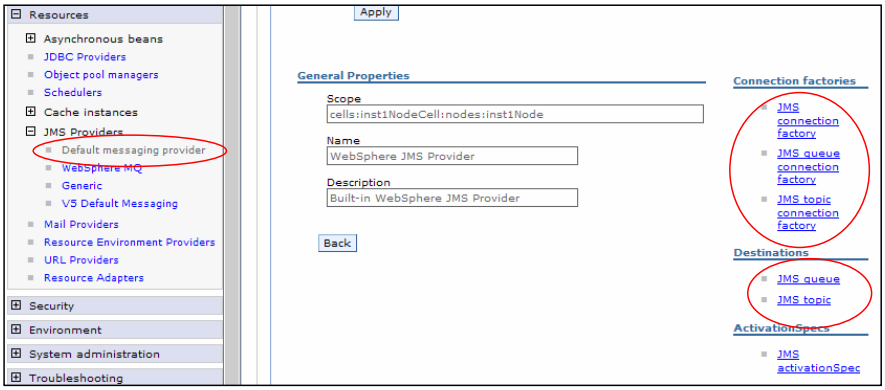


Section

Managing JMS resources

Configuring JMS resources

- Expand Resources > JMS Providers > Default Messaging Provider
- Under Additional Resources, select the link for the type of JMS resource

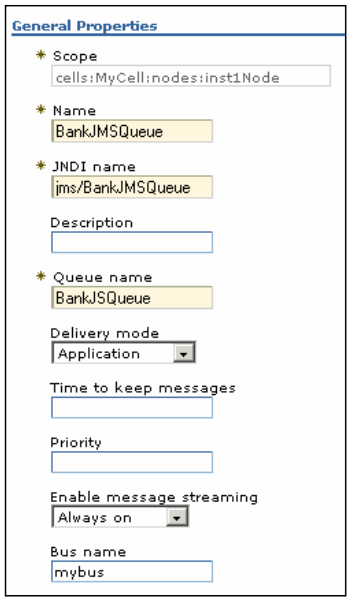


Creating JMS Connection Factory

- JMS Connection factory is used to obtain connections for both point-to-point and pub-sub messaging styles
  - This JMS Unified Connection factory is new in JMS 1.1
  - Enter Name, JNDI Name, Message Reliability, Bus Name Remote target type etc
- Can also create queue connection factory and topic connection factory separately

Creating JMS Queue

- JMS queue is the JMS destination that applications interact with
  - It is the application view of the SIB destination
- Admin configures it as a JMS Queue resource
  - Name of the SIB queue is entered as part of configuration



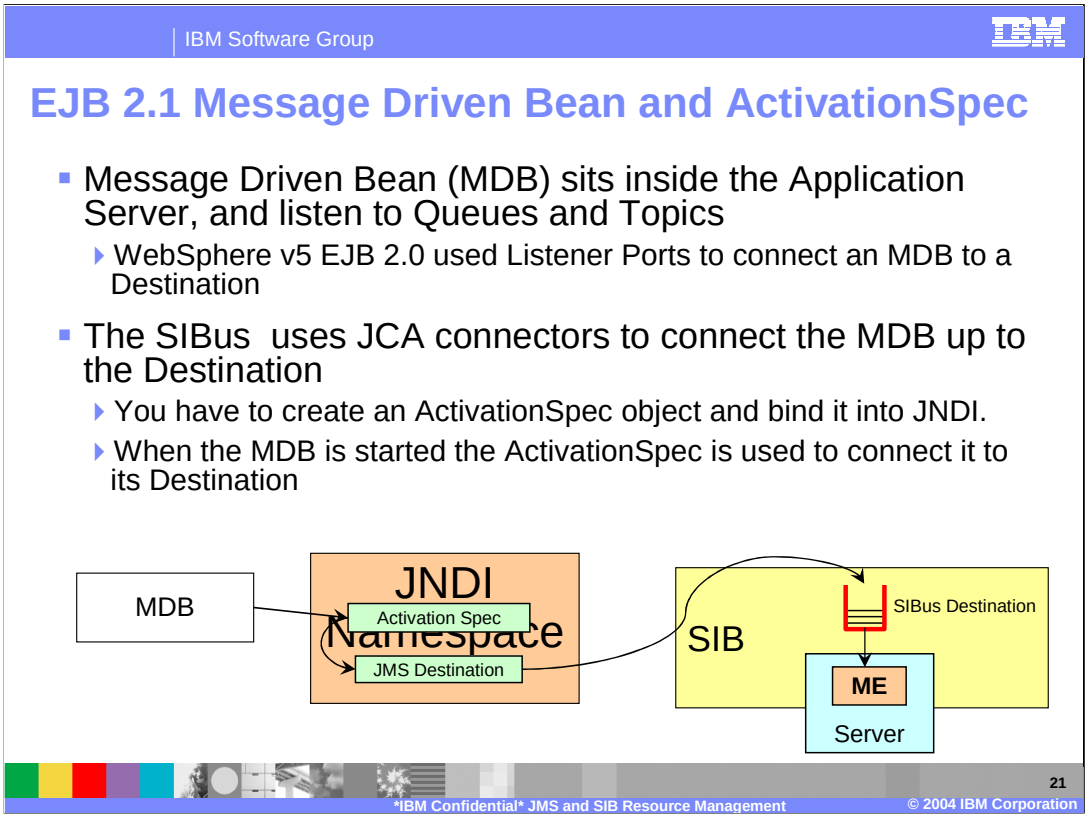
The term "JMS queue" is used to refer to the JMS destination that applications interact with, and which an administrator configures as a JMS resource of the default messaging provider. An application that uses JMS point-to-point messaging acts as a producer or consumer of messages with JMS queues, and has no need to know about other service integration resources that support the JMS queue.

