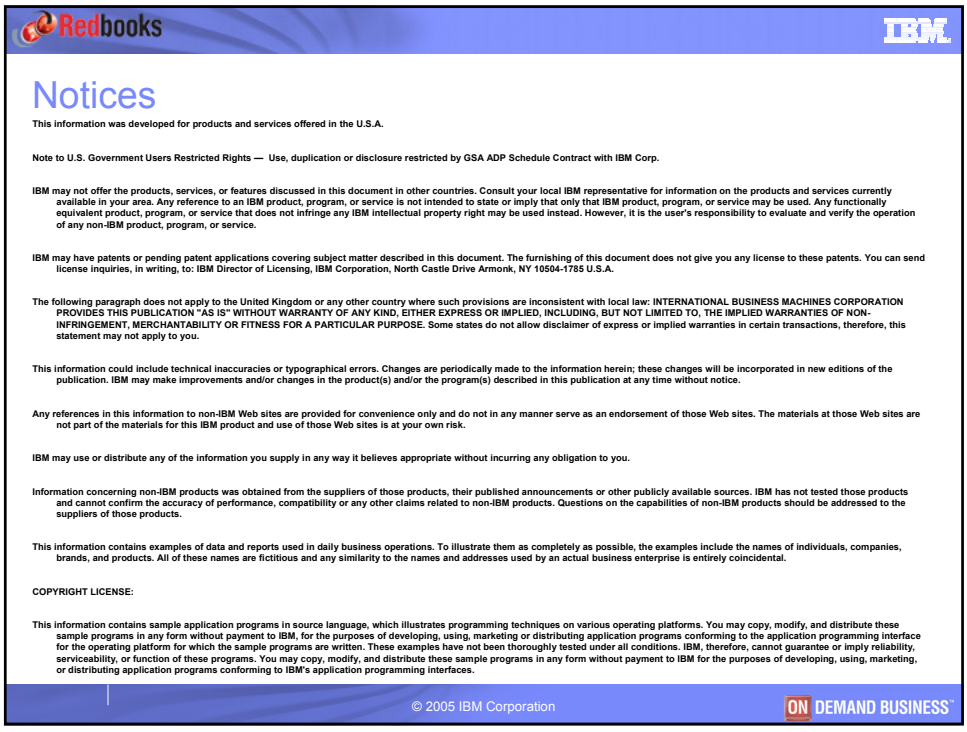

IBM
Redbooks
Managing and Deploying WebSphere applications on z/OS
Connectivity & Integration
ON DEMAND BUSINESS
© 2005 IBM Corporation


Redbooks
IBM
Notices
This information was developed for products and services offered in the U.S.A.
Note to U.S. Government Users Restricted Rights — Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.
IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to: IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.
The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.
This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.
Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.
IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.
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 and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.
COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.
© 2005 IBM Corporation
ON DEMAND BUSINESS

Redbooks IBM

## Trademarks

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

IBM eServer™ Redbooks (logo)™

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ibm.com® | z/OS® | zSeries® | AIX® | ClearCase® | Cloudscape™ | | CICS® | CICSPlex® | DB2 |
| Connect™ | DB2® | DFS™ | DRDA® | Informix® | IBM® | IMS™ | MQSeries® | MVS™ | |
| | Perform™ | Rational® | RACF® | S/390® | SAA® | TME® | VTAM® | WebSphere® | |

The following terms are trademarks of other companies:

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

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

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

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

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

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

 ON DEMAND BUSINESS

Redbooks IBM

## Managing and deploying WebSphere on z/OS

- **WebSphere Version 6 introduction**
- **Connectivity and integration**
- **End to end security solutions**
- **Designing and deploying High Avaialability solutions**
- **Problem determination**

 ON DEMAND BUSINESS

Redbooks IBM

## Agenda: connectivity & integration (1)

- **Module 1: Architectural choices**
- **Module 2: Messaging**
  - Java Message Service (JMS)
  - WebSphere default messaging
  - WebSphere Message Broker
- **Module 3: Data access**
  - Java Database Connectivity (JDBC)
  - SQLJ
  - IMS using JDBC

 ON DEMAND BUSINESS

Redbooks IBM

## Agenda: connectivity & integration (2)

- **Module 4: Web Services**
  - Overview Web services
  - What's new in V6
  - Support in CICS
  - Scenarios
    - IMS WS client to WAS
    - CICS client to WAS
    - CICS provider to WAS
    - DB2 client to WAS
    - DB2 provider to WAS

 ON DEMAND BUSINESS

Redbooks IBM

## Agenda: connectivity & integration (3)

- **Module 5: J2EE Connector Architecture**
  - Overview
  - J2CA using CICSTransaction gateway
  - J2CA using IMS Connect
- **Module 6: J2EE connectivity using RMI over IIOP**

 ON DEMAND BUSINESS

Redbooks IBM

# Module 1

# Architectural choices

 ON DEMAND BUSINESS

Redbooks IBM

## Key Themes affecting Integration

- **Service Oriented Architecture (SOA)**
  - An approach to building distributed systems that delivers application functionality as services to end-user applications or to other services
- **Enterprise Service Bus (ESB)**
  - a common way to let services interact with each other, independent on where they are
- **Business Process Management (BPM)**
  - Modelling, running, monitoring, analyzing and adapting a business process on a continuous base. A business process is preferably executed using workflow concepts and technology.

 ON DEMAND BUSINESS

Redbooks IBM

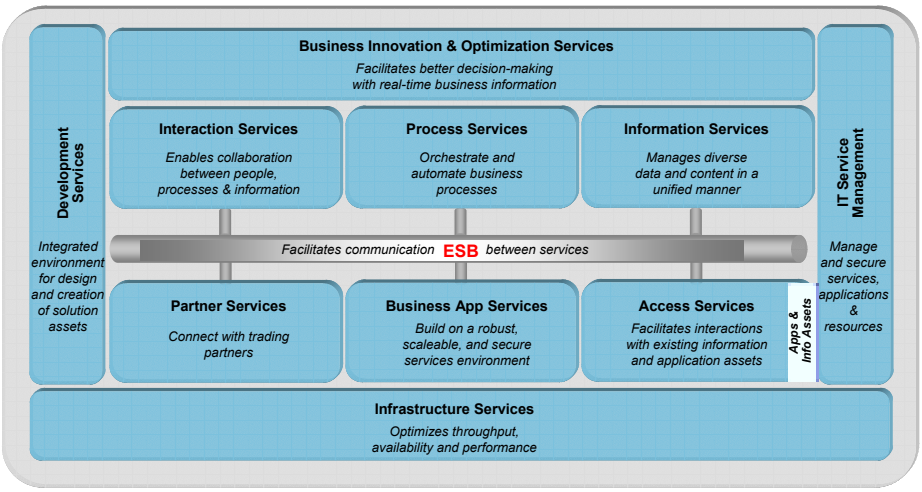
## Service Oriented Architecture Q&A

- **A SOA promotes flexibility via clear definition and loose coupling**
- **What is a service?**
  - A service is defined with explicit interfaces, independent of service implementations
- **How do services interact?**
  - may be invoked by clients inside and outside the enterprise
  - may interact with each other, invoking operations and exchanging data
- **What is service choreography?**
  - Choreography enables groups of services to represent a business process
- **What is service discovery?**
  - A registry of services can exist allowing services to be discovered at build time or runtime
- **How is it enabled?**
  - Exploits new, open standards and XML data definitions (Web Services)

 ON DEMAND BUSINESS

Redbooks IBM

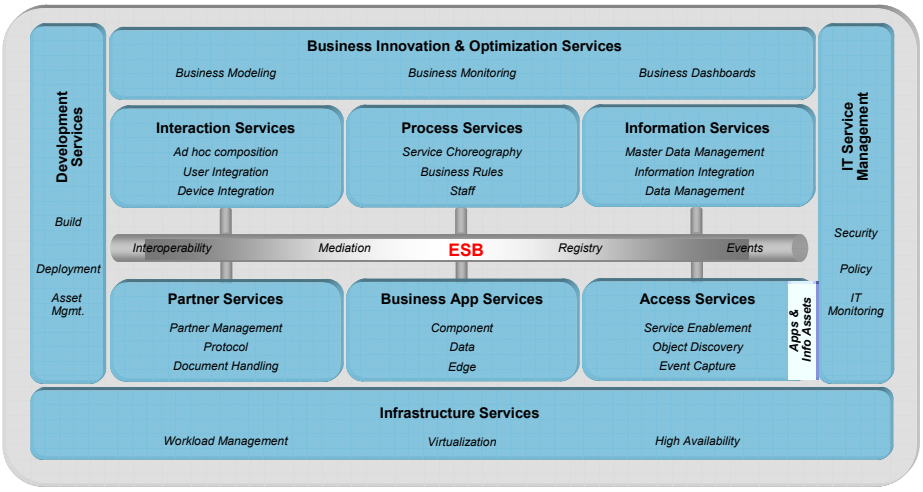
## SOA Reference Architecture




 ON DEMAND BUSINESS

Redbooks IBM

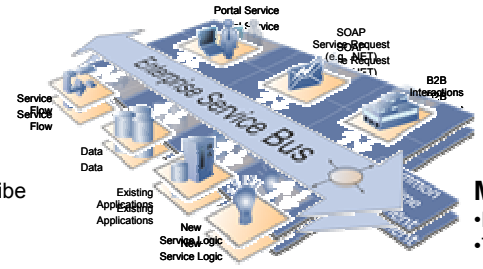
## SOA Reference Architecture




 ON DEMAND BUSINESS

Redbooks IBM

## The Enterprise Service Bus




**Event Services**
•Publish and Subscribe

**Mediation Services**
•Routing
•Transformation

**Transport Services**
•Synchronous/Asynchronous
•Persistent/Non-persistent
•Loosely-coupled/Tightly-coupled

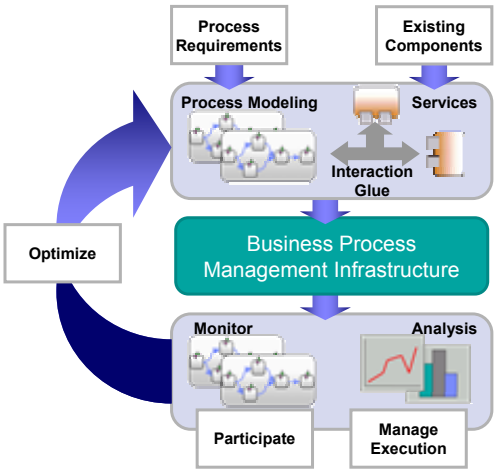
- **… with unsurpassed technical characteristics …**
  - Multiple standards – JAX-RPC, JMS, WebSphere MQ, Web Services …
  - Scale to match performance and throughput needs
  - Security

 ON DEMAND BUSINESS

Redbooks IBM

## Business Process Management

- **Business Process Management**
  - Improves quality in the execution of a business process
  - Includes modelling, executing, monitoring, analyzing and adapting




 ON DEMAND BUSINESS

Redbooks IBM

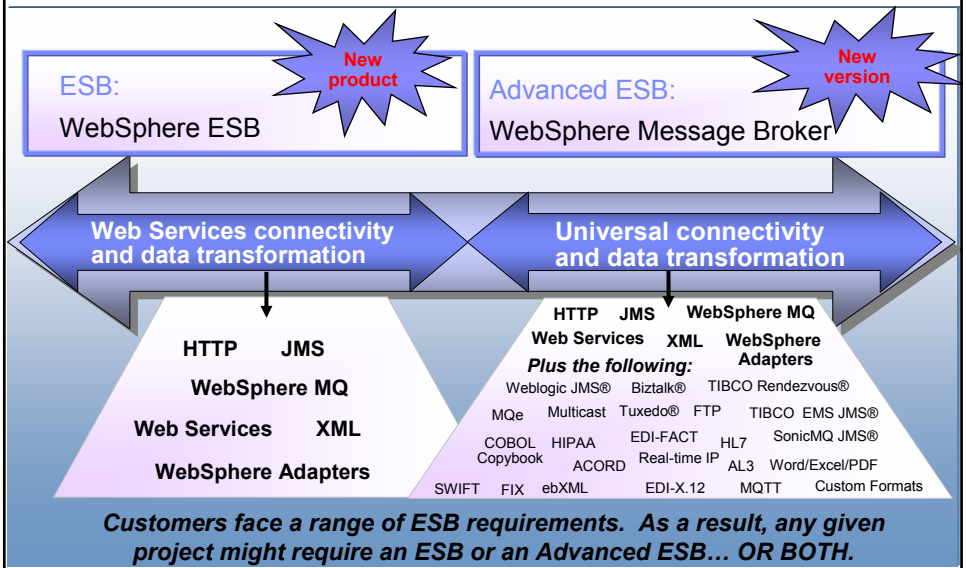
## Key products on z/OS

- **WebSphere Application Server V6.0x for z/OS**
  - Available now
    - WebSphere Base, Network Deployment
    - Extensions
      - WBI Server Foundation
      - WebSphere Portal Server
  - Coming
    - WebSphere Extended Deployment (XD)
    - WebSphere ESB
    - WebSphere Process Server V6
- **WBI Message Broker V6**

 ON DEMAND BUSINESS

Redbooks IBM

## WebSphere ESB and WebSphere Message Broker




***Customers face a range of ESB requirements. As a result, any given project might require an ESB or an Advanced ESB… OR BOTH.***

 ON DEMAND BUSINESS

Redbooks IBM

## Application Integration with WAS on z/OS

- **Synchronous**
  - J2EE Connector Architecture (J2CA)
  - J2EE RMI/IIOP
  - MQSeries or JMS (“pseudo” synchronous)
  - Native, e.g. APPC, TCP/IP…
- **Asynchronous**
  - HTTP
  - WebSphere MQ
    - Point to point
  - WebSphere BI Message Broker
    - Pub/sub
    - Brokering

 ON DEMAND BUSINESS

Redbooks IBM

## The synchronous/asynchronous discussion concluded

- **Synchronous from an *application* perspective means that the sending component waits for a reply back from receiving component until it continues again. This can be achieved with:**
  - MQSeries using request/reply
  - J2CA and RMI/IIOP
  - HTTP
- **Synchronous from a *QoS* perspective means that the sending applications and all its counterparts taking place in one global transaction will all roll back upon failure or commit upon success. This can only be achieved by using:**
  - Certain configurations of J2CA
  - RMI/IIOP

 ON DEMAND BUSINESS

Redbooks IBM

## SOA and Transactions

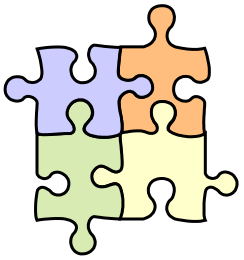
- **A SOA presumes loose coupling of application functions**
- **The Web services standards propagated to be used to implement a SOA use asynchronous communication protocols by default:**
  - HTTP, MQ/JMS
- **But, additional standards are in the works to make Web services behave synchronously**
  - They are not widely supported/implemented yet

 ON DEMAND BUSINESS

Redbooks IBM

## Connecting to EIS and DBMS from WebSphere on z/OS

- **Many choices to connect WebSphere to EIS's and DBs:**
  - Local vs. remote
  - Synchronous vs. asynchronous
- **Different connectivity attributes:**
  - Performance
  - Availability
  - Security
  - Transactionality
  - Scalability
  - Other issues, drawbacks
- **What to choose under which circumstances ?**
- **Architectural perspectives and practices.**

