#SHAREorg

SHARE
Technology · Connections · Results

Using IBM WebSphere Application Server and IBM WebSphere MQ Together

Chris J Andrews
IBM

Thursday 15th August, 2013
Session 13885

SHARE in Boston

Agenda

- Why are we here?
- Messaging in WAS
  - What is JMS?
- Accessing WMQ from WAS
  - How to configure
  - New WMQ V7 features useful for WAS
- Configuring WMQ and WAS
  - Useful tuning parameters
  - High Availability and clusters

Why are we here?

Some reasons for being here...

“We currently have applications using WebSphere MQ, and now we have applications running in WebSphere Application Server that need to access the same data”

“We’re writing applications for WebSphere Application Server, we’re trying to choose a messaging provider and are thinking that WebSphere MQ might fit the bill”

…

How do we best communicate between WebSphere Application Server and WebSphere MQ?

- WebSphere Application Server is a fully compliant Java Enterprise Edition (**JEE**) Application Server
  - Provides integrated support for application messaging using Java Message Service (**JMS**) providers.
- WebSphere MQ is a fully compliant JMS provider.
- Therefore, JMS is the answer!

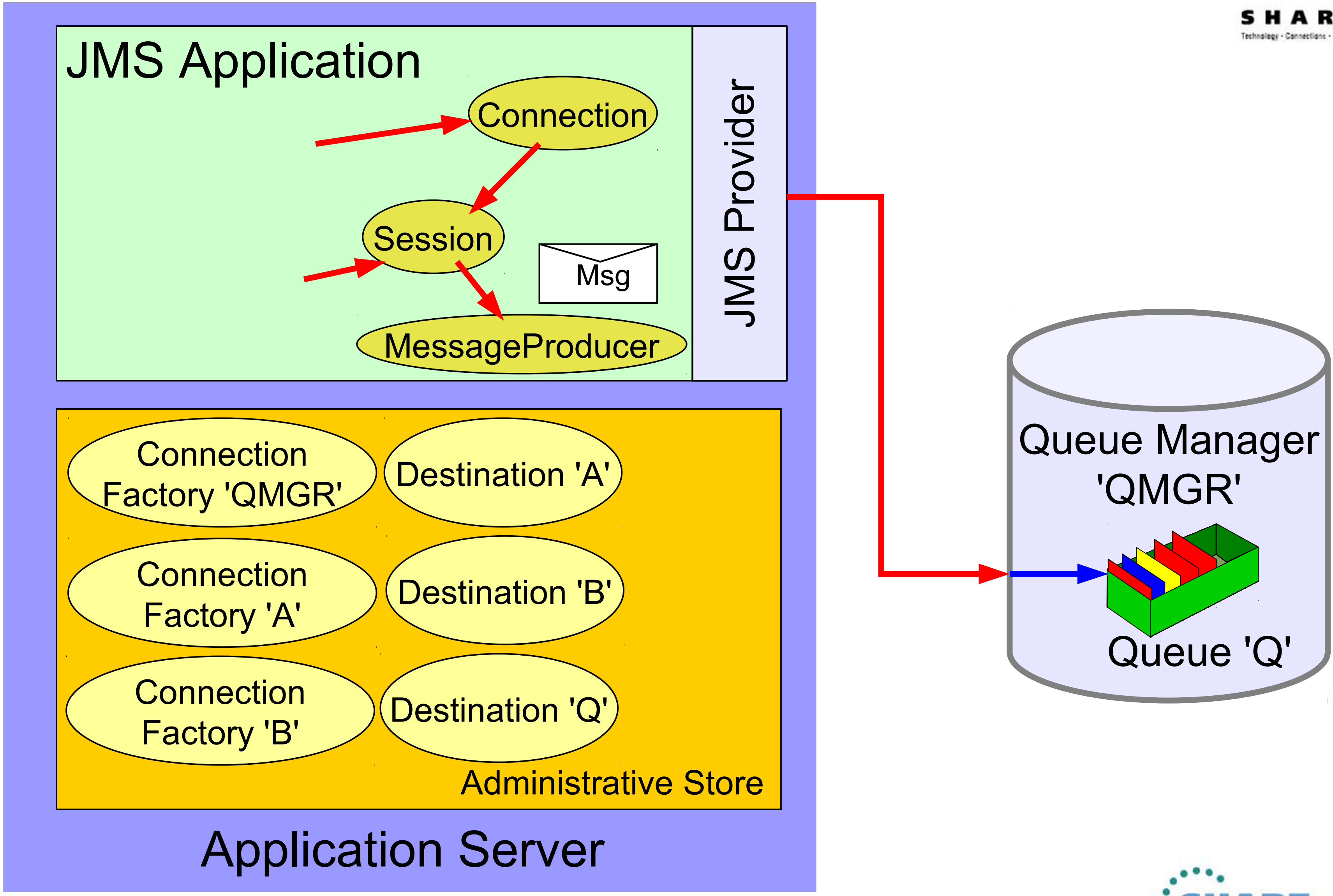
Agenda

- Why are we here?
- Messaging in WAS
  - What is JMS?
- Accessing WMQ from WAS
  - How to configure
  - New WMQ V7 features useful for WAS
- Configuring WMQ and WAS
  - Useful tuning parameters
  - High Availability and clusters

What is JMS?

- A standardized Java API to allow applications to perform messaging
  - Applications can be unaware of implementation
  - *Not a wire protocol!*
- Uses administered objects to define provider and configuration
  - Can be changed without recompiling application
- Supports point-to-point messaging, and publish/subscribe messaging

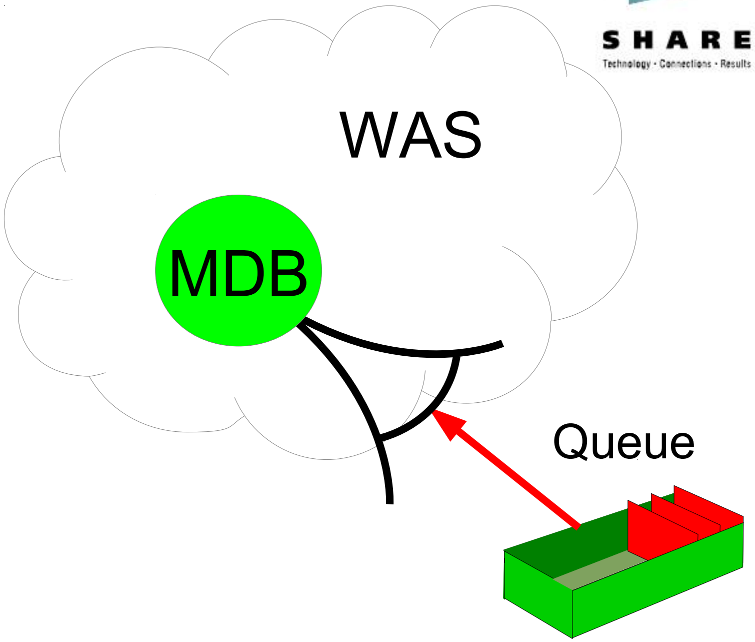
How do JMS based applications function?

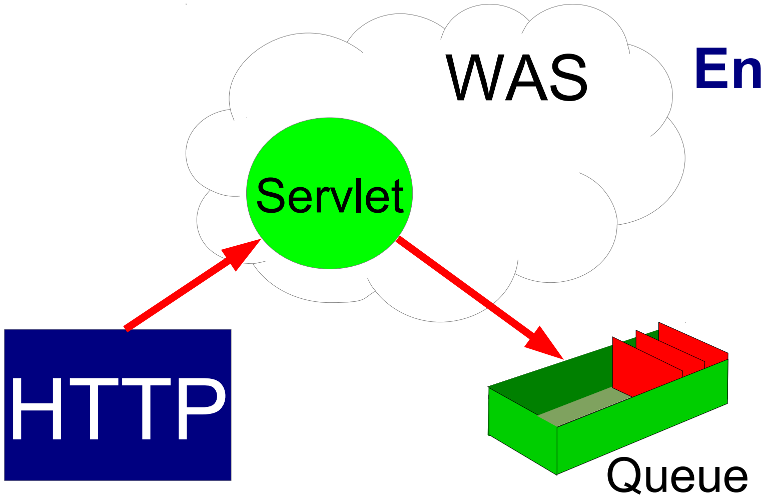


Types of JMS applications in WAS

Message Driven Bean (MDB):

- Specific type of EJB which has an “onMessage” Java method to process messages
- JEE Container (WAS) deals with getting the message, transactions, and running the application.
- Mechanism to get messages **into** WAS





Enterprise Java Bean (EJB), Servlets/JSPs:

- Application creates connections and sends/receives messages to WMQ
- Container provides connection pooling and transaction management

Message Driven Beans disected

Method invoked when a message is available

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

Enterprise Java Beans (EJB)