An administrator can define a *JMS queue*, an administrative object that encapsulates the JMS queue name, which applications can look up in the JNDI namespace, and other JMS configuration properties that the administrator wants to preserve. The JMS queue also defines the name of a queue on a service integration bus that is the virtual location on the bus that provides the JMS queue.

Creating SIB and JMS resources with wsadmin

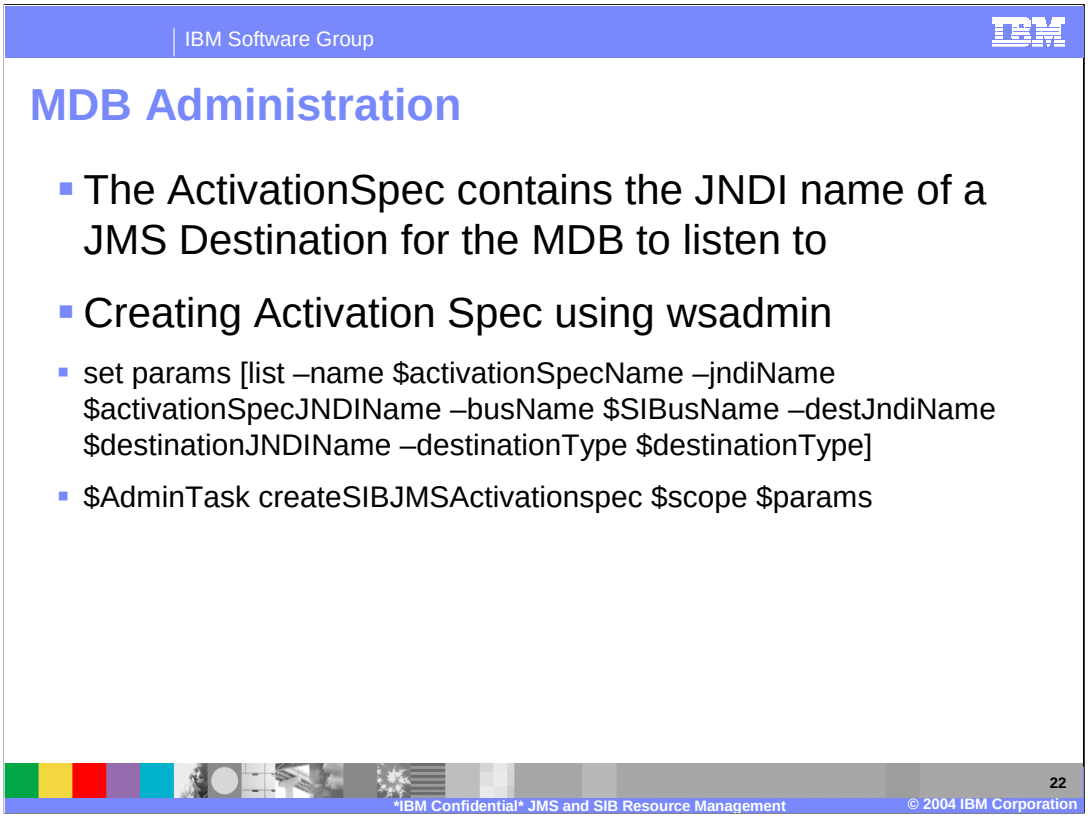
- Most/all of the new SIB Admin commands use the new $AdminTask wsadmin commands
  - To list all of the $AdminTask commands available use:
    - $AdminTask help –commands
  - To get detailed information about a specific command:
    - $AdminTask help <command>
- Example (for creating a JMS Queue that refers a SIB queue)

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



Message Driven Beans that sit inside the AppServer and listen to Queues and Topics. No other Appserver Component can add itself as a Listener to a JMS Destination. Past versions of WAS used Listener Ports to connect an MDB to a Destination, but no-longer (unless you are using full MQ as your JMS Provider). The SIBus uses J2C (JCA) connectors to connect the MDB up to the Destination, and as such you now create an ActivationSpec object and bind it into JNDI. When the MDB is started the ActivationSpec is used to connect it to its 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 on the ActivationSpec in JNDI.