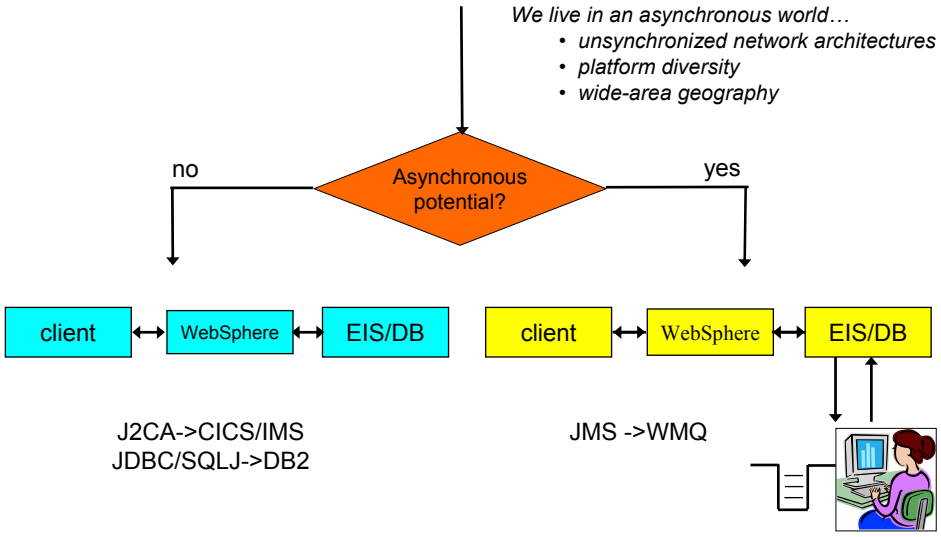


 ON DEMAND BUSINESS

Redbooks IBM

## Synchronous vs. asynchronous options

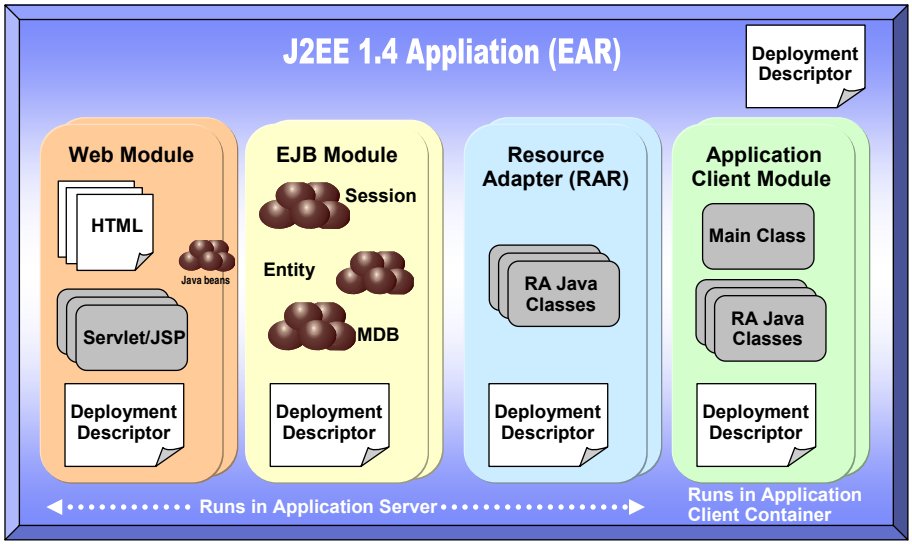
*We live in an asynchronous world…*
- *unsynchronized network architectures*
- *platform diversity*
- *wide-area geography*




 ON DEMAND BUSINESS

Redbooks IBM

## J2EE 1.4 Application Components




 ON DEMAND BUSINESS

Redbooks IBM

## J2EE and Connectivity

- **Java 2 Platform Enterprise Edition (J2EE) defines different ways of accessing back ends**
  - Java DataBase Connectivity (JDBC)
    - for relational databases
    - well, not entirely true, also IMS DB can be accessed through JDBC APIs
  - Java Message Service (JMS)
    - for message oriented middleware
  - J2EE Connector Architecture (J2CA)
    - Transaction managers
    - ERP
    - Any other EIS or legacy system providing a Resource Adapter (RA) for J2CA

 ON DEMAND BUSINESS

Redbooks IBM

## Facts about Transactions

- A ***Local transaction*** is when there is one resource manager controlling all the changes made within the scope of a transaction.
- A ***Global transaction*** is when multiple resource managers are involved in the scope of a transaction.
- ***2-PC*** (2-Phase Commit) is a series of actions which ensures that either all changes made to multiple resources managers are committed or all changes made to multiple resources managers are rolled back. The first phase tells each resource manager to prepare to commit. When positive responses have been received from each resource manger, the second phase is to tell them to commit the changes. The alternative is ***1-PC*** where there is no prepare phase and therefore no opportunity to recover if a failure occurs during the commit process.
- ***XA*** refers to the X/Open distributed transaction processing (DTP) open standard implementation of 2-PC processing for the coordinatopn of changes to multiple distributed relational databases and other resource managers which support the XA interface.
- ***Last Participant Support (LPS)*** refers to the scenario where a transaction manager coordinates a global transaction involving any number of 2-PC capable resource managers and a single 1-PC capable resource manager. The transaction manager uses 2-PC protocol to prepare all of the 2-PC capable resources and when these prepares are successful, the 1-PC resource is called to commit its updates. Depending on the outcome of the 1-PC operation, the 2-PC resources are then committed or rolled back.

 ON DEMAND BUSINESS

Redbooks IBM

## Transaction Management - 2 Phase Commit (2-PC)

- **IMS Connect supports 2-PC in a global transaction in both a local and remote connection**
  - Local - WebSphere for z/OS "manages" request using RRS
  - Remote - IMS Connect "manages" request using RRS, WebSphere for z/OS maintains an interest
- **CTG supports 2-PC in global transaction when the connection is "local"**
  - WebSphere for z/OS "manages" request using RRS
  - CICS "manages" its resources using RRS
- **CTG supports 1-PC in a global transaction in a remote connection only with other 2-PC RRS-enabled resources managers**
  - CTG participates as the Last Participant or Last Agent in the global transaction
  - WebSphere will "prepare" the other resources managers and then commit CICS resources
    - WebSphere will commit/rollback the other resources managers changes depending on the response from CICS
    - If no response is received, WebSphere will issue rollback to the other participating Resource Managers.
      - Note, if CICS successfully committed but was unable to inform WebSphere, a heuristic hazard situation has occurred

 ON DEMAND BUSINESS

Redbooks IBM

## High Availability Recommendations

- **Advantages of "Remote" Connections**
  - Eliminates a Single Point of Failure with “local" connection
    - If the "local" CICS or IMS Connect is unavailable
  - Allows the use of Dynamic Virtual IP Address (DVIPA)
    - Which eliminates a single point of failure and
    - Enables Sysplex Distributor exploitation of Work Load Manager (WLM)
- **Disadvantages of "Remote" Connections**
  - The loss of 2-PC with CICS
  - More "network" overhead involving the use of the TCP/IP stack

 ON DEMAND BUSINESS

Redbooks IBM

# Module 2

# Messaging

 ON DEMAND BUSINESS

Redbooks IBM

## Agenda

- **Messaging overview for WebSphere**
- **WebSphere Default Messaging**
  - A.k.a. “Platform Messaging”
- **Developing and deploying JMS applications**
- **WBI Message Broker**

 ON DEMAND BUSINESS

Redbooks IBM

## WebSphere and Messaging - Intro

- **WebSphere can both send messages to and receive messages from external MQ Queue Managers**
- **WebSphere V6 has an internal messaging architecture called "WebSphere Default Messaging"**
- **The most current API to be used in Java for messaging is the Java Message Service (JMS) Version 1.1**

 ON DEMAND BUSINESS

Redbooks IBM

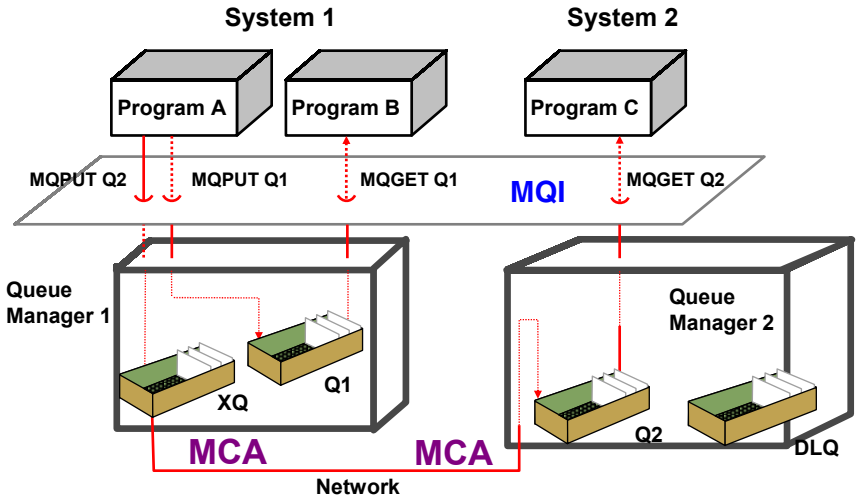
## The Principles of Messaging

- **The principal objective of messaging is to exchange information:**
  - in an asynchronous way
  - offering assured delivery
  - offering failure independence of applications
  - offering a triggering mechanism on the reception of messages
  - using various qualities of service such as message persistence, transactional integrity, and security
  - using a common set of APIs, independent of platform or network complexity

 ON DEMAND BUSINESS

Redbooks IBM

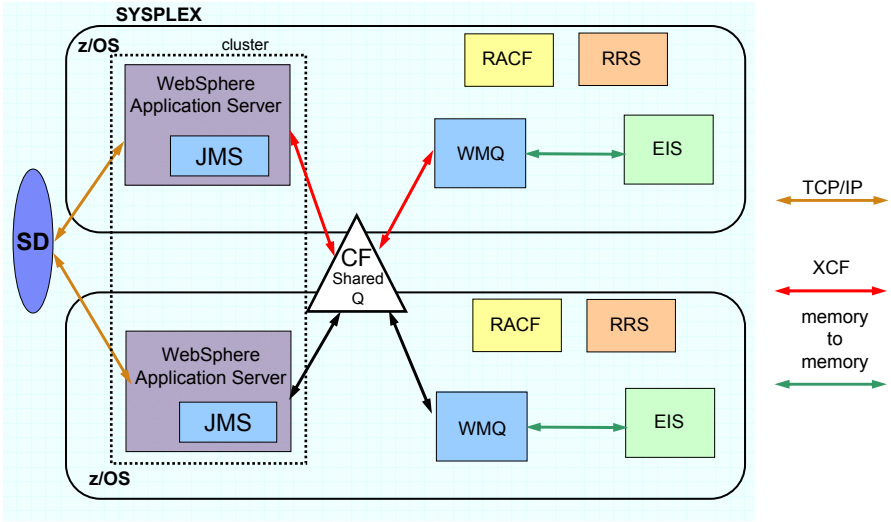
## MQ Overview





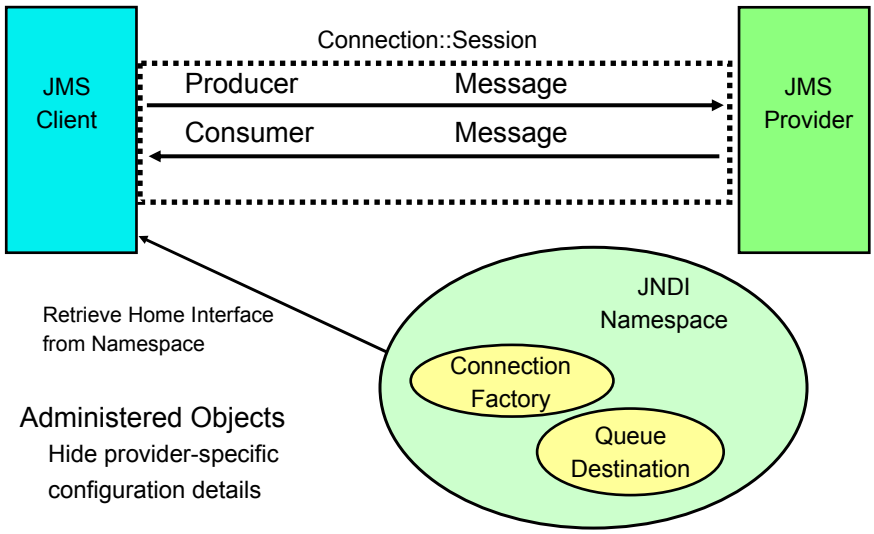

Redbooks IBM

## High Availability on z/OS with WebSphere MQ



Redbooks IBM

## JMS 1.1 Topology




 ON DEMAND BUSINESS

Redbooks IBM

## JMS Terminology

- **Point-to-point domain:**
  - Connection Factory
    - Generates the connection
    - Holds the properties of the target queue manager
  - Destination
    - Holds the properties of the target queue
- **Publish-subscribe domain:**
  - Topic Connection Factory
  - Topic Destination
  - Broker required!

 ON DEMAND BUSINESS

Redbooks IBM

## The new JMS 1.1 Specification- What changed?

- **Terminology…**
  - Different names for objects and connection factories
- **Unification of Point to point and pub/sub domains**
  - Makes it possible to use both P to P and P/S in the same application
  - One connection factory can be used for both domains
- **No structural changes in programming or deployment**
- **JMS 1.0 applications (still) run in WebSphere V6 with all supported JMS Providers**


ON DEMAND BUSINESS

Redbooks IBM

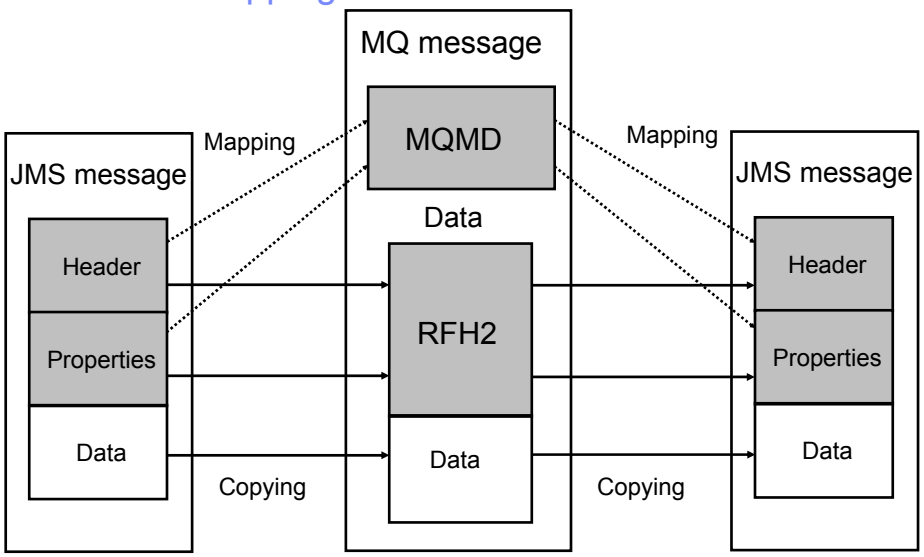
## JMS 1.02b vs. JMS 1.1 objects

| JMS 1.02b | JMS 1.1 |
|---|---|
| QueueConnectionFactory, TopicConnectionFactory | ConnectionFactory |
| Queue, Topic | Destination |
| QueueConnection, TopicConnection | Connection |
| QueueSession, TopicSession | Session |
| QueueSender, TopicPublisher | MessageProducer |
| QueueReceiver, TopicSubscriber | MessageConsumer |