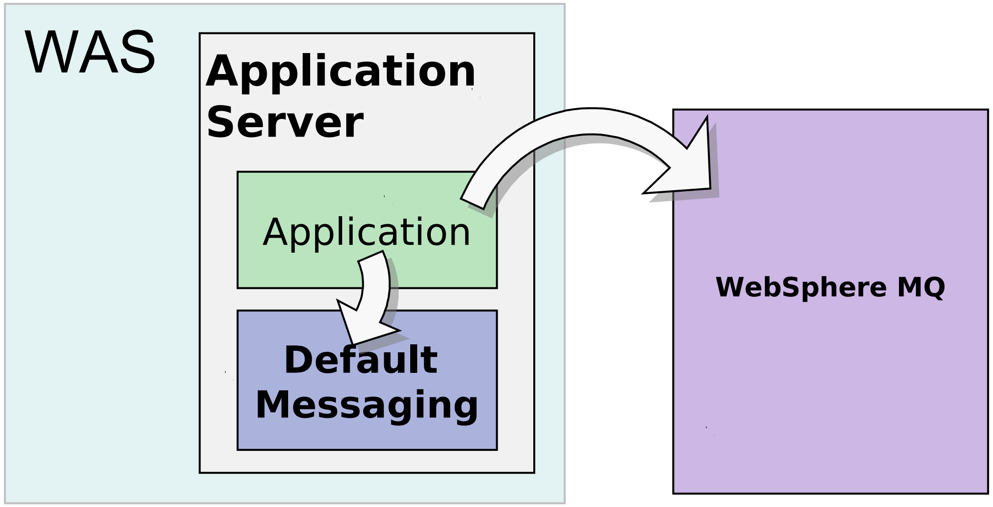
```
@Resource()
private ConnectionFactory cf;
@Resource()
private Queue q;
public void receiveMessage() {
  try  {
    Connection conn = cf.createConnection();
    conn.start();
    Session sess =
            conn.createSession(true,Session.AUTO_ACKNOWLEDGE);
    MessageConsumer consumer = sess.createConsumer(q);
    Message msg = consumer.receive(30000);
    if (msg instanceof TextMessage) {
      System.out.println(“Message received:“ +
                              ((TextMessage) msg).getText());
    }
   conn.close();
  } catch (Exception ex) {
   System.out.println("Exception : " + ex);
}}
```

Define EJB 3 Resource references that will be initialised at runtime

More code required to get a message compared to an MDB, and less efficient

WAS JMS messaging *in a slide*

- WebSphere Application Server provides first class support for two JMS messaging providers
  - (1) An embedded messaging provider, the ***Default Messaging provider*** (or **Service Integration Bus**)
  - (2) **WebSphere MQ**
- The nature of JMS and JEE allows easy switching between providers with little or no application changes



Agenda

- Why are we here?
- Messaging in WAS
  - What is JMS?
- Accessing WMQ from WAS
  - How to configure
  - New WMQ V7 features useful for WAS
- Configuring WMQ and WAS
  - Useful tuning parameters
  - High Availability and clusters

Ways of accessing WMQ from within WAS

- Three key options:

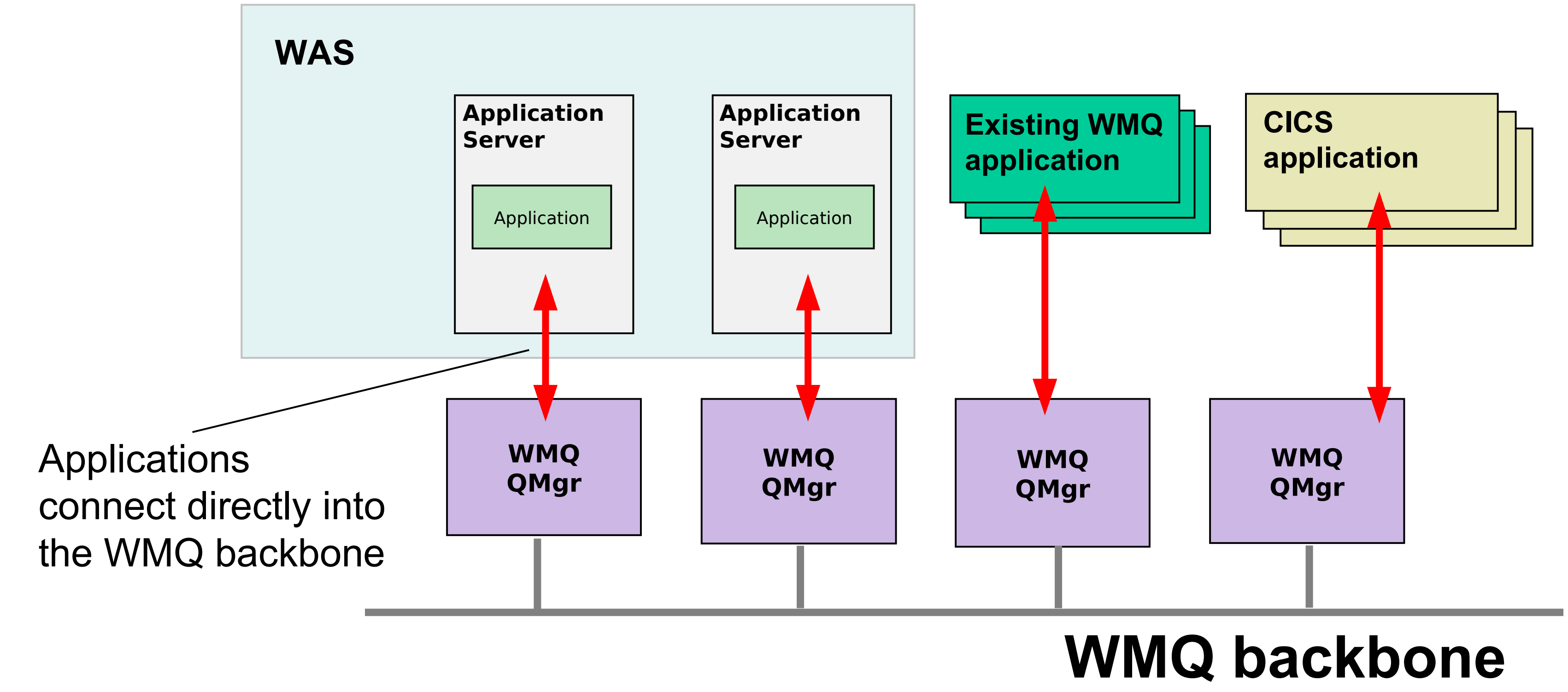
1. Use WMQ as the JMS provider in WAS

2. Use the service integration bus in WAS and connect that messaging system with your WMQ backbone

3. Use a mixture of both
   - *SIBus for intra-WAS messaging*
   - *WMQ for inter-system messaging*

Option 1:

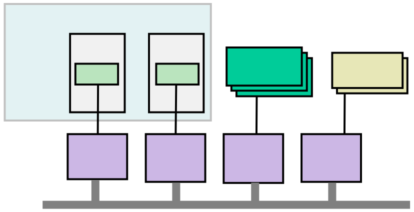
WebSphere MQ as the JMS provider in WAS



Applications connect directly into the WMQ backbone

Option 1: WMQ as the JMS provider in WAS

- ☑ **Pros**
  - The most **direct** route to WMQ queues
    - *Best performing way to get messages on and off a WMQ queue*
    - *Least moving parts*
      - *Least things to configure*
      - *Least places where things can go wrong*
  - A single messaging system to maintain rather than two