In WebSphere v6, the JCA 1.5 message inflow contract will be used for message delivery to all EJB2.1 MDBs. Platform Messaging and other third party JMS resource adapters will implement this contract natively. For providers that only support ASF an 'ASF resource adapter' will be provided that maps between ASF and JCA message inflow. The properties on the *ActivationSpec* will be the union of those from the MDB and those that would have been defined on the listener port



Admin of Activation Specs

a unique name

set activationSpecName "MyMDBsActivationSpec"

the name the MDB will look up

set activationSpecJNDIName "eis/$activationSpecName"

the bus the MDB will connect to

set SIBusName [$AdminConfig showAttribute [$AdminConfig list SIBus] name]

JNDI Name of a JMS Destination

set destinationJNDIName "jms/MyJMSQueue"

javax.jms.Queue or javax.jms.Topic

set destinationType "javax.jms.Queue"

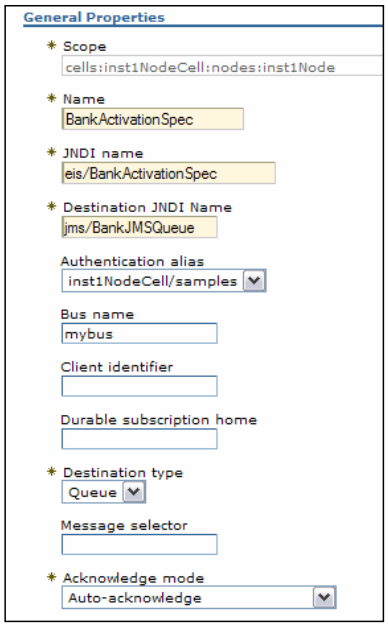
set params [list –name $activationSpecName –jndiName $activationSpecJNDIName –busName $SIBusName –destJndiName $destinationJNDIName –destinationType $destinationType]

$AdminTask createSIBJMSActivationspec $scope $params

$AdminConfig save

Creating Activation Spec using Admin Console

- Resources > Default Messaging Provider > JMS Activation Spec
- When installing the application, bind the MDB to the Activation Spec JNDI Name



Support for EJB 2.0 MDBs

- Existing EJB 2.0 MDBs may be deployed against a listener port configured as in WebSphere v5
- EJB 2.0 MDBs can also use ActivationSpecs as in v6 Platform Messaging
- Any new EJB 2.1 MDBs should use ActivationSpecs

Section

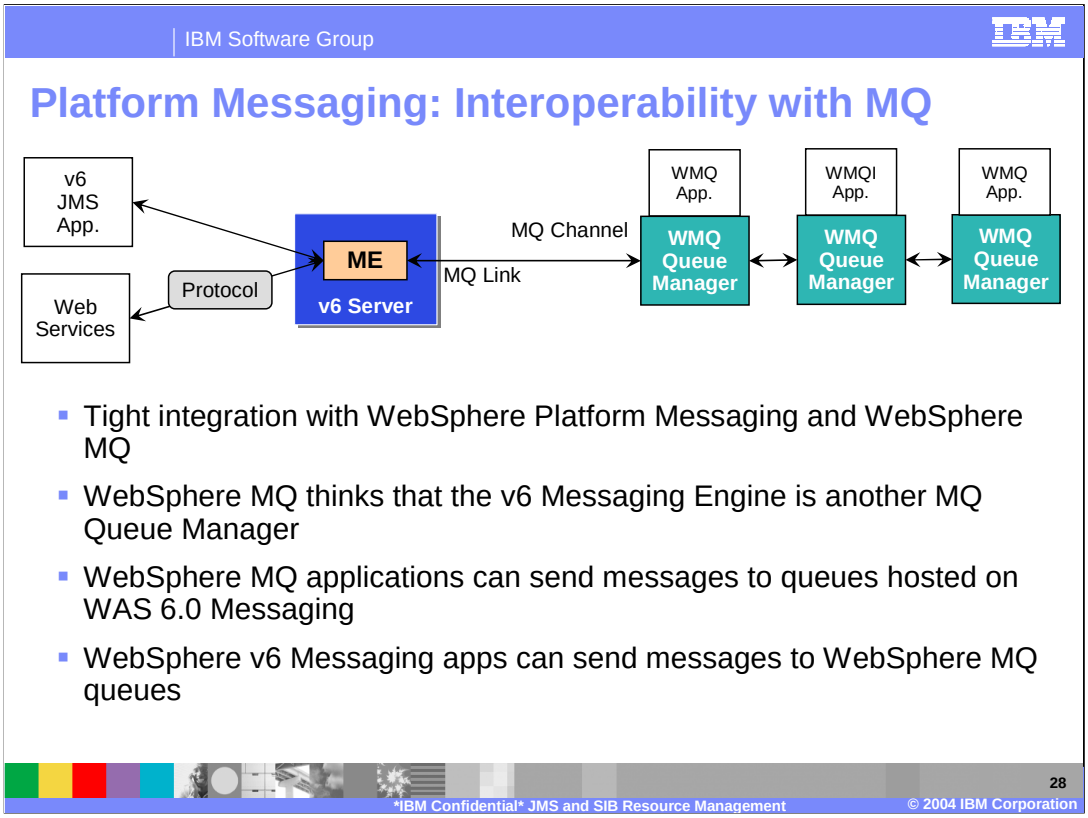
WebSphere Platform Messaging Interoperability