ON DEMAND BUSINESS

Redbooks IBM

## MQ – JMS Mapping




 ON DEMAND BUSINESS

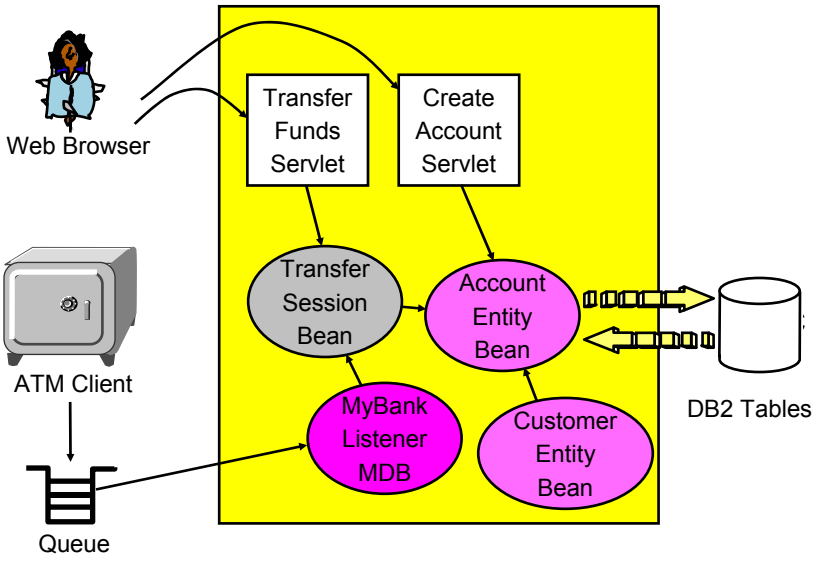
Redbooks IBM

## Four JMS Providers in WebSphere V6

- **WebSphere-supplied JMS providers**
  - WebSphere Default Messaging
    - Only in V6.01
  - V5 Default Messaging
    - Deprecated in V6, but still usable
- **External JMS providers**
  - WebSphere MQ
    - Shipped with WebSphere MQ product
  - Generic
    - Can be any third party JMS-compliant product

 ON DEMAND BUSINESS

## The concept of a Message-Driven Bean



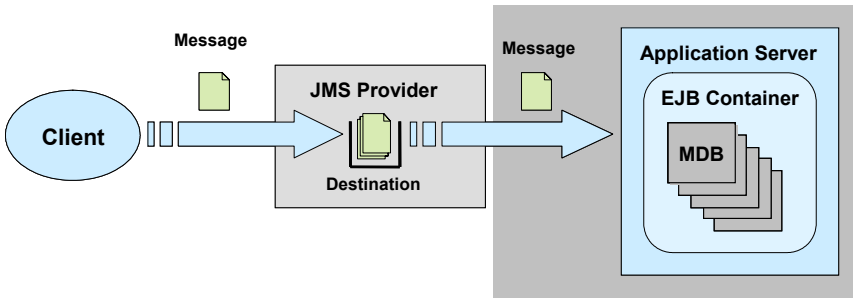



## Message-Driven Beans (MDBs) - Artifacts

- **Introduced in EJB 2.1 specification**
- **An MDB is triggered upon arrival of an incoming message, by implementing a:**
  - *Message Listener Service* for WebSphere MQ and V5 default messaging
  - *Activation Specification* for V6 default messaging
    - based on the JCA 1.5 specification
- **MDBs support both the point-to-point and publish/subscribe domains**
- **An MDB is modelled after a stateless EJB**
- **MDBs do not expose their home interfaces**
- **The messages they process contain no client credentials**
- **They preserve transactional integrity**

Redbooks IBM

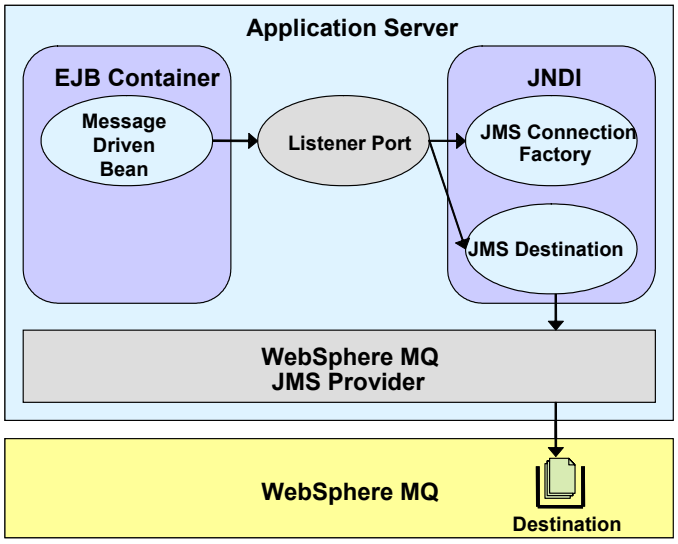
## MDB triggering concept



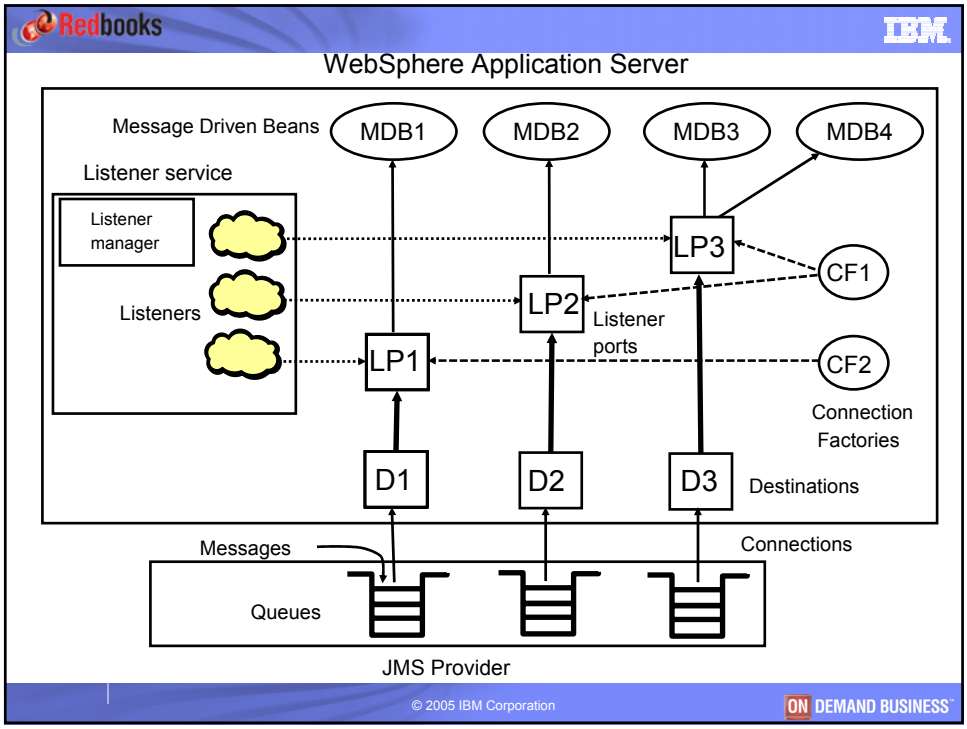

 ON DEMAND BUSINESS

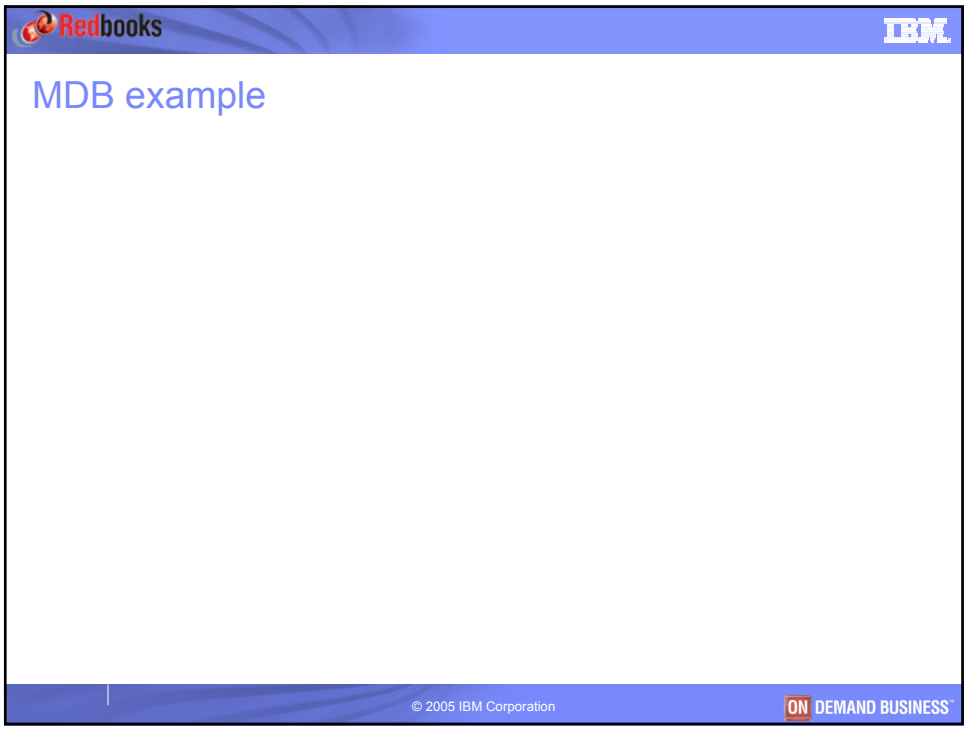
Redbooks IBM

## Basic Listener Port Configuration




 ON DEMAND BUSINESS


Redbooks
IBM
WebSphere Application Server
Message Driven Beans
MDB1
MDB2
MDB3
MDB4
Listener service
Listener manager
Listeners
LP1
LP2
LP3
Listener ports
CF1
CF2
Connection Factories
D1
D2
D3
Destinations
Messages
Connections
Queues
JMS Provider
© 2005 IBM Corporation
ON DEMAND BUSINESS


Redbooks
IBM
MDB example
© 2005 IBM Corporation
ON DEMAND BUSINESS

Redbooks IBM

## Message Driven Beans – Programming Directives

- **Delegate business logic to another EJB to provide clear separation of message handling and business processing**
- **Minimize message processing time for the MDB**
- **Avoid large message bodies as much as possible**
- **Do not maintain any conversational state in the MDB**
- **Avoid dependencies on the order in which message arrive**
- **Prevent to have "poison" messages exhausting resources as they are rolled back to immediately reinvoke an MDB**

 ON DEMAND BUSINESS

Redbooks IBM

## Agenda

- **Messaging overview for WebSphere**
- **WebSphere Default Messaging**
  - A.k.a. "Platform Messaging"
- **Developing and deploying JMS applications**
- **WBI Message Broker**

 ON DEMAND BUSINESS

Redbooks IBM

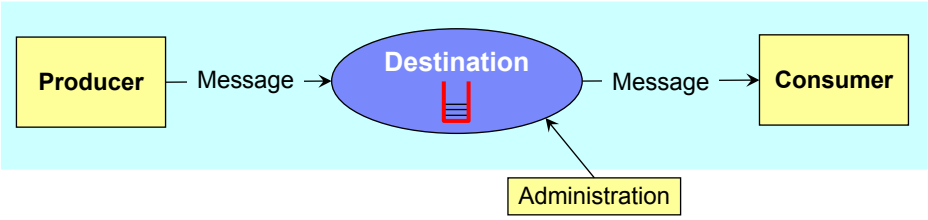
## WebSphere Default Messaging

- **Introduced in Version 6.01**
- **Based on the "Service Integration Bus (SIB)" concept**
- **Built inside WebSphere**
- **Can be used to exchange messages between applications within the same server and/or cell or between application in different servers/cells.**
- **Applications use JMS API**
- **A Service Integration Bus can be linked to an external MQ network**

 ON DEMAND BUSINESS

Redbooks IBM

## Messaging – Basic Flow




- **Producers send/put messages to destinations**
- **Consumers receive/get messages from destinations**
- **Destinations are Default Messaging managed points of communication rendezvous**
  - Example: JMS Queues, Topics, Web Service endpoints

 ON DEMAND BUSINESS

Redbooks IBM

## Service Integration Bus (SIB)

- **The SIB is actually the WebSphere implementation of the ESB**
- **A SIB cannot exist beyond the cell boundary**
- **There can be more than one SIB in one cell**
  - A common pattern is to have one SIB in a stand-alone single server
- **Application servers can be a *member* of multiple SIB's**
  - If application servers are clustered, the cluster becomes the member
- **SIB's can be interconnected**
- **The SIB can also flow Web services requests**
- **A SIB can be connected to an MQ Queue Manager using an *MQ Link***

 ON DEMAND BUSINESS

Redbooks IBM

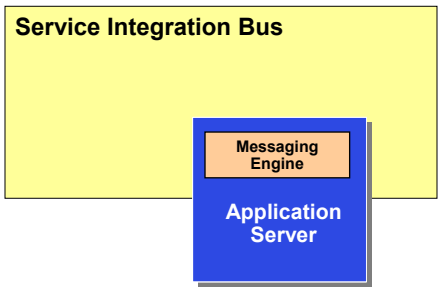
## Messaging Engine (ME)

- **Represents the JMS provider and is equivalent to an MQ Queue Manager**
- **Runs in a separate address space, called the *Control Region Adjunct (CRA)***
  - Is started automatically when an ME is started
  - An ME will be started for each application server that is a member of the SIB
    - There is one-to-one relationship between application server and ME
- **Is the connection point for clients consuming and producing messages**
- **Each ME has its own set of tables in a data store (JDBC database)**

 ON DEMAND BUSINESS

Redbooks IBM

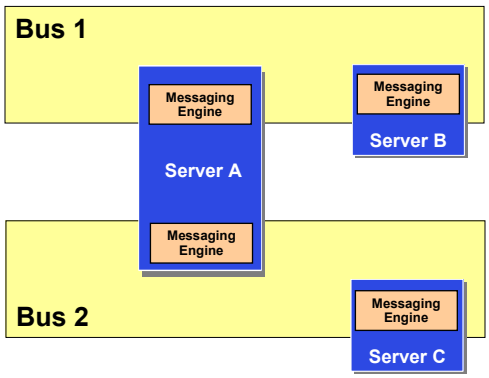
## Service Integration Bus (SIB) with one Messaging Engine




 ON DEMAND BUSINESS

Redbooks IBM

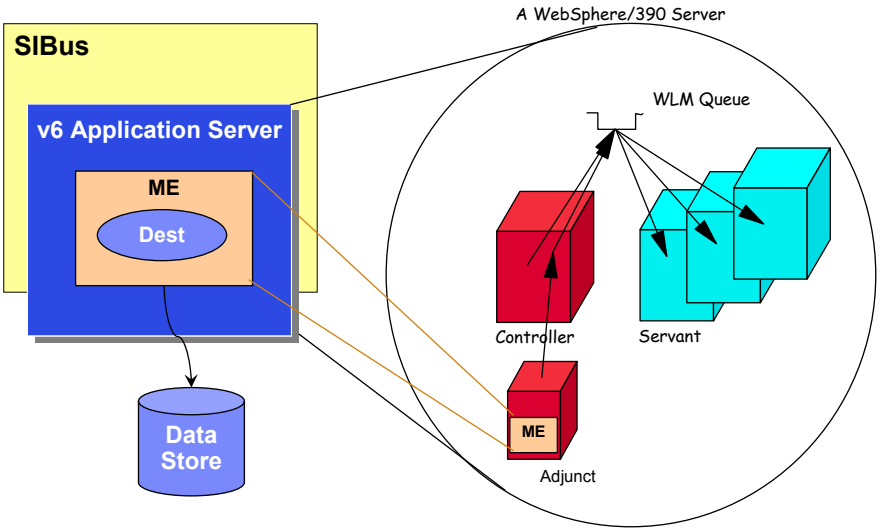
## Two SIB's with multiple Messaging Engines