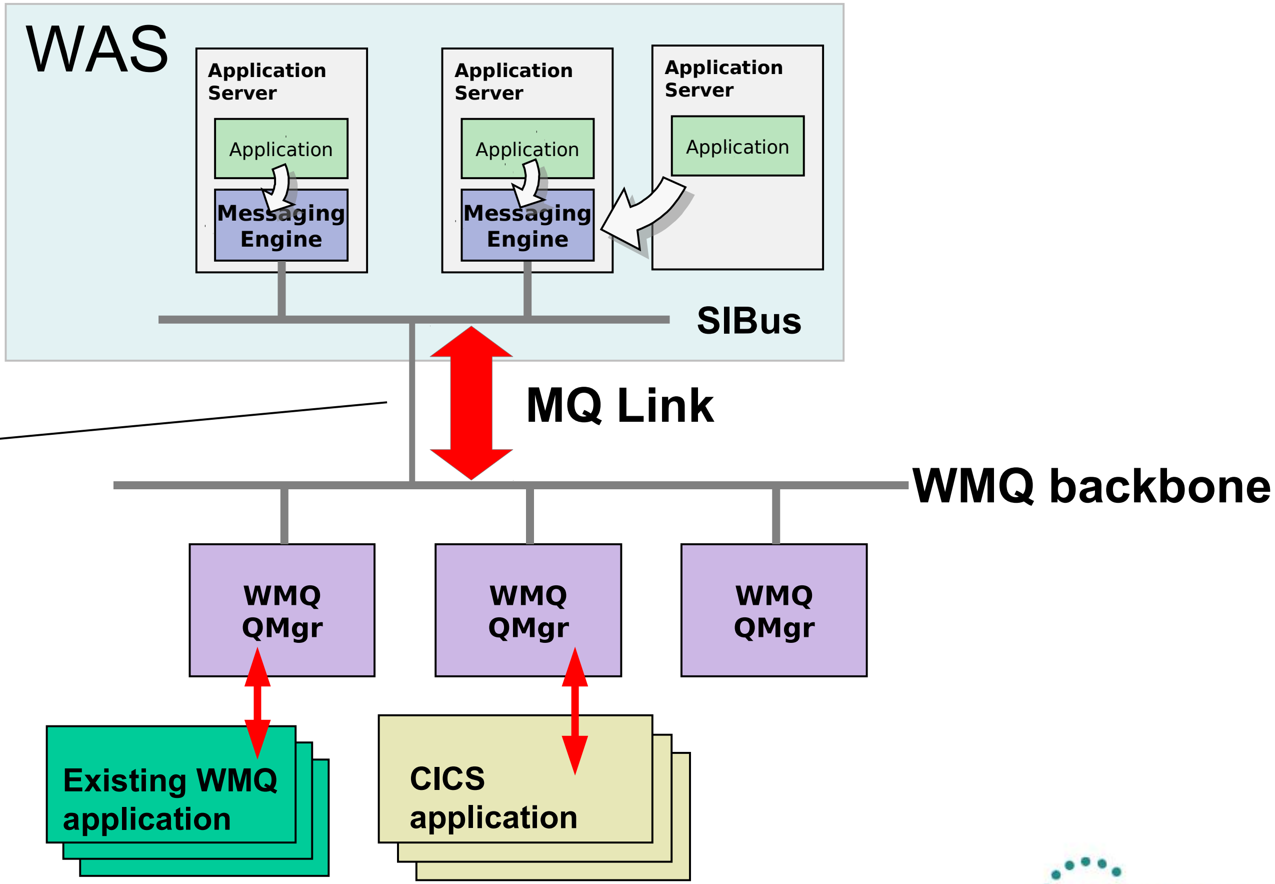


- ☒ **Cons**
  - WAS application deployment depends on knowledge of WMQ topology
    - *e.g. the WMQ queue location is required when connecting applications*
  - WAS applications dependent on two systems being simultaneously available
  - Can be complex to exploit WAS and WMQ high availability and scalability features

Option 2:

SIBus for WAS application messaging

SIBus interacts with WMQ to access the messaging backbone

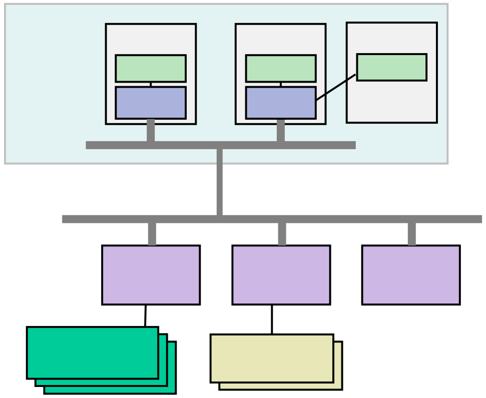


Messages flow between SIBus and WMQ where interactions with the WMQ backbone are necessary

Option 2: SIBus messaging system

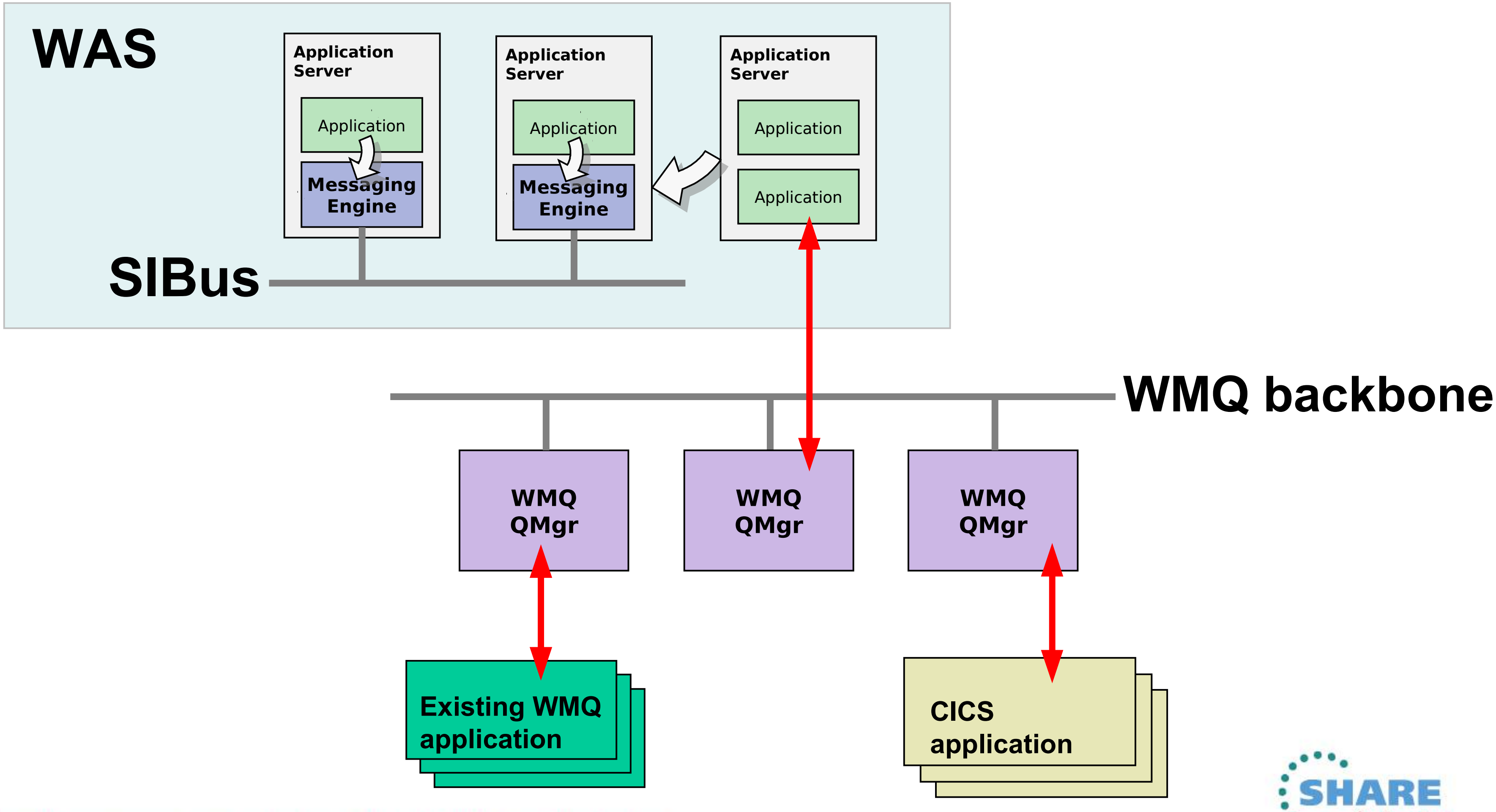
- ✓ **Pros**
  - Messaging runtime is embedded into the same runtime as the applications
  - Ability to asynchronously queue messages in the WAS system while connectivity with the WMQ backbone is unavailable
  - JMS resource configuration relates to WAS-defined SIBus configuration objects
    - *WMQ-aware configuration in WAS is minimised*
  - Applications can be unaware of actual queue locations

- ✗ **Cons**
  - Two messaging systems to configure and maintain
  - Additional complexity in configuring interactions between SIBus and WMQ
  - Performance overhead when passing from one messaging system to another



Option 3 : Mixture of both

SIBus for WAS-WAS application messaging,
WMQ for WAS-‘*other*’ messaging

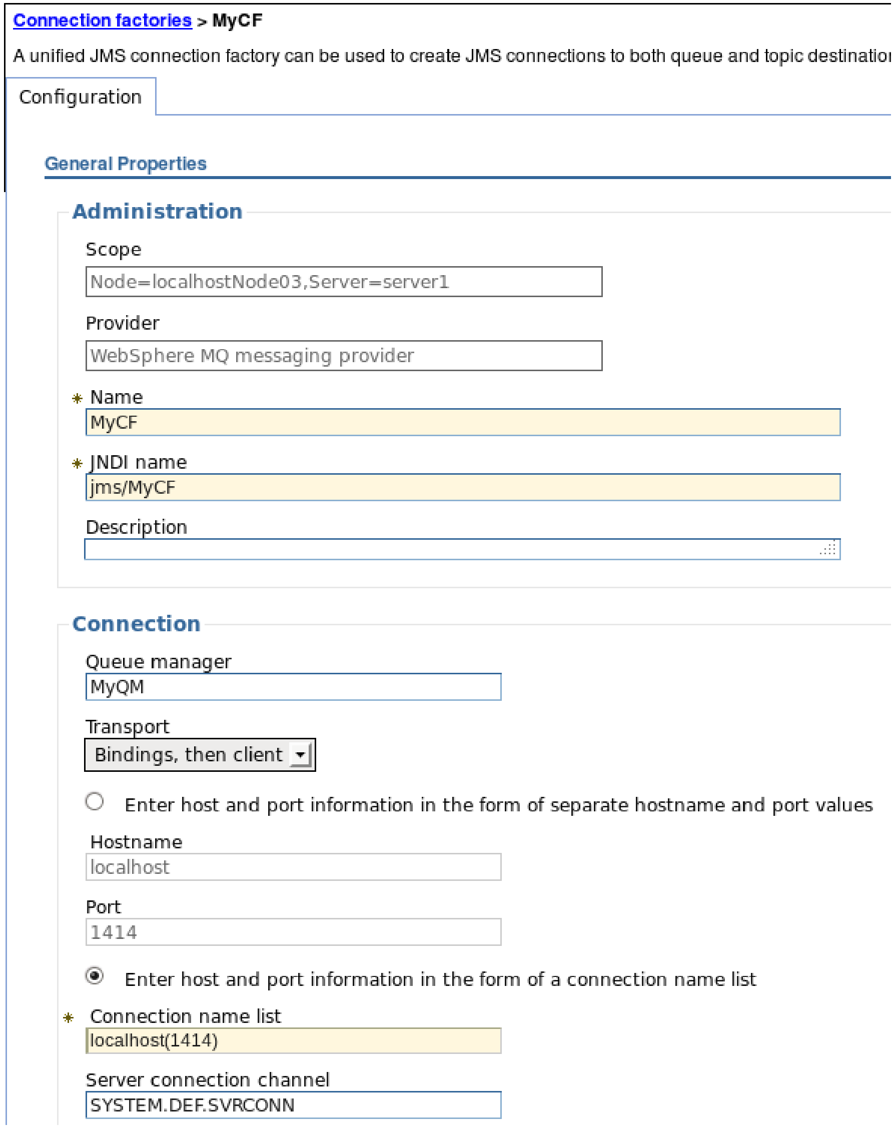


WebSphere MQ as the messaging provider in WAS 7+

- Pre-deployed and ready to go
- Fully integrated into WAS administrative tools
  - Can even do limited WMQ administration from WAS
- Access to WMQ via TCP or shared memory (“bindings”)
- Supports connections to variety of WMQ queue manager versions
- Interfaces with WAS using the well-defined JCA interface
  - WebSphere MQ provided as a single unit “Resource Adapter”
- WMQ Resource Adapter is a component of WAS
  - Updates applied using WAS process, not WMQ