Platform Messaging: Interoperability

- Full interoperability with other SIBus in the same or different cell
- WebSphere v5 Embedded JMS Server interoperation
  - Existing WebSphere v5 embedded JMS clients can connect to v6 destinations
  - v6 JMS application to connect to an embedded JMS provider hosted in a v5 server
  - Note that it is not possible to connect a v5 embedded JMS Server into a v6 SIBus
- **MQ Client Link** can be created to support any old WebSphere v5 clients to talk to WebSphere v6 ME

Relationship to WebSphere MQ

- WebSphere MQ queue manager and/or a WebSphere MQ Integrator or Event Broker can coexist on the same machine as a ME
  - WebSphere MQ and Platform Messaging are separate products and do not share any modules or configuration data
- Connectivity between ME and MQ Queue Manager is established by defining an MQLink
  - MQLink converts between the formats and protocols used by WebSphere MQ and Platform Messaging
- Functions not supported in WebSphere v6
  - An MQ queue manager cannot attach to WPM using any communications protocol other than TCP/IP
  - A PM messaging engine cannot participate in a WebSphere MQ cluster.



You can have multiple MQLinks out of a bus, but each link goes to a different queue manager, and further these queue managers should not be interconnected. The link engine can be part of a cluster, but the issue is in handling failover. The ME hosting the MQLink must keep a fixed host/port address because that is what MQ expects, and so you have to marry the new WAS HA support with more traditional HACMP like HA solutions.

WebSphere MQ Link details

- WebSphere MQLink enables a ME to exchange messages with applications in a WebSphere MQ network.
  - You can have multiple MQLinks out of a bus, but each link goes to a different queue manager
    - ME that supports the MQLink is known as **link engine**
  - The link engine must have a static endpoint
- To define an MQ Link
  - Select the Messaging Engine in the Admin Console to create WebSphereMQ link
  - Enter name, queue manager name, sender channel details, receiver channel details etc

You can have multiple MQLinks out of a bus, but each link goes to a different queue manager, and further these queue managers should not be interconnected. The link engine can be part of a cluster, but the issue is in handling failover. The ME hosting the MQLink must keep a fixed host/port address because that is what MQ expects, and so you have to marry the new WAS HA support with more traditional HACMP like HA solutions.

Usage Scenarios

- Use Platform Messaging:
  - Customers and J2EE developers currently using WAS v5 embedded JMS provider for intra-WAS messaging
  - Messaging between WAS and existing MQ backbone and its applications
- Use WebSphere MQ:
  - Customers currently using WebSphere MQ may continue to use it.
  - Access is required to heterogeneous non-JMS applications, WebSphere MQ clustering, or other WebSphere MQ functions

The messaging support of the default messaging provider is only accessible from WebSphere Application Server Web, EJB and client containers, and is interoperable with WebSphere MQ. If access is required to heterogeneous non-JMS applications, WebSphere MQ clustering, or other WebSphere MQ functions, you should install WebSphere MQ as a messaging provider.

Migration from v5

- When upgrading v5 single server, the existing JMS Server will be replaced by a Messaging Engine
- When upgrading a v5 ND node, JMS Server will be converted to an application server hosting an ME
- Existing JMS JNDI resource definitions remain unchanged
  - All applications will connect into new messaging engine using WAS 5.0 emulation mode. Permits phased migration from other servers/client environments
  - Old JMS JNDI definitions can be reset to new provider as required
- JMS applications deployed to WAS 6.0 servers can connect to WAS 5.0 servers in support of mixed version cells

Summary

- JMS and SIB resources can be managed with the Admin Console or wsadmin
- MQLink can be created to link a WPM messaging engine to a MQ Queue manager

Template Revision: 10/01/2004 4:46 PM

Trademarks and Disclaimers

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

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

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.

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.

This Page Intentionally Left Blank