 ON DEMAND BUSINESS

Redbooks IBM

## Messaging Engine and Application Server on z/OS




 ON DEMAND BUSINESS

Redbooks IBM

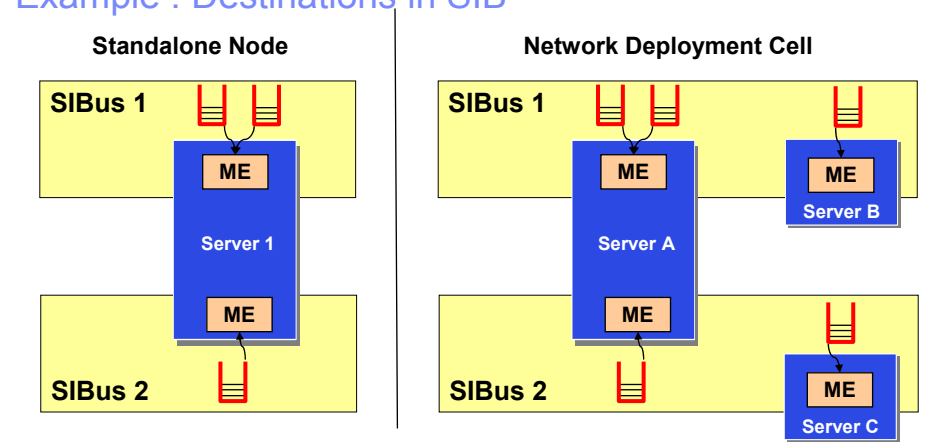
## Bus Destination

- **Endpoint for applications to send messages to or receive messages from**
- **Can be permanent or temporary**
  - Temporary - Created and deleted automatically for API specific destinations
    - created programmatically usually to specify a JMSReplyTo destination within a message
  - Permanent – Created by Administrator
    - Deleted only when Administrator deletes it.
- **Local or on a foreign bus**
- **Destination types**
  - Queue
    - Used for point-to-point messaging
  - Topic space
    - Used for publish/subscribe messaging
  - Alias
    - An alternate name for a destination; the actual destination that the alias maps to can either be on the local bus or on a foreign bus
  - Foreign
    - Used to identify a destination on a foreign bus
  - Exception
    - Destination that is used to handle messages that cannot be sent to intended bus destination

 ON DEMAND BUSINESS

Redbooks IBM

## Example : Destinations in SIB




- A Webphere Application Server standalone node can have multiple **buses**
- Each bus can have servers as **bus members**
- When a server is made a bus member, **Messaging Engine** is created
- **Destinations** are created on the bus and hosted on a Messaging Engine

 ON DEMAND BUSINESS

Redbooks IBM

## Destination Quality of Service for Reliability

Low

Reliability

High

- **BEST_EFFORT_NONPERSISTENT**
  - Messages are never written to disk
  - throw away messages if memory cache over-runs
- **EXPRESS_NONPERSISTENT**
  - Messages are written asynchronously to persistent storage if memory cache overruns, but are not kept over server restarts
  - No acknowledgement that the ME has received the message
- **RELIABLE_NONPERSISTENT**
  - Same as Express_Nonpersistent, except, we have a low level acknowledgement message that the client code waits for, before returning to the application with an OK or not OK response
- **RELIABLE_PERSISTENT**
  - Messages are written asynchronously to persistent storage during normal processing, and stay persisted over server restarts.
  - If the server fails, messages are lost if they are held in the cache at the time of failure.
- **ASSURED_PERSISTENT**
  - Highest degree of reliability where assured delivery is supported

High

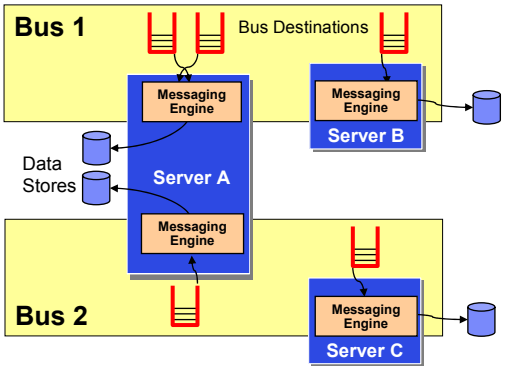
Performance

Low

 ON DEMAND BUSINESS

Redbooks IBM

## Messaging Engines with persistent datastores



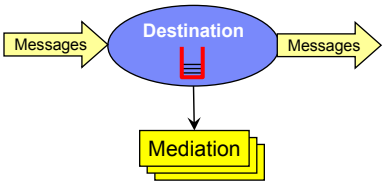

- **ME requires persistent backing data store – JDBC database used in WebSphere implementation**
- **MEs may share the database, but each ME has its own schema within the database (which results in different tables)**

 ON DEMAND BUSINESS

Redbooks IBM

## Mediation

- **Mediation - the ability to manipulate a message as it traverses the messaging bus (destination)**
  - Transform the message
  - Reroute the message to a different destination (or sequence of destinations)
  - Copy and route the message to additional destinations
  - Allow interaction with non-messaging resource managers (e.g. Databases)
- **Mediation attached administratively to a Destination**
- **Mediation construction scenarios:**
  - Built from supplied Mediation subcomponents (Mediation Beans)
    - Subcomponent implementations shipped with WPM
  - Mediation beans supplied by IBM or third-party
    - IBM supplied Mediation beans will come in the future – not in v6




 ON DEMAND BUSINESS

Redbooks IBM

## Default Messaging - Interoperability

- **Full interoperability with other SIB 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**

 ON DEMAND BUSINESS

Redbooks IBM

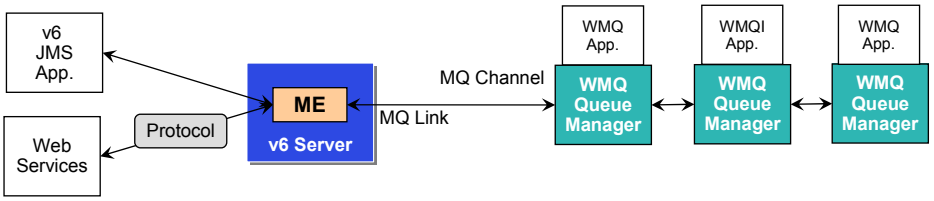
## Relationship to WebSphere MQ

- **A WebSphere MQ Queue Manager and/or a WebSphere MQ Integrator or Event Broker can coexist on the same system 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 the PM bus using any communications protocol other than TCP/IP
  - A PM messaging engine cannot participate in a WebSphere MQ cluster.

 ON DEMAND BUSINESS

Redbooks IBM

## Default Messaging: Interoperability with MQ




- **Tight integration between WebSphere Platform Messaging and WebSphere MQ**
- **WebSphere MQ thinks that the v6 Messaging Engine is another Queue Manager**
- **WebSphere MQ applications can send messages to queues hosted on WAS 6.0 Messaging**
- **WebSphere v6 Messaging apps can send messages to WebSphere MQ queues**

 ON DEMAND BUSINESS

Redbooks IBM

## When use WebSphere DM and when use WebSphere MQ?

- **Use WebSphere Platform Messaging:**
  - For new extensive J2EE apps needing a messaging infrastructure for intra-J2EE application messaging
  - When currently using WAS v5 embedded JMS provider for intra-WAS messaging
- **Use WebSphere MQ:**
  - When J2EE applications need integration with non-J2EE applications
    - ERP, CICS/IMS, DB2, stand-alone Cobol etc.
  - WhenJ2EE applications require specific WebSphere MQ functions

 ON DEMAND BUSINESS

Redbooks IBM

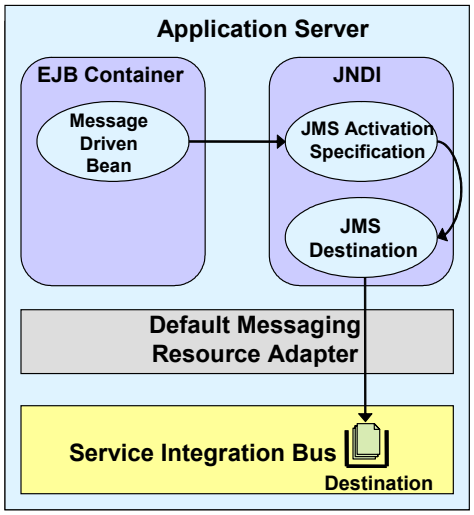
## WebSphere Default Messaging – Pros and Cons

| Pros | Cons |
|---|---|
| It comes shipped with WebSphere Application Server for z/OS, version 6.01 | You need to get familiar with the new concepts |
| It fully supports the JMS 1.1 API specification | It is only available on platforms with WebSphere Application Server, Version 6.01 |
| It uses a single point of administration by the WebSphere administrative console | The Messaging Engine cannot participate in an MQ cluster, although it can connect to a queue manager that is part of a cluster |
| It provides a broker type functionality by the concept of mediation | |
| It can coexist with WebSphere MQ on the same LPAR and be connected to a queue manager via an MQ link | |
| It offers a high granularity of qualities of service regarding message persistence | |
| It includes a fail-over mechanism for Messaging Engines in a cluster | |
| Existing V5 JMS clients can make use of it without any migration changes made to them | |
| Default messaging is a strategic messaging mechanism (as well as WebSphere MQ messaging) focused on the J2EE market | |


ON DEMAND BUSINESS

Redbooks IBM

## MDBs require Activation Specification with DM

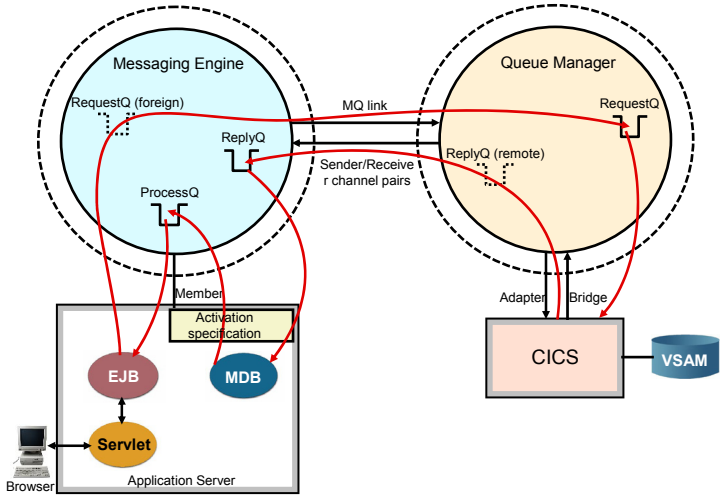




ON DEMAND BUSINESS

Redbooks Local TraderBus Foreign MQ4BBus IBM

## Sample scenario with SIB, WMQ and CICS






Redbooks IBM

## Agenda

- **Messaging overview for WebSphere**
- **WebSphere Default Messaging**
  - A.k.a. “Platform Messaging”
- **Developing and deploying JMS applications**
- **WBI Message Broker**

Redbooks IBM

## Configuration in WebSphere for JMS Applications

- **One time setup of WebSphere MQ in WAS**
- **For each application:**
  - Definitions for MDB's:
    - Listener port(s)
      - In case of WebSphere MQ JMS Provider
    - Activation Specification(s)
      - In case of using Default Messaging JMS Provider
  - Connection factories
  - Destinations

 ON DEMAND BUSINESS

Redbooks IBM

## One time setup of WebSphere MQ in WAS

| | | | |
|---|---|---|---|
| ☐ | MQJMS_LIB_ROOT | ${MQ_INSTALL_ROOT}/java/lib | cells:cl6481:nodes:nd6481 |
| ☐ | MQ_INSTALL_ROOT | /usr/lpp/mqm/V5R3M1 | cells:cl6481:nodes:nd6481 |

- **Authorized and program controlled:**
  - `extattr +ap /usr/lpp/mqm/V5R3M1/java/lib/libwmqjbatch.so`
  - `extattr +ap /usr/lpp/mqm/V5R3M1/java/lib/libwmqjbind.so`
  - `extattr +ap /usr/lpp/mqm/V5R3M1/java/lib/libwmqjrrs.so`
- **STEPLIB additions:**
  - `//STEPLIB DD DISP=SHR,DSN=MQ531.SCSQANLE`
  - `//         DD DISP=SHR,DSN=MQ531.SCSQAUTH`

 ON DEMAND BUSINESS

Redbooks IBM

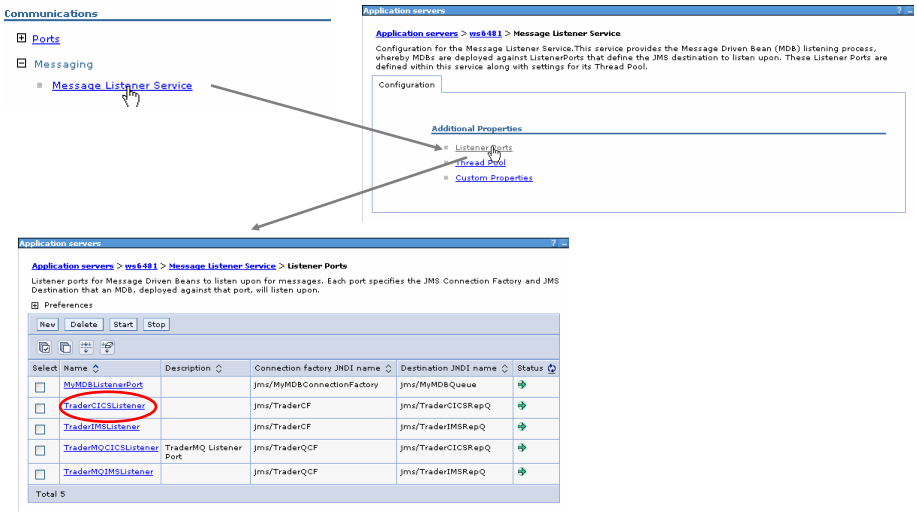
## Configuration – WebSphere MQ

- **Listener ports**
  - In case of using MDBs
- **Connection factories**
  - JMS 1.0 applications
    - WebSphere MQ queue connection factory
      - Point to point
    - WebSphere MQ topic connection factory
      - Pub/sub
  - JMS 1.1 applications
    - WebSphere MQ (unified) connection factory
      - Both point to point and pub/sub
- **Destinations**
  - WebSphere MQ queue destination
    - Point to point
  - WebSphere MQ topic destination
    - Pub/sub

 ON DEMAND BUSINESS

Redbooks IBM

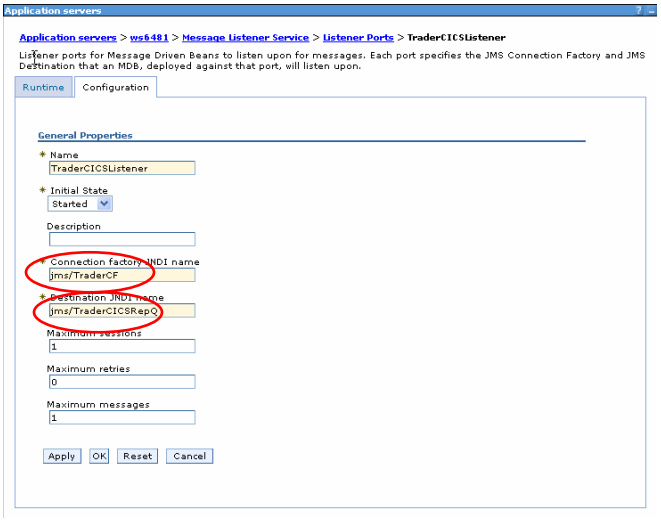
## Listener Ports Definition in WAS V6 – Part 1