Configuring for WMQ : Connection Factories

Resources > JMS > Connection factories > [New]



Connection factories > MyCF

A unified JMS connection factory can be used to create JMS connections to both queue and topic destinatio

Configuration

General Properties

Administration

Scope
Node=localhostNode03,Server=server1

Provider
WebSphere MQ messaging provider

* Name
MyCF

* JNDI name
jms/MyCF

Description

Connection

Queue manager
MyQM

Transport
Bindings, then client

○ Enter host and port information in the form of separate hostname and port values

Hostname
localhost

Port
1414

● Enter host and port information in the form of a connection name list

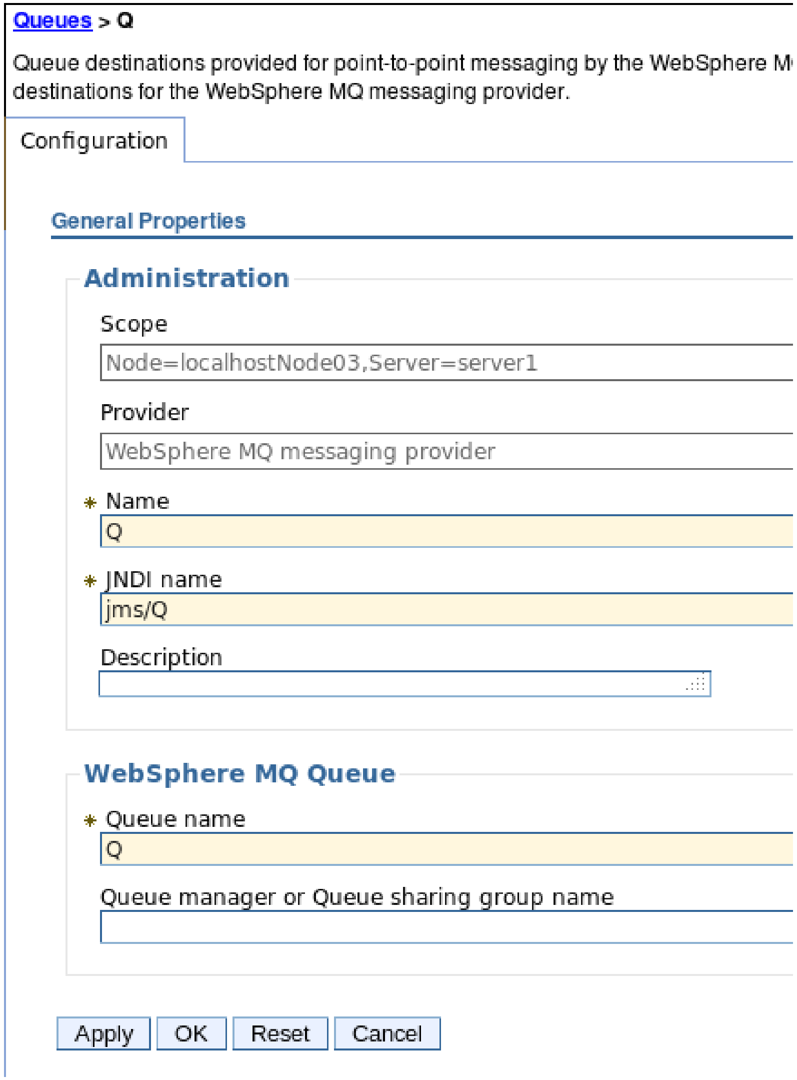
* Connection name list
localhost(1414)

Server connection channel
SYSTEM.DEF.SVRCONN

- Specify how an application connects to a WMQ queue manager
- Typically requires:
  - Queue manager name
  - Hostname and port
  - Channel name
  - Other parameters such as:
    - *Transport type (client or bindings)*
    - *Use of SSL*
- Or, can use WMQ **client channel definition table** (CCDT) URL

Configuring for WMQ : Destinations

Resources > JMS > Queues / Topics > [New]



Queues > Q

Queue destinations provided for point-to-point messaging by the WebSphere M
destinations for the WebSphere MQ messaging provider.

General Properties

Administration

Scope: Node=localhostNode03,Server=server1

Provider: WebSphere MQ messaging provider

* Name: Q

* JNDI name: jms/Q

Description:

WebSphere MQ Queue

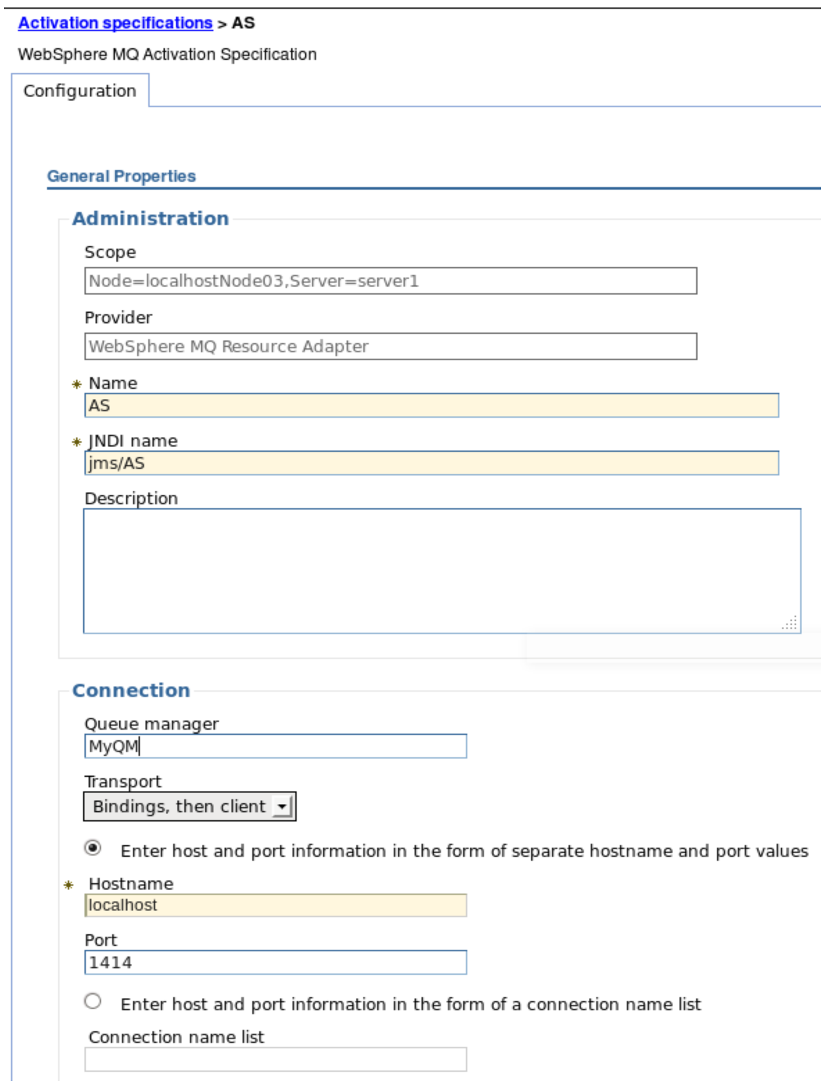
* Queue name: Q

Queue manager or Queue sharing group name:

- Defines references to the resources in WMQ that a JMS application will use
  - – The WMQ resources must be created using WMQ administration
- **Queues**
  - – Identifies the actual queue in WMQ
    - *Can be used to set properties such as persistence, priority, etc.*
- **Topics**
  - – Defines the WMQ publish/subscribe destination

Configuring for WMQ : Activation Specifications

Resources > JMS > Activation Specifications > [New]



Activation specifications > AS

WebSphere MQ Activation Specification

Configuration

General Properties

Administration

Scope
Node=localhostNode03,Server=server1

Provider
WebSphere MQ Resource Adapter

* Name
AS

* JNDI name
jms/AS

Description

Connection

Queue manager
MyQM

Transport
Bindings, then client

● Enter host and port information in the form of separate hostname and port values

* Hostname
localhost

Port
1414

○ Enter host and port information in the form of a connection name list

Connection name list

- Activation Specs are the standardised way of delivering messages to an MDB
- The WebSphere MQ messaging provider in **WAS V7** adds support for Activation Specifications
  - Listener Port functionality is stabilized.
- Activation Specs combine the configuration of connectivity, the JMS destination to be processed and the runtime characteristics of the MDB itself
- Can be tested during creation to verify they work.
- Activation specifications can be defined at all WAS configuration scopes, as can be done for ConnectionFactories and Destinations.