 ON DEMAND BUSINESS

Redbooks IBM

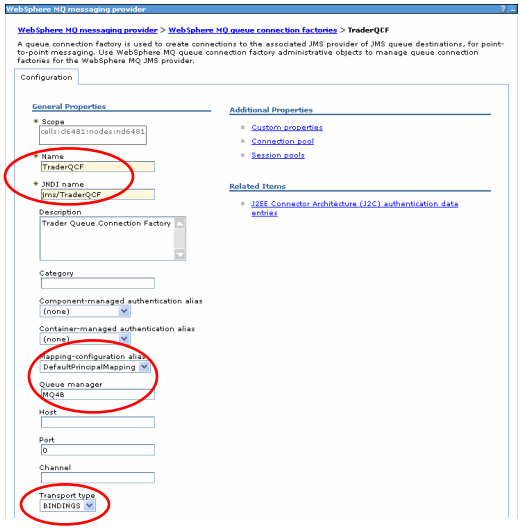
## Listener Ports Definition in WAS V6 – Part 2




 ON DEMAND BUSINESS

Redbooks IBM

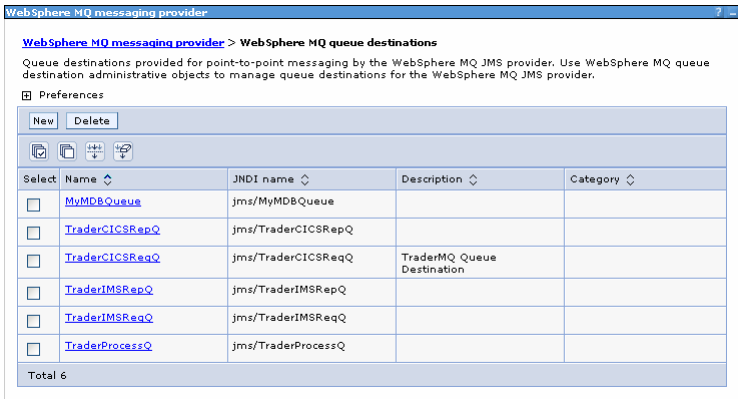
## WebSphere MQ queue connection factory




 ON DEMAND BUSINESS

Redbooks IBM

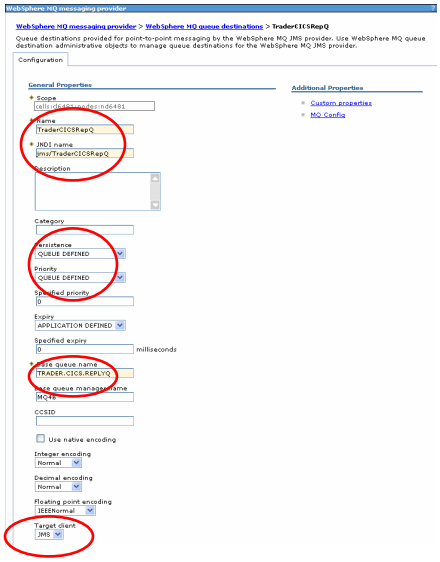
## WebSphere MQ queue destinations – Part 1




 ON DEMAND BUSINESS

Redbooks IBM

## WebSphere MQ queue destinations – Part 2




 ON DEMAND BUSINESS

Redbooks IBM

## Configuration – WebSphere Default Messaging

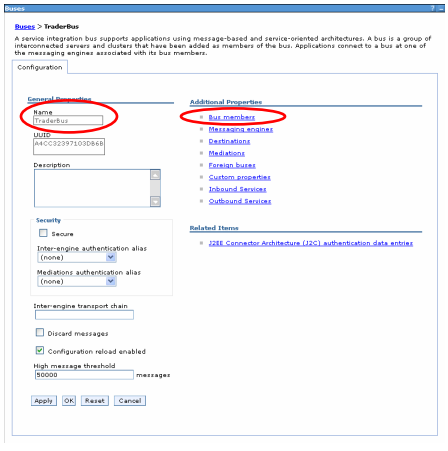
- **Service Integration Bus**
- **Optionally, MQ links or links to other Service Integration Buses**
- **Activation Specifications**
  - In case of using MDBs
- **Connection factories**
  - JMS 1.0 applications
    - JMS queue connection factory
      - Point to point
    - JMS topic connection factory
      - Pub/sub
  - JMS 1.1 applications
    - JMS (unified) connection factory
      - Both point to point and pub/sub
- **Destinations**
  - JMS queue
    - Point to point
  - JMS topic
    - Pub/sub

 ON DEMAND BUSINESS

Redbooks IBM

## Defining a Service Integration Bus in WAS V6

- **Service Integration → Buses → New**

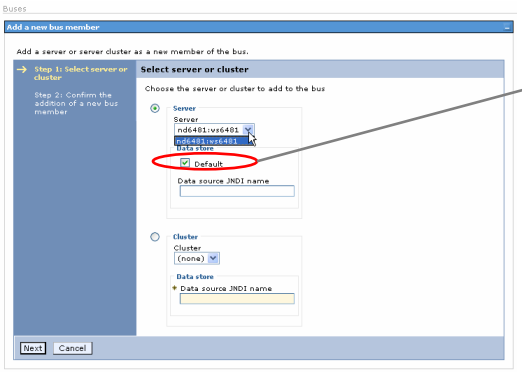



 ON DEMAND BUSINESS

Redbooks IBM

## Add the Application Server as a Member to the Bus

- **Service Integration → Buses → <bus name> → Bus members → Add**

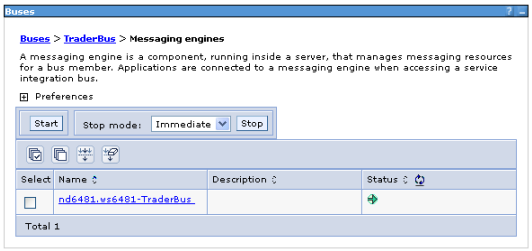



 ON DEMAND BUSINESS

Redbooks IBM

## Messaging Engine

- **When a bus member (= application server) has been added, a Messaging Engine will be created automatically**
- **To display the created ME, click Service integration → Buses → <bus name> → Messaging engines**




 ON DEMAND BUSINESS

Redbooks IBM

## Connecting a SIB in WAS to a WebSphere MQ QMGR

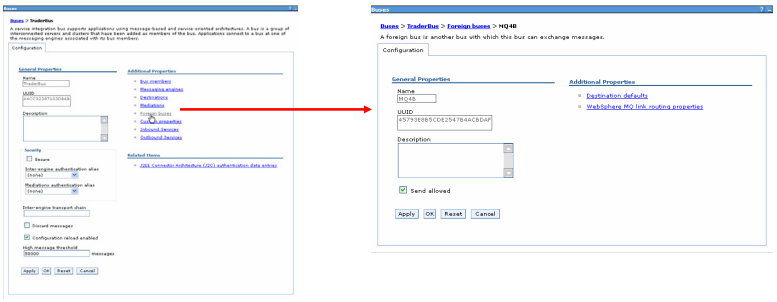
- **If the bus in WebSphere needs to be connected to a WebSphere MQ queue manager**
  - A foreign bus needs to be created in WAS that represents the WebSphere MQ queue manager
  - An MQ link needs to be defined between the SIB in WAS and the WebSphere MQ queue manager
  - A receiver and sender channel need to be defined in both WebSphere and the WebSphere MQ queue manager to communicate with the endpoint of the bus in WebSphere
  - Remote queues need to be added to the WebSphere MQ queue manager to point to the bus in WebSphere

 ON DEMAND BUSINESS

Redbooks IBM

## Creating a Foreign Bus in WebSphere

- **Service Integration → Buses → <bus name> → Foreign buses → New**
- **Recommendation: use the WebSphere MQ qmgr name for the foreign bus**

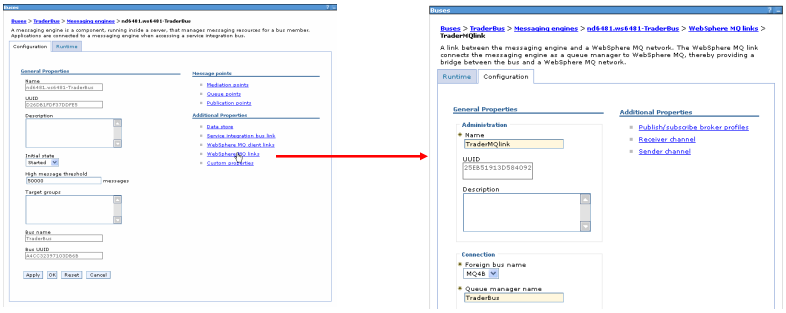


 ON DEMAND BUSINESS

Redbooks IBM

## Creating the MQ link

- **Service integration → Buses → <bus name> → Messaging engines → <engine name> → WebSphere MQ links → New**



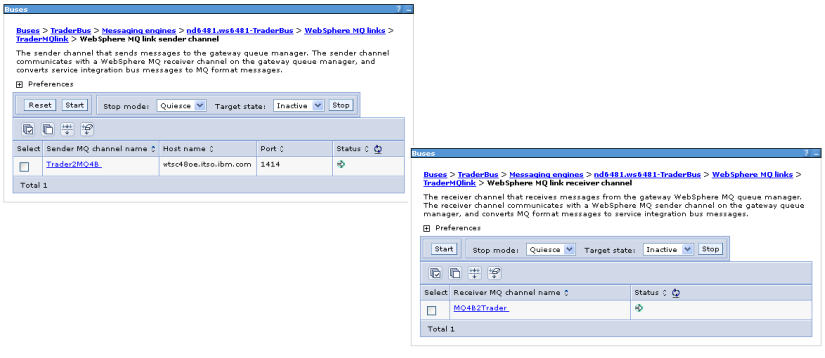


ON DEMAND BUSINESS

Redbooks IBM

## Creating a Receiver and Sender channel in WebSphere

- **WebSphere MQ links → <link name> → Receiver channel or Sender channel**
- **The TCP/IP address and portnumber in the sender must match those of the WebSphere MQ queue manager**





ON DEMAND BUSINESS

Redbooks IBM

## Creating Sender and Receiver in WebSphere MQ

- **The sender channel must be defined as follows:**
  - Transport type: T (TCP/IP)
  - Connection name: TCP/IP address(portnumber)
    - TCP/IP address must match the TCP/IP address of the WAS server
    - Portnumber must match the portnumber specified in SIB_MQ_ENDPOINT_ADDRESS or SIB_MQ_ENDPOINT_SECURE_ADDRESS in WAS
  - Transmission queue: the name of the transmission queue to be used
    - The name of the transmission queue MUST match the transmission queue to be used in the (remote) reply queue
- **The receiver channel can be defined with defaults**

 ON DEMAND BUSINESS

Redbooks IBM

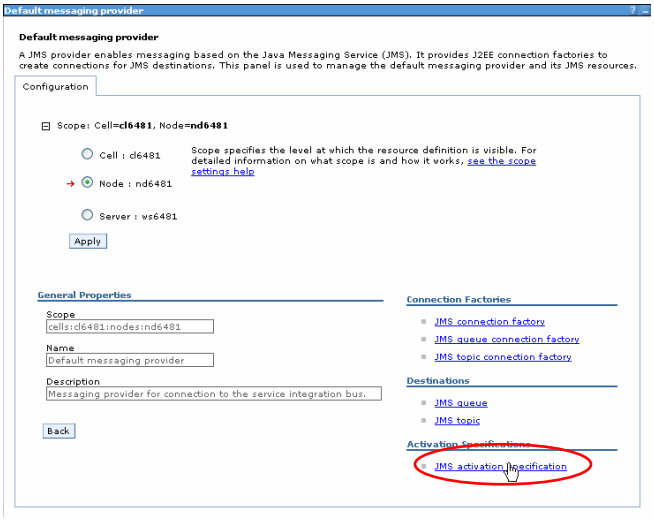
## Defining remote queues in WebSphere MQ

- **The remote queue representing the ME on the bus in WebSphere needs to be defined as follows:**
  - Queue name: name used in the sending application
  - Remote Queue Manager: name of the bus as defined in WebSphere
  - Transmission queue: name of the transmission queue specified in the sender channel

 ON DEMAND BUSINESS

Redbooks IBM

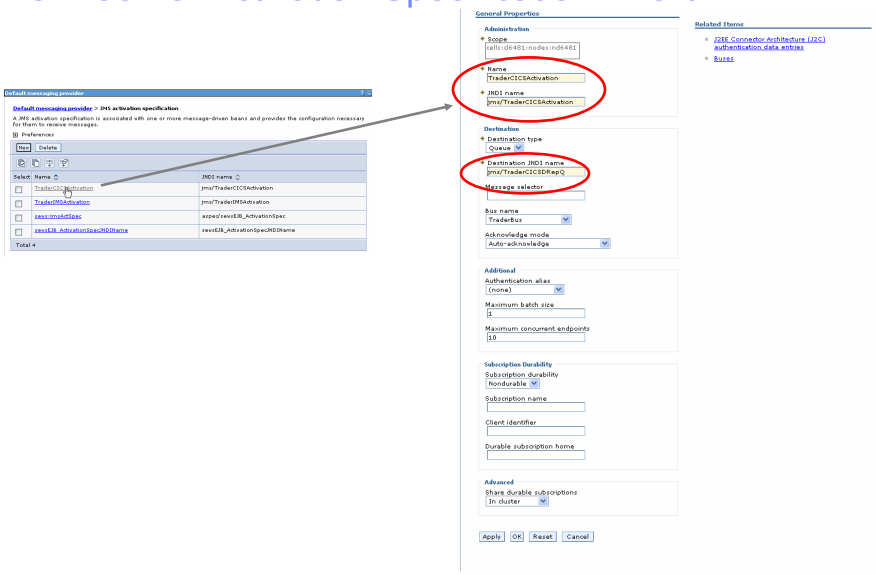
## Definition of Activation Specification - Part 1





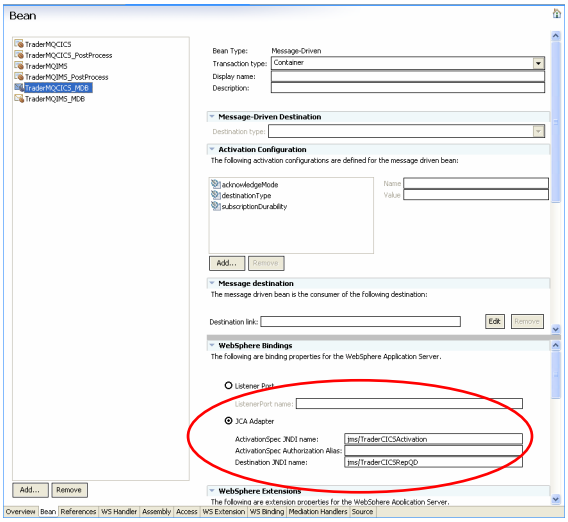

Redbooks IBM

## Definition of Activation Specification - Part 2



Redbooks IBM

## JNDI name for Activation Specification in RAD

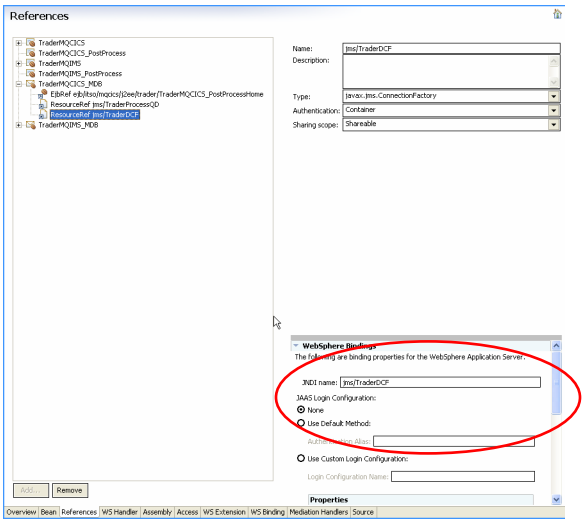





ON DEMAND BUSINESS

Redbooks IBM

## JNDI name for connection factory in RAD



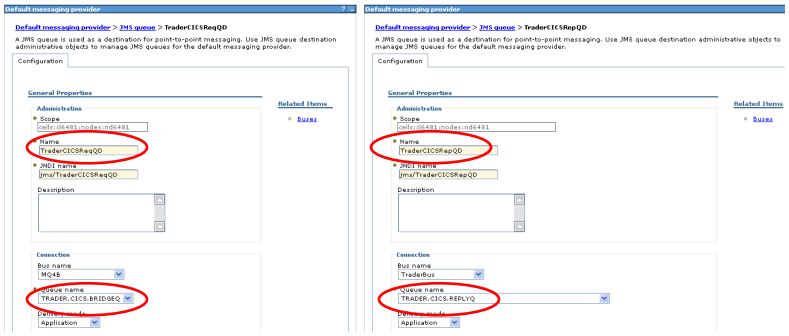



ON DEMAND BUSINESS

Redbooks IBM

## Destinations for WebSphere Default Messaging in WAS

- **You need to specify a JMS queue on the bus for sending and receiving messages**
- **The JMS queues will be linked to the WebSphere MQ queues specified**




 ON DEMAND BUSINESS

Redbooks IBM

## Agenda

- **Messaging overview for WebSphere**
- **WebSphere Default Messaging**
  - A.k.a. “Platform Messaging”
- **Developing and deploying JMS applications**
- **WBI Message Broker**

 ON DEMAND BUSINESS

Redbooks IBM

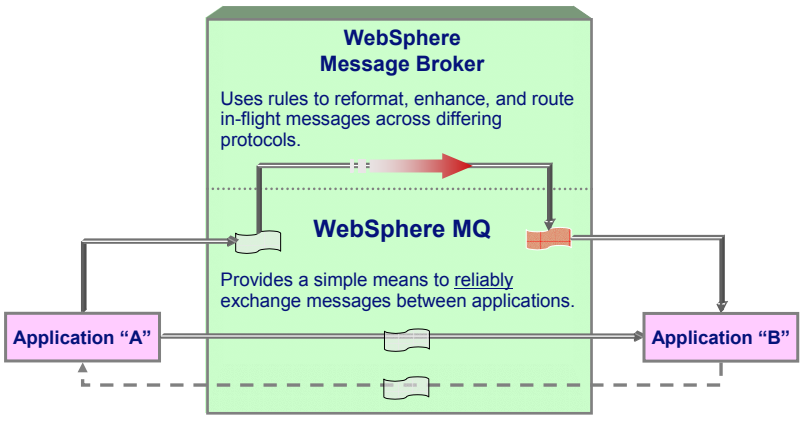
## Agenda

- **Messaging overview for WebSphere**
- **WebSphere Default Messaging**
  - A.k.a. “Platform Messaging”
- **Developing and deploying JMS applications**
- **WBI Message Broker**
  - Overview

 ON DEMAND BUSINESS

Redbooks IBM

## Messaging Overview




Event Notification (1 way communication), Request / Response (2 way communication)

 ON DEMAND BUSINESS

Redbooks IBM

## WBI Message Broker Version 5.0 - Release highlights

- **Tools re-based on WebSphere Studio Workbench**
  - full stand-alone version included in product package
  - will also integrate into existing WSADIE 5.0 workbench
  - supports v2.1 and v5.0 brokers
- **Version control and change management**
  - exploit standard WSWB repository support (CVS, ClearCase, …)
- **Message formatting enhancements**
  - message model re-based on XML Schema
  - full support for XML namespaces
- **Web Services**
  - support for HTTP (SOAP) protocols
  - WSDL generation from message model
- **Flexible protocol support**
  - real-time protocol and IP Multicast support
- **Accounting and statistics**
  - flexible solution for data collection and reporting
  - pub/sub reporting model and support for SMF records on z/OS

 ON DEMAND BUSINESS

Redbooks IBM

## WBI Message Broker Version 5.0 - Release highlights

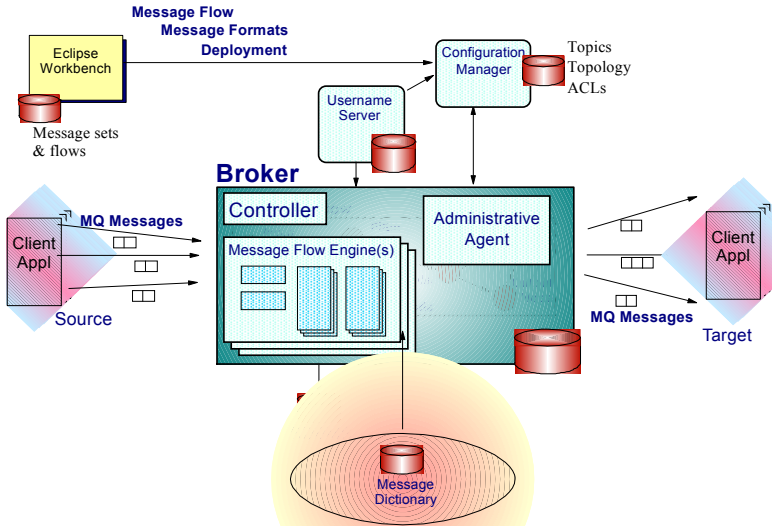
- **Tools re-based on WebSphere Studio Workbench**
  - full stand-alone version included in product package
  - will also integrate into existing WSADIE 5.0 workbench
  - supports v2.1 and v5.0 brokers
- **Version control and change management**
  - exploit standard WSWB repository support (CVS, ClearCase, …)
- **Message formatting enhancements**
  - message model re-based on XML Schema
  - full support for XML namespaces
- **Web Services**
  - support for HTTP (SOAP) protocols
  - WSDL generation from message model
- **Flexible protocol support**
  - real-time protocol and IP Multicast support
- **Accounting and statistics**
  - flexible solution for data collection and reporting
  - pub/sub reporting model and support for SMF records on z/OS

© 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

## Broker Topology

*Message Sets and flows are built in Eclipse and deployed to the runtime broker. The user name server sends principals to the Config Manager and broker. The z/OS broker contains all of the functions of the broker platform family, as well some z/OS specific features.*

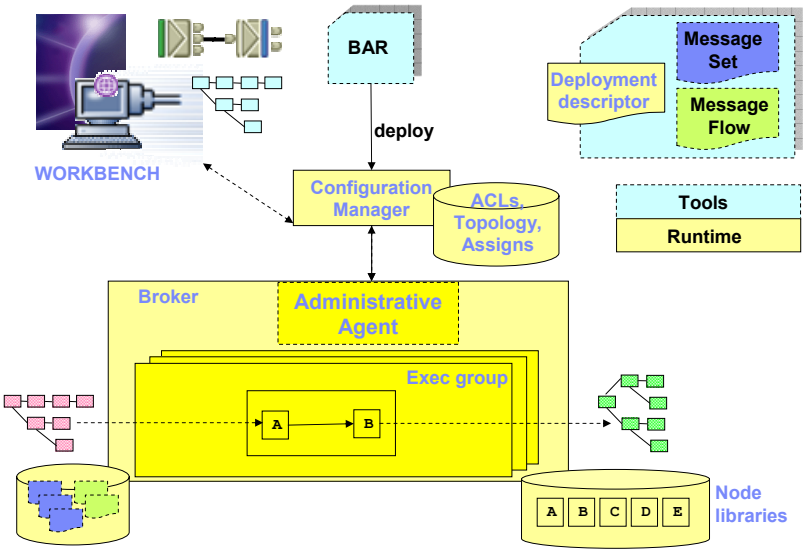



 ON DEMAND BUSINESS

Redbooks IBM

## Message Flow Construction

*Message flows and sets begin life as development artefacts inside the Broker Toolkit. The compiled forms of these artefacts are sent to the broker in a BAR file as XML configuration messages. These tailor the broker runtime to create flows and parsers implementing development artefacts.*

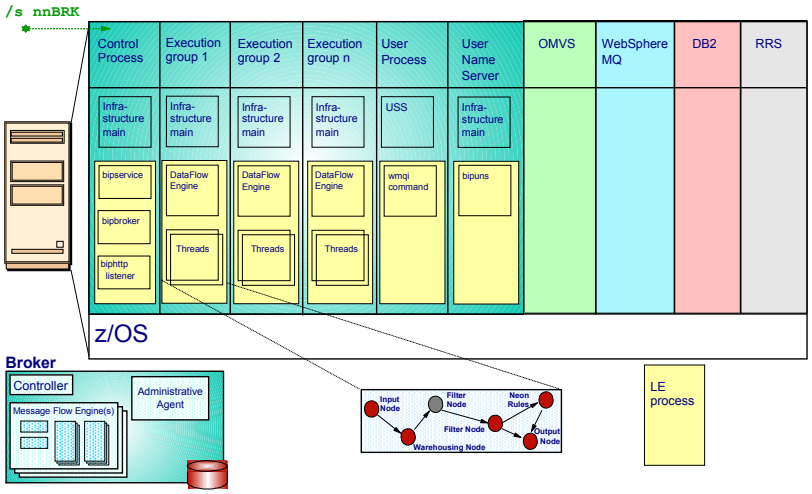



 ON DEMAND BUSINESS

Redbooks IBM

## Address Spaces

*The runtime comprises several address spaces. Each contains an infrastructure process starting main processors such as Admin Agent & Execution Groups; inside which flows run as TCBs. The broker interacts with many other address spaces.*

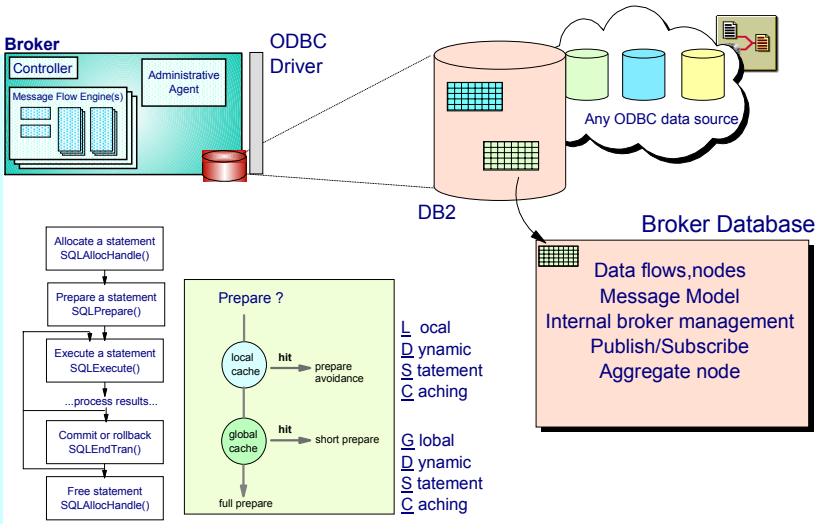



 ON DEMAND BUSINESS

Redbooks IBM

## Database Connectivity

*The broker accesses databases for flows with database nodes and for internal configuration information. Access is via ODBC; full statement caching and prepare avoidance is used to give high performance access.*

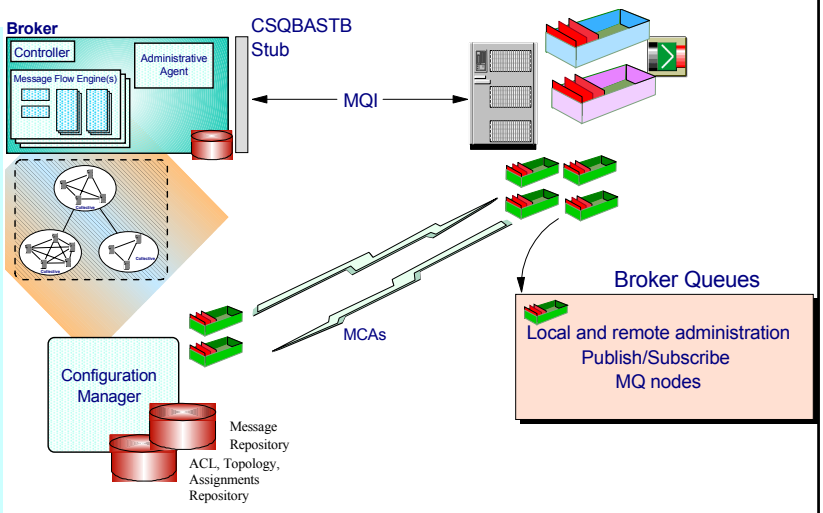



 ON DEMAND BUSINESS

Redbooks IBM

## WebSphere MQ

*The broker accesses WebSphere MQ for flows with MQ nodes and for internal configuration information. Access is via CSQBASTB which gives optimized performance for flows involving implicit RRS commit processing and context management.*

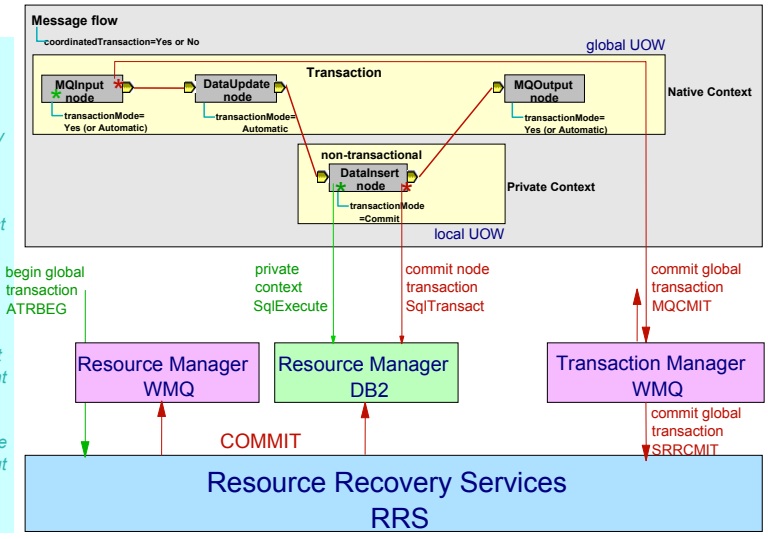



 ON DEMAND BUSINESS

Redbooks IBM

## Transaction Model

*The z/OS broker has a global transaction model exactly as you'd expect. It is possible for nodes to elect to commit outside this transaction. RRS is used for context management & commitment control between the flows resource managers, but only when required.*