Agenda

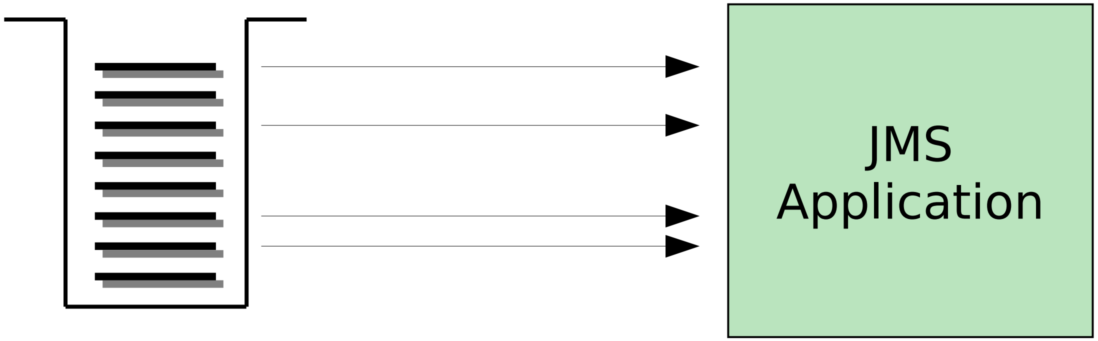
- Why are we here?
- Messaging in WAS
  - What is JMS?
- Accessing WMQ from WAS
  - How to configure
  - New WMQ V7 features useful for WAS
- Configuring WMQ and WAS
  - Useful tuning parameters
  - High Availability and clusters

New functional areas in WMQ V7 of interest to JMS

- Message Properties + Selectors
- Asynchronous Consumption of messages
- Simplified Browse + Co-operative Browse
- Asynchronous Put Response
- Read-ahead of messages
- Connection changes
- Multi-instance Queue Managers (WMQ 7.0.1)

Message properties and Selectors

- JMS Selectors allow messages from a destination to be filtered
  - Filter on system or user message properties
- WMQ V7 queue manager understands JMS message properties
- Queue manager evaluates JMS selector
  - Only transmits matching messages to the client
  - Efficient delivery of messages

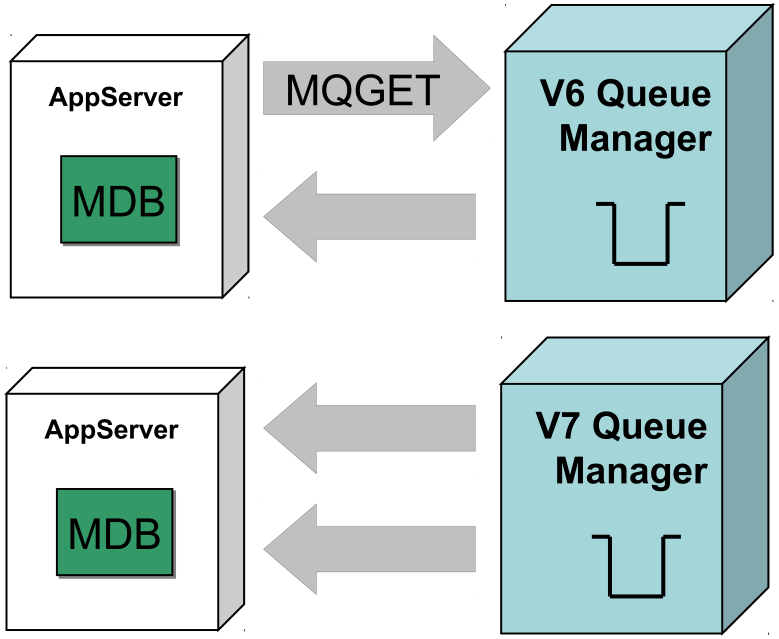


Asynchronous Consume

- JMS has had the concept of asynchronous consumer since inception
  - MessageListener / onMessage code

- Event driven processing ‘natural’ for the Java Environment

- WMQ V7 introduced ability of queue manager to drive consumer when suitable message available
  - Avoids need for 'polling thread'
  - More efficient message delivery, especially with selectors.



- With proper support in WMQ – JMS implementation simplified

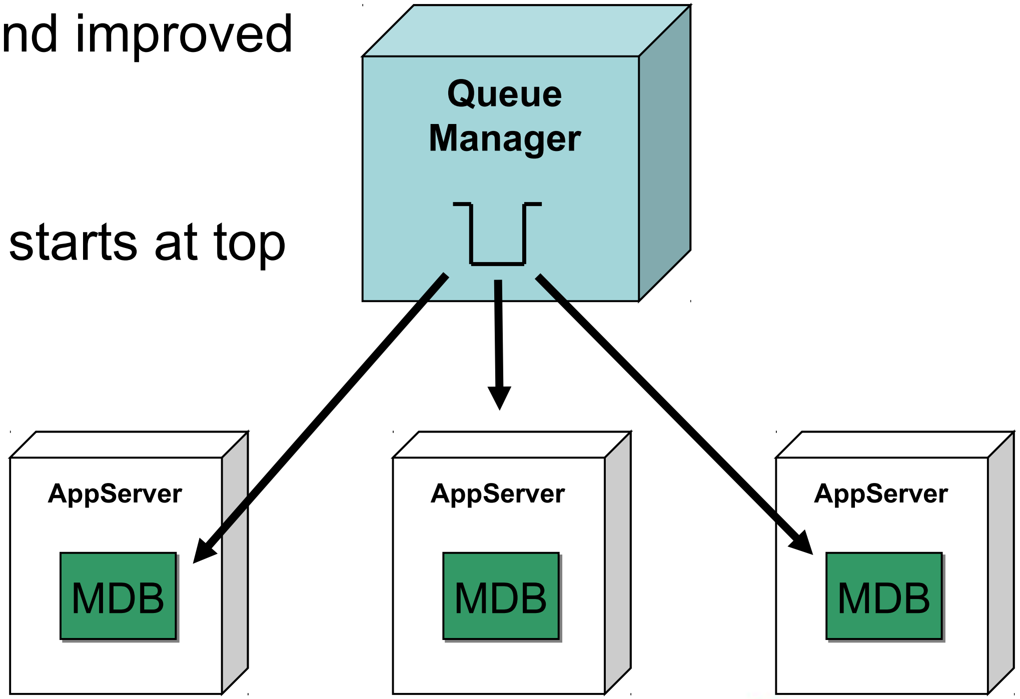
- Less client CPU overhead of this busy thread

Asynchronous Consumer and Browse-with-mark

- Multiple application servers will contend for messages when processing the same queue.
  - Can result in failed message delivery and wasted resources

- Browse-with-mark allows the application server to hide messages it has browsed and intends to process
  - No contention for messages, and improved throughput.

- The MDB browse thread always starts at top
  - Message priority honoured



WAS 8.5.5 – Liberty Profile

WAS 8.5 introduced a new 'lightweight' profile, called the “Liberty” profile, intended for use by developers.

Out-of-the-box, the Liberty profile provides no WebSphere MQ connectivity.

Instructions for obtaining the WMQ-RA:
http://www.ibm.com/support/docview.wss?uid=swg21633761

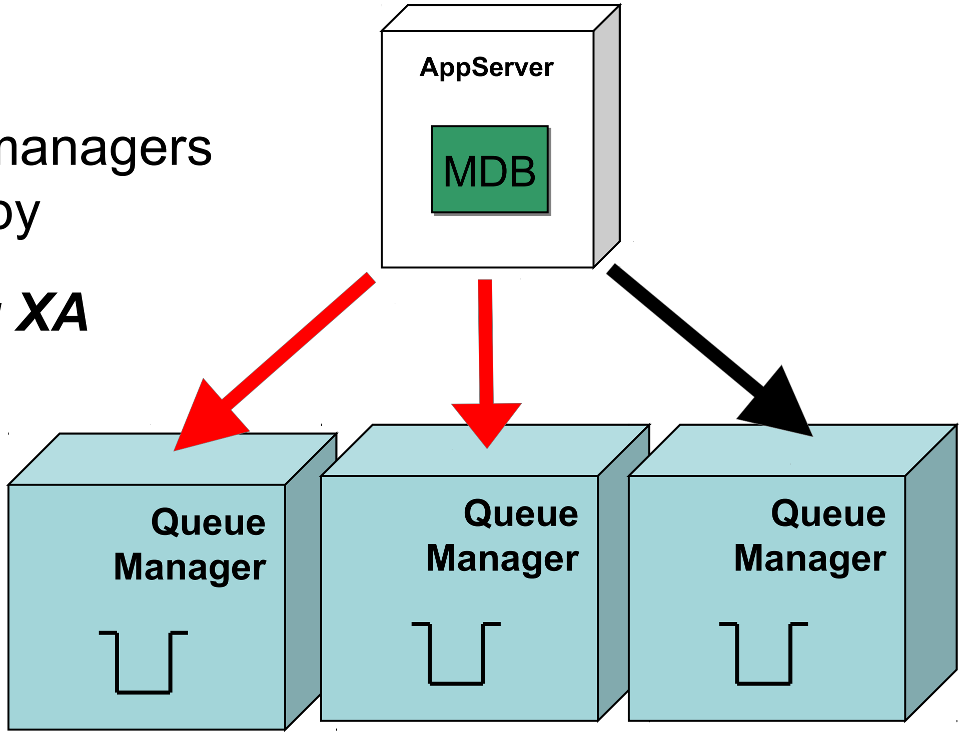
Requires WMQ-RA 7.5.0.1 + IC92914 (included in WMQ-RA 7.5.0.2)
http://www.ibm.com/support/docview.wss?uid=swg1IC92914
To correct problems with: various trace problems (NPE to formatting), JMS UserID field could not be changed from the JVM process ID, SHARECNVALLOWED property ineffective.

Agenda

- Why are we here?
- Messaging in WAS
  - What is JMS?
- Accessing WMQ from WAS
  - How to configure
  - New WMQ V7 features useful for WAS
- Configuring WMQ and WAS
  - Useful tuning parameters
  - High Availability and clusters

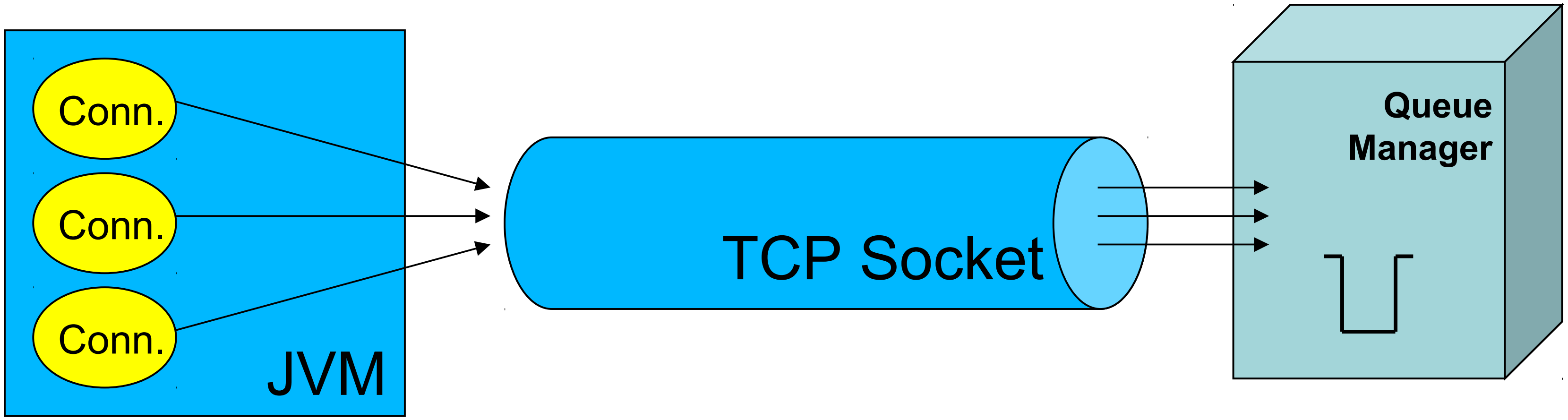
Connection Name List

- This property specifies a list of hostnames and ports to attempt to connect to.
  - Comma-separated list of “hostname(port)” entries
- Entries in the list are tried until a working connection is found, or the end of the list is reached.
- Used with multi-instance queue managers automatically reconnect to standby
- ***Must be same QM data if using XA***
- Standard property in WAS V8
  - WAS V7 custom property



Shared Conversation Allowed

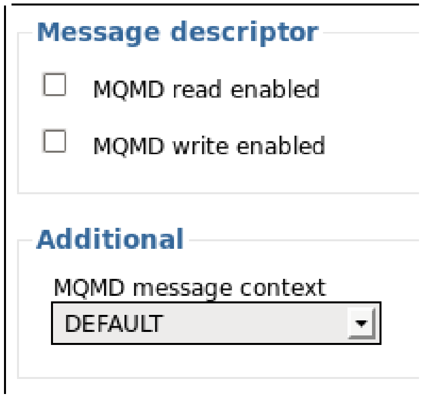
- This property specifies whether JMS applications that use this Factory can share their connection to a Version 7+ queue manager.



- Useful to reduce the number of network connections to a queue manager.
- Can have slight performance impact.
  - Multiple JMS applications will be sending data to a queue manager and waiting for a response over the same channel.
  - Set server connection channel SHARECNV to 1 for maximum performance *for high data throughput JMS Connections*.

Using WMQ JMS with existing WMQ applications

- MQMD Read/Write enabled
  - Determines if MQMD properties can be set via JMS get/set message property methods.
  - Allows full access to the MQMD header values
  - Useful for sending or receiving messages from MQ applications that use specific header properties.
  - JMS message property names begin with "JMS_IBM_MQMD…"



Message descriptor

□ MQMD read enabled

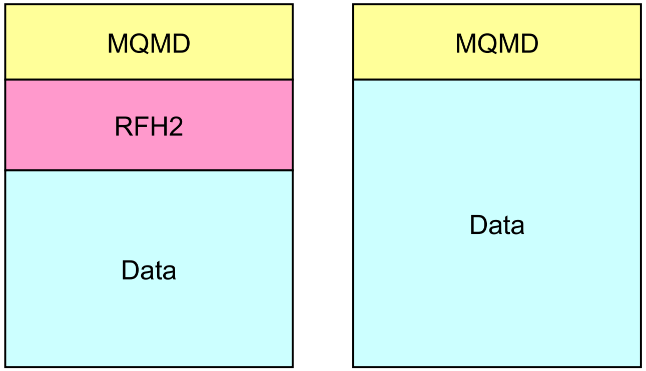
□ MQMD write enabled

Additional

MQMD message context

DEFAULT

- Target Client (JMS/MQ)
  - Indicates what the receiving application is
  - Removes the RFH2 header from **sent** messages WMQ applications may have issues with