 ON DEMAND BUSINESS

Redbooks IBM

## Appearance in WMQ, DB2, RRS

*We can see how the broker is represented by the various subsystems which it uses, including WMQ, DB2 and RRS. Resource Managers usually represent connected applications as threads. The broker is registered as a Work Manager with RRS. Such displays can be useful for problem determination.*

```
MQ03 DIS THREAD(*)

NAME     STA   REQ THREAD-XREF                USERID   ASID URID
RRSBATCH T       8                           MQ03BRK  0096 000000000000
RRSBATCH T    7847                           MQ03BRK  0096 000000000000
RRSBATCH T    7840                           MQ03BRK  0096 000000000000
MQ03BRK  T    4003 00000000000000000000000000 MQ03BRK 0098 000000000000
```

```
#DFK4 DIS THREAD(*) [RRSURID]

NAME     ST A    REQ ID           AUTHID   PLAN     ASID TOKEN
RRSAF    T      3613 MQ03BRK      MQ03BRK  DSNACLI  0096   463
RRSAF    T        16 MQ03BRK      MQ03BRK  DSNACLI  0096   464
RRSAF    DI        6 MQ03BRK      MQ03BRK  DSNACLI  0096   465
RRSAF    T        26 MQ03BRK      MQ03BRK  DSNACLI  0098   461
```

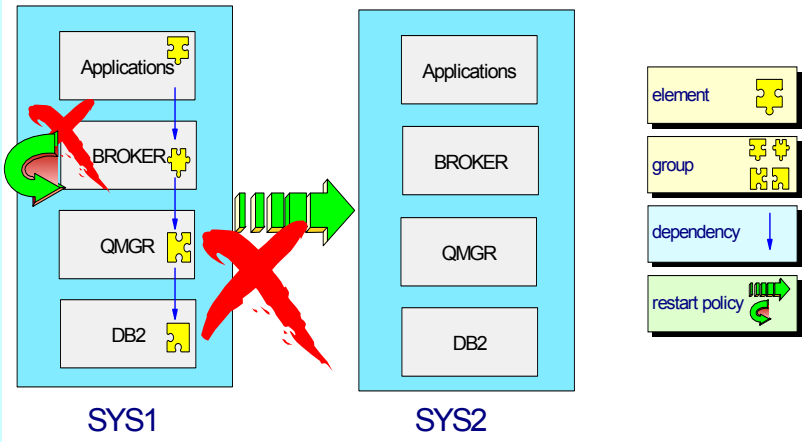
```
UR identifier : BA49C7047E9B66E80000003701050000
Create time : 2003/11/07 17:55:49.214233      Comments :
UR state : InFlight      UR type : Unpr
System : MVK4      Logging Group : PLEXK
SURID : N/A
Work Manager Name : BIP.MQ03BRK.0050659796.IBM.UA
   Display Work IDs              Display IDs formatted
   Luwid  . : Not Present
   Eid  . . : Not Present
   Xid  . . : Not Present
Expressions of Interest:
S   RM Name                              Type  Role
    DSN.RRSATF.IBM.DFK4                 Unpr  Participant
```



Redbooks IBM

## Restart Management

*Restart is defined by the Systems Administrator to organize applications into dependent groups for recovery after failure. Failures can include application failure or system failure and restart can be performed in place or on an adjacent Sysplex image.*



Redbooks IBM

## Agenda

- **Messaging overview for WebSphere**
- **WebSphere Default Messaging**
  - A.k.a. “Platform Messaging”
- **Developing and deploying JMS applications**
- **WBI Message Broker**
  - z/OS installation specifics

 ON DEMAND BUSINESS

Redbooks IBM

## Creating a WBI Message Broker on z/OS - Tasks

- **Set up users, file systems and security**
- **In UNIX System Services:**
  - "Create" component
  - Provide customization information
  - Run script to create broker structure in HFS and copy resource definition JCL to PDS
- **In TSO:**
  - Submit JCL members to create resources
  - Copy broker procedure to procedure library
  - Start broker

 ON DEMAND BUSINESS

Redbooks IBM

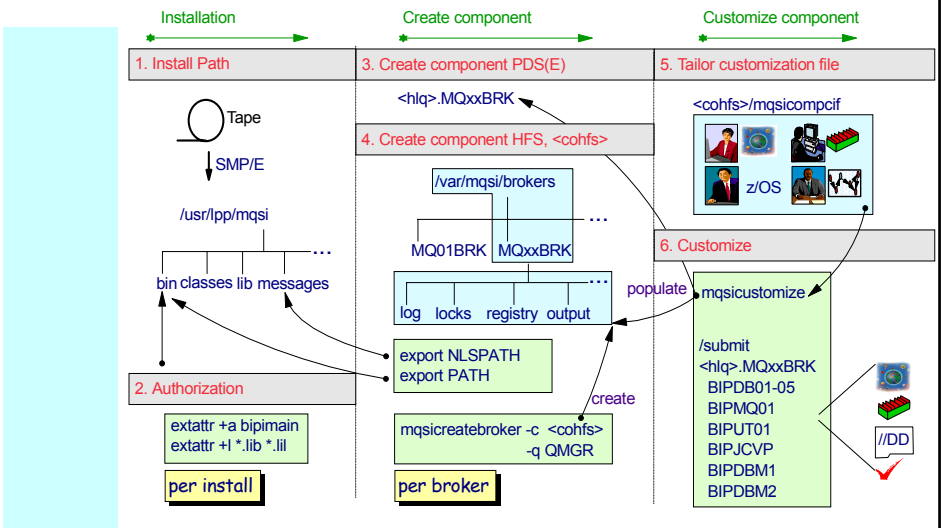
## Users, File systems and security

- **Create a started task ID for the Broker**
- **Create a profile for the Broker in the STARTED class**
- **Assign the started task ID to the profile**
- **Define an OMVS segment and a home directory under USS for the started task ID**
- **Setup DB2 and MQ authorizations for administering, running and operating the Broker**



Redbooks IBM

## Installation and Customization



Redbooks IBM

## JCL Settings

*Understanding the broker structure and associated subsystems allows us to make sense of the brokers started task JCL. We can see the internal process structure, the broker's use of classic MVS with USS, and associated products runtime availabilities via the STEPLIB.*

```
//MQ03BRK PROC INSTP='/usr/lpp/mqsi',
//       MAINP='bipimain',
//       SRVMP='bipservice',
//       COMPK='MQ03BRK',
//       STRTP='AUTO',
//       COHFS='/var/mqsi/brokers/MQ03',
//       STEPLIB='MQ03STEP'
//* 1
//CCPGM EXEC PGM=BPXBATCH,REGION=0M,TIME=NOLIMIT,
//       PARM='PGM &INSTP./bin/&MAINP. &SRVMP. &COMPK. &STRTP.'
//*
// INCLUDE MEMBER=(&STEPLIB.)  2
//*      3
//STDENV   DD PATH='&COHFS./ENVFILE'
//STDOUT   DD PATH='&COHFS./output/stdout',
//       4     PATHOPTS=(OWRONLY,OCREAT),
//             PATHMODE=(SIRWXU,SIRWXG)
//STDERR   DD PATH='&COHFS./output/stderr',
//             PATHOPTS=(OWRONLY,OCREAT),
// 5           PATHMODE=(SIRWXU,SIRWXG)
//SYSMDUMP DD SYSOUT=*
//         PEND\
MQ03STEP:
//         DD DISP=SHR,DSN=PP.ADLE390.ZOSR140.SCEERUN
//         DD DISP=SHR,DSN=DSN710PK.RUNLIB.LOAD
//         DD DISP=SHR,DSN=MQM.V531.SCSQAUTH
//         DD DISP=SHR,DSN=SYS2.DB2.SDSNLOD2
//         DD DISP=SHR,DSN=SYS2.DB2.SDSNLOAD
```

- ✓ Install path
- ✓ Infrastructure
- ✓ Component HFS
- ✓ STEPLIB
- ✓ BPXBATCH
- ✓ USS Environment
- ✓ Output files
- ✓ Dump

 ON DEMAND BUSINESS

Redbooks IBM

## USS Settings

*The STDENV reference in the broker JCL indicates the environment file that the broker uses to determine USS behaviour. Some of this information is also used for broker-wide configuration, and in some cases, user information. The lifecycle of this information is address space related.*

**/var/wmqi/brokers/MQxxBRK/ENVFILE**

```
LANG=C
LC_ALL=C
_BPX_BATCH_SPAWN=NO
_BPX_SHAREAS=NO
CLASSPATH=/usr/lpp/mqsi/classes:/usr/lpp/java/IBM/J1.4/lib
DSNAOINI=/var/brokers/MQ03/dsnaoini
JAVAHOME=/usr/lpp/java/IBM/J1.4
LIBPATH=/usr/lpp/mqsi/lib/wbimb:/usr/lpp/mqsi/lib:……
PATH=/usr/lpp/mqsi/bin:/usr/lpp/java/IBM/J1.4/bin
DISTHUB_PATH=/usr/lpp/DistHub/V5R0M1
TZ=GMT0BST
_CEE_RUNOPTS=XPLINK(ON)
MQSI_ARM_ELEMENTNAME=MQ03BRK
MQSI_ARM_ELEMENTTYPE=SYSWMQI
MQSI_COMPONENT_NAME=MQ03BRK
MQSI_REGISTRY=/var/brokers/MQ03
MQSI_USE_ARM=NO
NLSPATH=/usr/lpp/mqsi/messages/%L/%N
MQSI_CONSOLE_NLSPATH=/usr/lpp/mqsi/messages/En_US
MQSI_DB2_ALWAYS_PREPARE=NO
MQSI_MC_MESSAGES=++MC_MESSAGES++
```

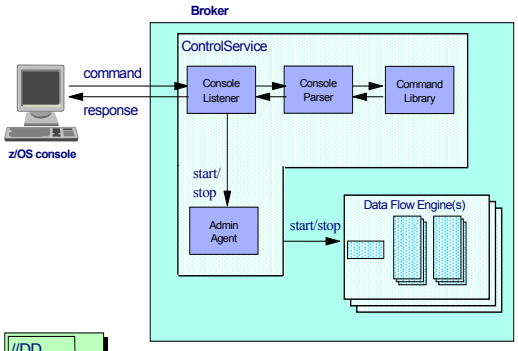
- ✓ Locale
- ✓ Java paths
- ✓ ODBC
- ✓ Broker paths
- ✓ Time zone
- ✓ Data conversion
- ✓ ARM
- ✓ Messages
- ✓ Internal OBDC control
- ✓ Not configured

 ON DEMAND BUSINESS

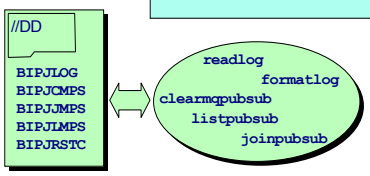
Redbooks IBM

## Administration and Usage

*Once the broker has been customized, operational interactions are via the MODIFY interface, so that local broker interactions are standard. Internally, the broker has an MVS console listener component that listens for MODIFY requests and translates these requests to, and responses from, the broker commands.*




| Modify Command | Abbrev |
|---|---|
| startcomponent | sc |
| stopcomponent | pc |
| changetrace | ct |
| reporttrace | rt |
| changebroker | cb |
| changeusernameserver | cu |
| list | l |
| nrfreload | nrfr |
| reload | re |
| stop | p |
| start | s |




```
/F MQ03BRK,PC
/F MQ03BRK,cb l=/usr/lpp/wmqi/lil:/u/odowda/lil
/F MQ03BRK,SC
```

 ON DEMAND BUSINESS

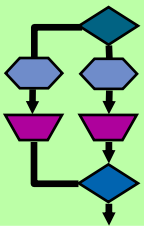
Redbooks IBM

## Useful Operational Output

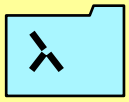
*As you'd expect from a regular z/OS subsystem, the operational output will be available from the MVS log and JOBLOGs in the form of BIP product messages. JOBLOGs have the advantage of partitioning messages by address space, typically execution groups. The hfs & trace may also provide some output in some circumstances.*

SYSLOG
JOBLOG


Trace



hfs



 ON DEMAND BUSINESS

Redbooks IBM

# Module 3

# Data access



ON DEMAND BUSINESS

Redbooks IBM

## Agenda

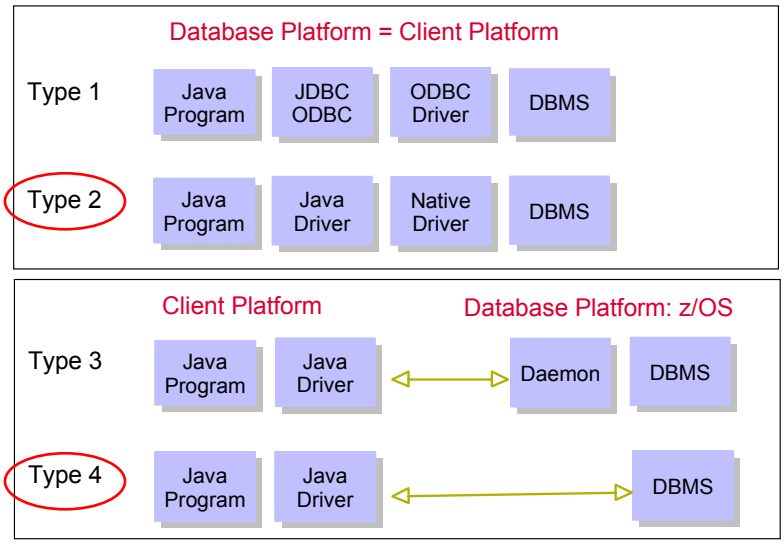
- **Java Database Connectivity (JDBC)**
- **SQLJ**
- **Accessing IMS DB through JDBC**
- **Java Record IO (JRIO)**



ON DEMAND BUSINESS

Redbooks IBM

## JDBC Driver Types




 ON DEMAND BUSINESS

Redbooks IBM

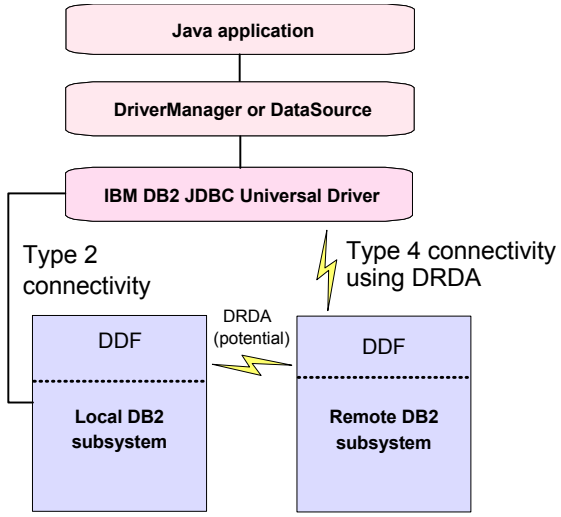
## Accessing DB2 on z/OS from J2EE/WAS

- **Locally**
  - from WebSphere on z/OS (servlets, JavaBeans and EJBs), UNIX and batch applications
  - using a JDBC Type 2 or Type 4 driver
- **Remotely**
  - from WebSphere on any other LPAR or system than where DB2 is running
  - using a JDBC Type 4 driver
- **Remote access from a distributed platform (Windows, Linux, AIX…) to DB2 on z/OS is also possible using DB2 Connect, but this requires a DB2 client to be installed on each client**
- **Stored Procedures in DB2 can be called from Java using the JDBC or SQLJ APIs**