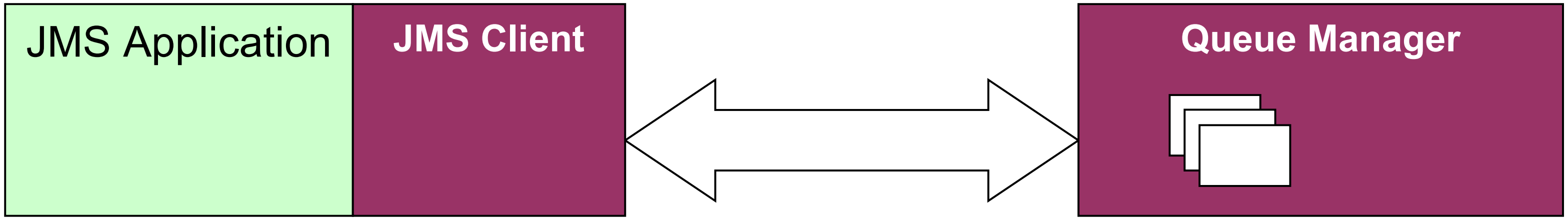


JMS Messages

Non-JMS Messages

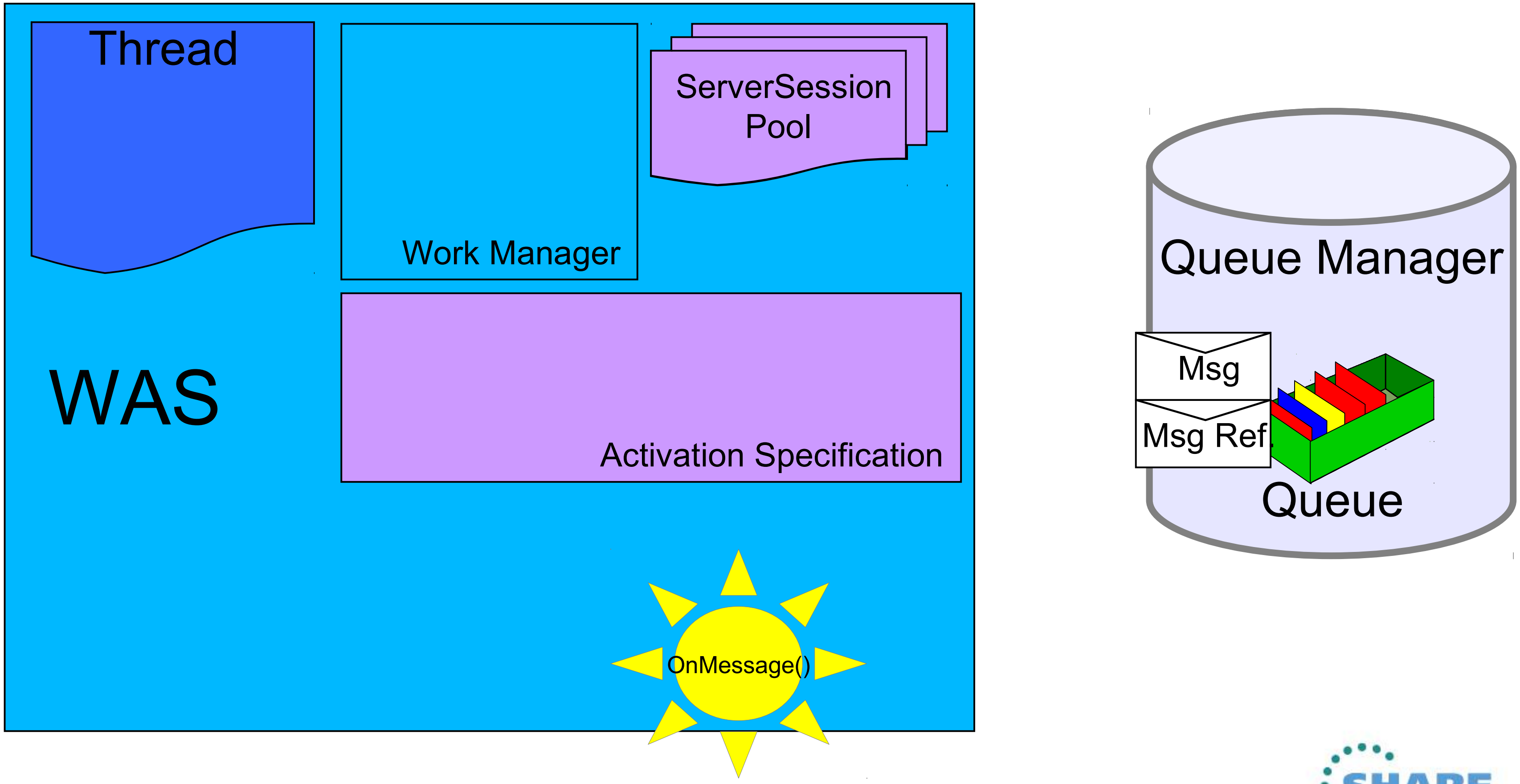
Read Ahead Allowed

- In general, messages are sent to JMS applications one at a time.
- The Read Ahead Allowed property tells the queue manager whether non-persistent messages can be streamed to the client application in preparation for them being consumed.
  - Messages are stored in a buffer on the client.
  - If the client application terminates unexpectedly, all unconsumed non-persistent messages are discarded.



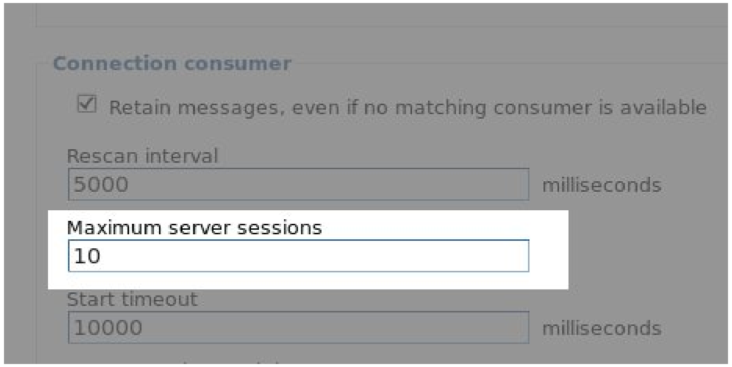
WAS Configuration: Activation Specifications

- How are messages destined for MDBs processed?



Activation Specifications - Threads

- To process a message, a ServerSession **and** a thread is required
- Activation Specification parameter Maximum server sessions configures ServerSession pool, default 10
  - ServerSession pool per MDB
- Application Server Thread pool WMQJCAResourceAdapter used, default Maximum size 25
  - Thread pool used by all MDBs.
- So, by default, 3 MDBs could exhaust the threads
  - Will cause slower processing
  - Recommendation is that thread pool maximum accounts for all MDB maximums



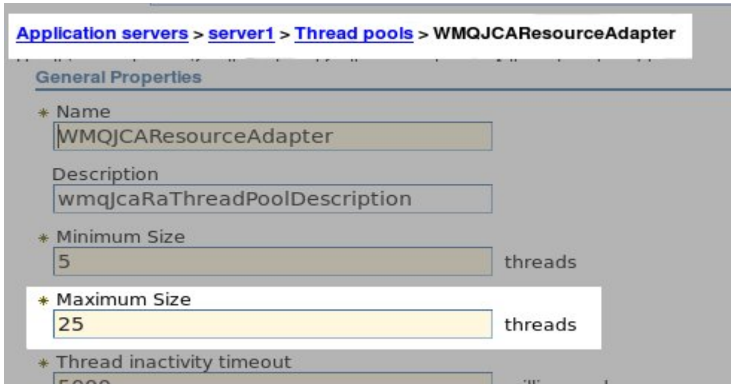
Connection consumer

☑ Retain messages, even if no matching consumer is available

Rescan interval
5000 milliseconds

Maximum server sessions
10

Start timeout
10000 milliseconds



Application servers > server1 > Thread pools > WMQJCAResourceAdapter

General Properties

* Name
WMQJCAResourceAdapter

Description
wmqJcaRaThreadPoolDescription

* Minimum Size
5 threads

* Maximum Size
25 threads

* Thread inactivity timeout

Activation Specifications - Recovery

- What happens if the connection to a queue manager used by Activation Specifications is broken?

- The Activation Specification has recovery features to allow it to attempt several reconnection attempts before stopping.

- Default is to try 5 times in 5 minute intervals.
  - Configured using reconnectionRetryCount / reconnectionRetryInterval (ms) on the Resource Adapter, for all Activation Specifications.

- If the Activation Specification stops, this is only reported in Application Server SystemOut.log logs.

Activation Specifications - Recovery

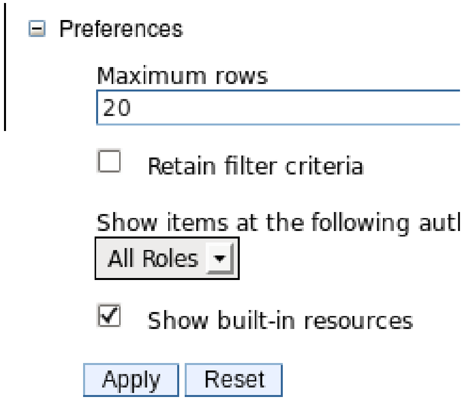
- To configure WMQ RA settings, go to Resources → Resource Adapters, select a scope to configure, and in the Preferences, select “Show built-in resources”, and Apply.

- Select the “WebSphere MQ Resource Adapter”

- Select custom properties, and modify the “**reconnectionRetryCount**” and “**reconnectionRetryInterval**” properties

- Upon first failure, the RA will try an immediate reconnection attempt, and if that fails, retry at the configured interval.

- The properties are configured on the RA, and so affect ALL Activation Specifications using that RA



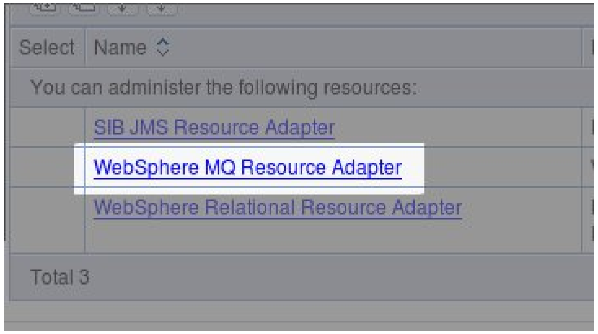
⊟ Preferences

Maximum rows
20

□ Retain filter criteria

Show items at the following autl
All Roles

☑ Show built-in resources



Select	Name
You can administer the following resources:	
	SIB JMS Resource Adapter
	WebSphere MQ Resource Adapter
	WebSphere Relational Resource Adapter
Total 3	

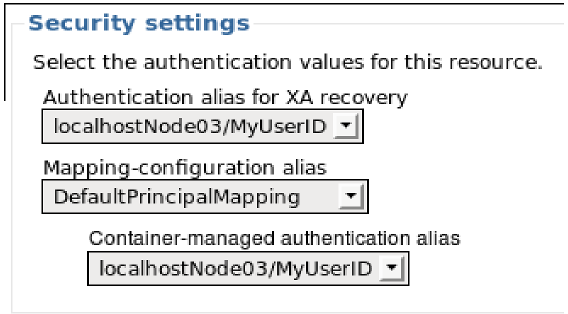
JMS User authentication

`createConnection(user,password)`

- In JEE, JMS application-specified user and passwords are not necessarily used.
  - “Container-managed” authentication

- Activation Specifications / Connection Factories can be associated with authentication data
  - JAAS – J2C Authentication Data defines username and password details
  - The application needs to use JEE resource-references to access the connection factory
  - The *authenticationType* parameter needs to be set to container for container-managed authentication.

- As for other WMQ clients, security exits are required to validate passwords, WMQ only checks user id
  - User ids more important with WMQ 7.1 channel authentication



Security settings

Select the authentication values for this resource.

Authentication alias for XA recovery: localhostNode03/MyUserID

Mapping-configuration alias: DefaultPrincipalMapping

Container-managed authentication alias: localhostNode03/MyUserID

Related Items

- JAAS - J2C authentication data

Agenda

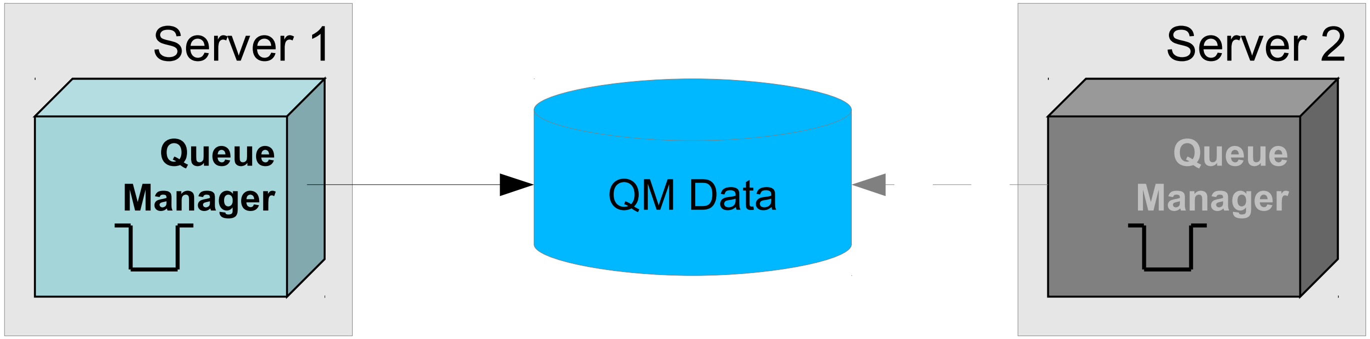
- Why are we here?
- Messaging in WAS
  - What is JMS?
- Accessing WMQ from WAS
  - How to configure
  - New WMQ V7 features useful for WAS
- Configuring WMQ and WAS
  - Useful tuning parameters
  - High Availability and clusters

High Availability

- Treat WAS and WMQ as separate resources
- WMQ V7.0.1 features make it easier for WAS to reconnect during failover
  - Quicker detection of broken connections
  - Connection Name Lists
- OS-level HA solutions (HACMP etc.)
  - Active – Passive failover
  - Transparent as resources such as IP addresses are moved over
  - Can include WMQ as well as other resources in the same failover
  - Third party solution, introduces expense

Multi-instance Queue Managers

- WMQ's implementation of Active – Passive failover
  - Only moves WMQ resources
  - Can be faster than OS-level HA
- Needs data to be stored in external storage with locking, such as NFS V4
- One active queue manager, with other standby instance
- Standby instance can detect failure and take over
  - Can also trigger failover
- Same data is shared, so it's the SAME queue manager.



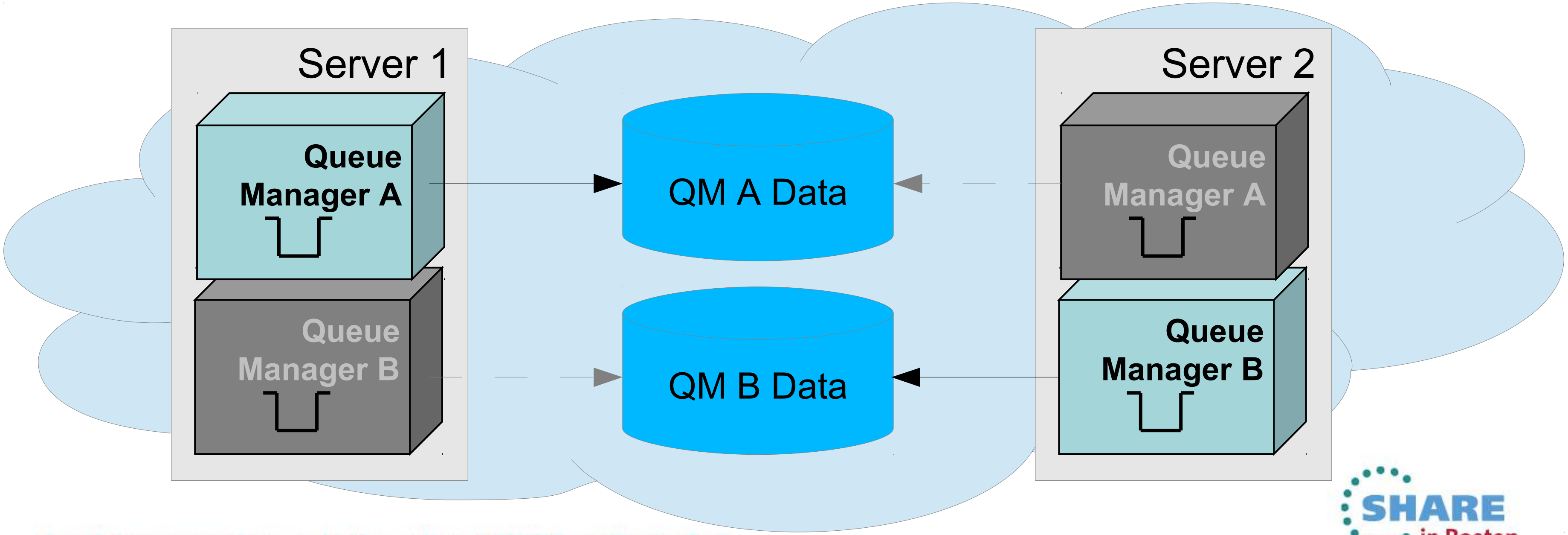
Multi-instance and Java EE

- Connection to Multi-Instance Queue Managers is supported
  - Minimum of WAS 7.0.0.13 required (contains WMQ 7.0.1.3)
  - Can be configured using CCDT
  - Can be configured using Connection Name Lists

- Automatic Client Reconnect ***is not supported*** (in EJB/Web container)
  - MDB based applications use ListenerPort/Activation Specification based reconnection
  - Automatic Client Reconnect can be used in unmanaged/client environments

High Availability continued...

- Active – Active WMQ
  - Multiple active running queue managers
  - Provides service resilience and load balancing
    - *e.g. WMQ Clusters, z/OS Queue Sharing Groups (QSG)*
  - Can have stranded messages problems if queue manager becomes unavailable.
    - *Can use HACMP/Multi-instance queue managers to avoid*



Useful Information: Product Connectivity Scenarios

- Scenario-based instructions for implementing a solution in a business context
- Providing a thorough, complete, and single end-to-end main path from the business need (high level) to the tasks (low level)
- Optional features help you learn as you progress through each task
  - Information about *why* you are instructed to do something
  - Information about *what else* you might do, or want to learn about, related to what you are reading in the main window
- Includes samples that you can try out and adapt

http://publib.boulder.ibm.com/infocenter/prodconn/v1r0m0/index.jsp

Further Information

- WAS product information : http://www-306.ibm.com/software/webservers/appserv/was/
  - WAS Information Centers :
    6.0 http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp
    6.1 http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp
    7.0 http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp
    8.0 http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp
    8.5 http://publib.boulder.ibm.com/infocenter/wasinfo/v8r5/index.jsp
  - Product Connectivity Information Center
    http://publib.boulder.ibm.com/infocenter/prodconn/v1r0m0/index.jsp
  - Using WebSphere MQ Java Interfaces in J2EE/JEE Environments
    http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg21266535
  - IBM developerWorks : http://www.ibm.com/developerworks

    (Searching on "Service Integration Bus" returns a number of interesting articles)

    - *http://www.ibm.com/developerworks/websphere/techjournal/0901_leming/0901_leming.html - WASV7*
    - *http://www.ibm.com/developerworks/websphere/techjournal/0601_ratnasinghe/0601_ratnasinghe.html - Security*
    - *http://www.ibm.com/developerworks/websphere/techjournal/0601_smithson/0601_smithson.html - Security*
  - IBM RedBooks : http://www.redbooks.ibm.com
    - *WebSphere Application Server V7: Messaging Administration Guide SG24-7770-00*
    - *WebSphere Application Server V7: Concepts, Planning and Design, SG24-7708-00*
    - *WebSphere Application Server V7: Technical Overview, REDP-4482-00*
    - *WebSphere Application Server V6.1: JMS Problem Determination, REDP-4330-00*
    - *WebSphere Application Server V6.1: System Management & Configuration, SG24-7304-00*
    - *WebSphere Application Server V6 Scalability and Performance Handbook, SG24-6392-00*
    - *WebSphere Application Server V6.1 Security Handbook, SG24-6316-01*
    - *WebSphere Application Server V6.1: Technical Overview, REDP-4191-00*
    - *WebSphere Application Server V6.1: Planning and Design, SG24-7305-00*
    - *WebSphere Application Server V6.1: Installation Problem Determination, REDP-4305-00*

This was session 13787 - The rest of the week ……

	Monday	Tuesday	Wednesday	Thursday	Friday
08:00				Extending IBM WebSphere MQ and WebSphere Message Broker to the Cloud	CICS and WMQ - The Resurrection of Useful
09:30	Introduction to MQ				Can I Consolidate My Queue Managers and Brokers?
11:00		MQ on z/OS - Vivisection	Hands-on Lab for MQ - take your pick!	MOBILE connectivity with Broker	Migration and Maintenance, the Necessary Evil. Into the Dark for MQ and Message Broker
12:15					
1:30	MQ Parallel Sysplex Exploitation, Getting the Best Availability From MQ on z/OS by Using Shared Queues	What's New in the MQ Family	MQ Clustering - The basics, advances and what's new	**Using IBM WebSphere Application Server and IBM WebSphere MQ Together**	
3:00	First Steps With Message Broker: Application Integration for the Messy	What's New in Message Broker	BIG Connectivity with mobile MQ	WebSphere MQ CHINIT Internals	
4:30	What's available in MQ and Broker for high availability and disaster recovery?	The Dark Side of Monitoring MQ - SMF 115 and 116 Record Reading and Interpretation	MQ & DB2 – MQ Verbs in DB2 & Q-Replication performance	Big Data Sharing with the Cloud - WebSphere eXtreme Scale and IBM Integration Bus Integration	
6:00				WebSphere MQ Channel Authentication Records	

Copyright and Trademarks