 ON DEMAND BUSINESS

Redbooks IBM

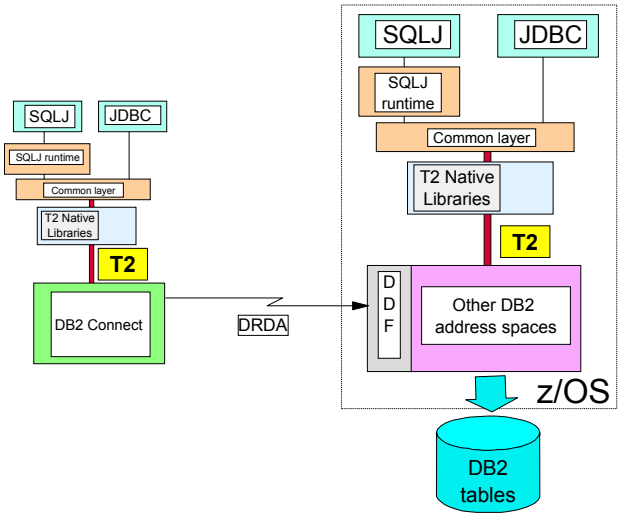
## JDBC Type 2 and Type 4 Drivers




 ON DEMAND BUSINESS

Redbooks IBM

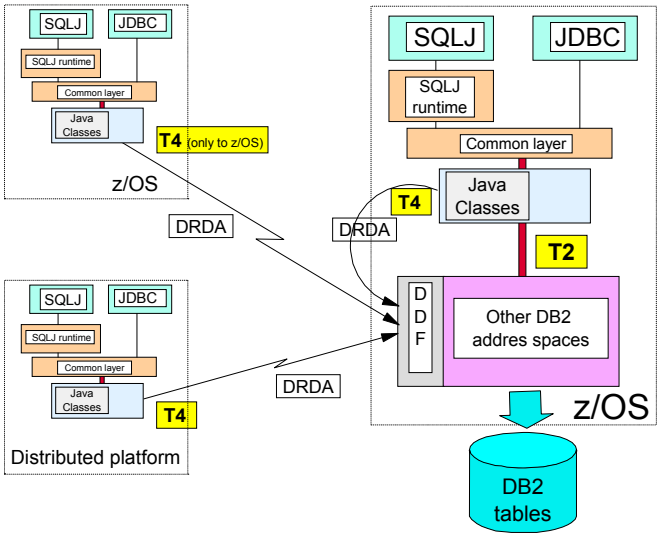
## JDBC T2 Driver Usage




 ON DEMAND BUSINESS

Redbooks IBM

## JDBC T4 Driver Usage




 ON DEMAND BUSINESS

Redbooks IBM

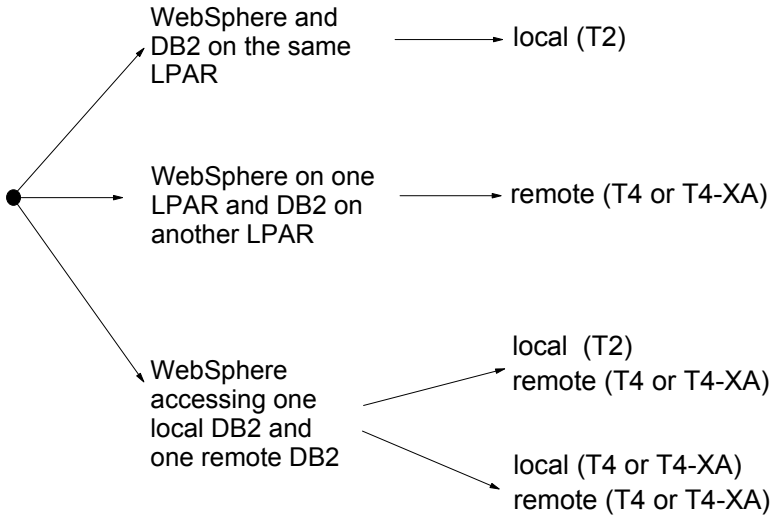
## JDBC Providers in WebSphere V6 on z/OS

- **Database type**
  - DB2, Cloudscape, Informix, Oracle etc.
- **Provider type (in case of DB2)**
  - Type 2 Legacy driver (CLI based) provider
    - Only T2 support
  - DB2 for z/OS local JDBC provider (RRS)
    - Only T2 support
    - 2-PC support
  - DB2 Universal JDBC driver provider
    - Both T2 and T4 support
    - 2-PC in T2 mode and 1-PC in T4 mode
- **Implementation type (data source)**
  - Connection pool data source
  - XA data source
    - Not in combination with RRS driver

 ON DEMAND BUSINESS

Redbooks IBM

## JDBC Driver Options on z/os




 ON DEMAND BUSINESS

Redbooks IBM

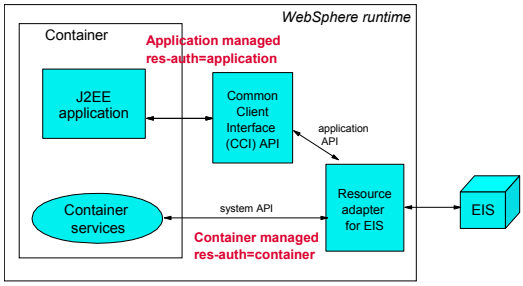
## DB2 Connectivity Type and Performance

- **For Local DB2, T2 gives best performance**
  - T2 (RRS) is better than T4 XA – no trip through network layer
- **When to use T4 vs. T4 XA?**
  - Remote DB2 - Choice is local vs global tran
  - Performance of both is close (1% if no 2-PC processing is done)
- **For multiple DB2s, choose the connectivity that gives the best performance for each location (T2 for local, T4 for remote)**
  - Application considerations? None, but deployer needs to know
- **Universal Driver T2 performance equivalent to Legacy T2 driver**
- **SQLJ: Regarding connectivity the same applies as for JDBC**
  - All arguments for SQLJ are still valid as for legacy case

 ON DEMAND BUSINESS

## Security in accessing a DB2 Database using JDBC

- **res-auth = Container**
  - The container will pass the required identity and credential to the connector
    - User ID/password provided in the datasource definition using the Authentication Alias
    - For a local connection you can specify the Connection Manager RunAS Identity Enabled, causing the thread ID to b passed to DB2.
- **res-auth = Application**
  - The application is expected to provide the required identity and credential, for example, getConnection(userid,password)
  - If no user ID is supplied, J2C defaults to the user ID defined in the datasource Authentication Alias.






## IBM DB2 Universal Driver for JDBC and SQLJ

- **Part of DB2 V8 for z/OS and DB2 V7 for z/OS as of APAR PQ80841 (and follow-on maintenance)**
- **Supports both Type 2 and Type 4**
- **When using in T2 mode**
  - native code is called
  - RRS controls 2PC

Redbooks IBM

## Configuration steps of DB2 Universal JDBC Driver (1)

- **STEPLIB**

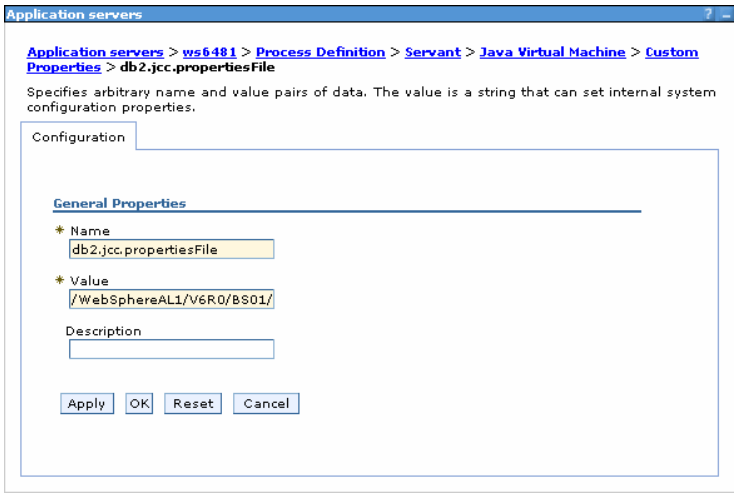
```
//STEPLIB  DD DISP=SHR,DSN=BBO6048.SBBOLD2
//           DD DISP=SHR,DSN=BBO6048.SBBOLOAD
//           DD DISP=SHR,DSN=DB8I8.SDSNEXIT
//           DD DISP=SHR,DSN=DB8I8.SDSNLOAD
//           DD DISP=SHR,DSN=DB8I8.SDSNLOD2
```

- **WebSphere environment variables**
  - DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH
  - DB2UNIVERSAL_JDBC_DRIVER_PATH
- **Binding the packages**
  - Instructions are in README file of JDBC driver directory
- **Authorize the resulting DNSJDBC plan**
- **Define a db2.jcc.properties file**
  - Should contain as a minimum the DB2 SSID to connect to
    - For example, `db2.jcc.ssid=D8I1`
  - You can only access ONE DB2 subsystem from a server
- **The properties files needs to be defined in WebSphere under `application servers -> <server instance> -> Java and Process management -> Process Definition -> Servant -> Java Virtual Machine -> Custom Properties`**

 ON DEMAND BUSINESS

Redbooks IBM

## db2.jcc.propertiesFile defined in WAS




 ON DEMAND BUSINESS

Redbooks IBM

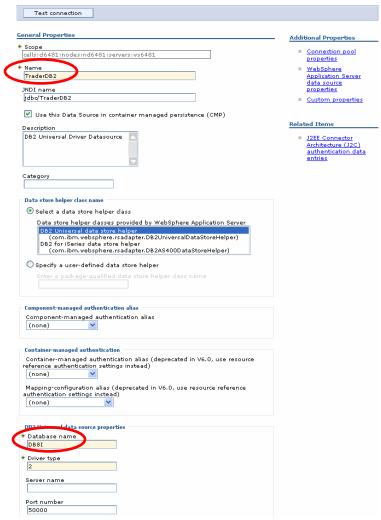
## Configuration steps of DB2 Universal JDBC Driver (2)

- **Define the DB2 Universal JDBC driver provider**
  - Recommendation is to do this at the *server* level
    - In order to be able to have a specific DB2 susbystem per server
- **Define a data source for the provider; specify the:**
  - JNDI name for the data source
  - Database name to connect to
    - May not be the same as the DB2 SSID!
  - Driver Type (2 or 4)
  - Server name and port number
    - Only in case of T4

 ON DEMAND BUSINESS

Redbooks IBM

## DB2 Universal JDBC Driver Provider – Datasource in WAS




 ON DEMAND BUSINESS

Redbooks IBM

## Agenda

- **Java Database Connectivity (JDBC)**
- **SQLJ**
- **Accessing IMS DB through JDBC**
- **Java Record IO (JRIO)**


ON DEMAND BUSINESS

Redbooks IBM

## SQLJ - Overview

- **An extension to the Java language to execute static SQL**
- **Also using JDBC drivers to connect to DB2**
- **Like in JDBC, both DriverManager and DataSource interfaces ara avaialable**
- **Additional steps are required to implement an SQLJ program**
  - Translation
  - Customization
    - If you do not perform the customization, the program will work fine, but using dynamic SQL!
- **SQLJ statements start with #sql and end with ;**


ON DEMAND BUSINESS

Redbooks IBM

## JDBC versus SQLJ

| static SQL | dynamic SQL |
|---|---|
| Authorization checking can be performed at the program level | Authorization checking can only be done at the user ID level |
| SQL logic easy to follow in a program | SQL statements wrapped in java method calls |
| Easier to define, maintain and access host variables | |
| Syntax errors are caught during translation or customization at the latest | Syntax errors are caught during runtime and will impose an additional effort on system testing |
| Persistent: lasts as long as package exists | Statements are cached until invalidated or freed for space management reasons |
| Statements exist after database is shut down | Statements cease once database is shut down |
| In most cases, a better performance | |
| SQLJ programs are smaller - certain code is provided by SQLJ | |
| SQLJ can do data type checking at preparation time | JDBC passes data type values without checking |

 ON DEMAND BUSINESS

Redbooks IBM

## Differences in coding between JDBC and SQLJ - Sample

**JDBC**

```
import javax.naming.InitialContext;
import javax.sql.DataSource;
import java.sql.Connection;

InitialContext context = new InitialContext();
DataSource ds = (DataSource) context.lookup("java:comp/env/jdbc/TraderDB2");
Connection conn = ds.getConnection();
```

**SQLJ**

```
import javax.naming.InitialContext;
import javax.sql.DataSource;
import java.sql.Connection;

// Create a connection context
#sql static context MyConnectionContext;

//Look up the datasource
InitialContext context = new InitialContext();
DataSource ds = (DataSource) context.lookup("java:comp/env/jdbc/TraderDB2");
Connection conn = ds.getConnection();

// Create an instance of the context class
MyConnectionContext myContext = new MyConnectionContext(con);
```

 ON DEMAND BUSINESS

Redbooks IBM

## SQLJ Executable Statement - Select

- **Syntax: #sql [context] { SQL-statement};**

```
String company= "IBM";
// Define the connection context
#sql public static context MyContext with (dataSource="jdbc/TraderDB2");

// Setup a Cursor to store values
#sql iterator cursor1 (String, float, float, float, float, float, float, float,
float, int, int);

// Create an instance of the context class
MyContext context = new MyContext();

// The SQLJ Select statement
#sql [context] cursor1 {
          SELECT * FROM TRADER.COMPANY
          WHERE COMPANY = :company
}
// Now, cursor1 can be used to iterate through the returned data
// Alternately, results can be stored in variable
// Use the INTO sqlj statment and specify variables to store into.
// Tell the Transaction Manager that we are done with the connection for
// now
context.close();
```

 ON DEMAND BUSINESS

Redbooks IBM

## SQLJ – Using Host variables

```
String empno= "000130";
// Define the connection context
#sql public static context MyContext with (dataSource="jdbc/ITSO1");

// Host variables to select into
String firstname = null;
String lastname = null;
java.sql.Date hiredate = null;
java.math.BigDecimal salary = null;

// Create an instance of the context class
MyContext context = new MyContext();

// The SQLJ Select statement
#sql [context] {
         SELECT FIRSTNME, LASTNAME, HIREDATE, SALARY
         INTO :firstname, :lastname, :hiredate, :salary
         FROM EMP WHERE EMPNO = :empno
}
// Tell the Transaction Manager that we are done with the connection for now
context.close();
```

 ON DEMAND BUSINESS

Redbooks IBM

## SQLJ – Setting Transaction Isolation Level

- **Syntax: #sql [context] { SET TRANSACTION ISOLATION LEVEL level};**

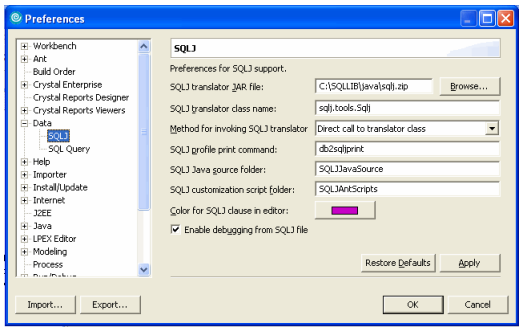
| SQLJ isolation level | DB2 isolation level |
|---|---|
| READ UNCOMMITTED | UR (uncommitted read) |
| READ COMMITTED | CS (cursor stability) |
| REPEATABLE READ | RS (read stability) |
| SERIALIZABLE | RR (repeatable read) |


ON DEMAND BUSINESS

Redbooks IBM

## Setting up the SQLJ Environment in RAD

- **Add SQLJ development capability in Window → Preferences → Workbench → Capabilities**
- **Set up SQLJ translator in Window → Preferences → Data → SQLJ**



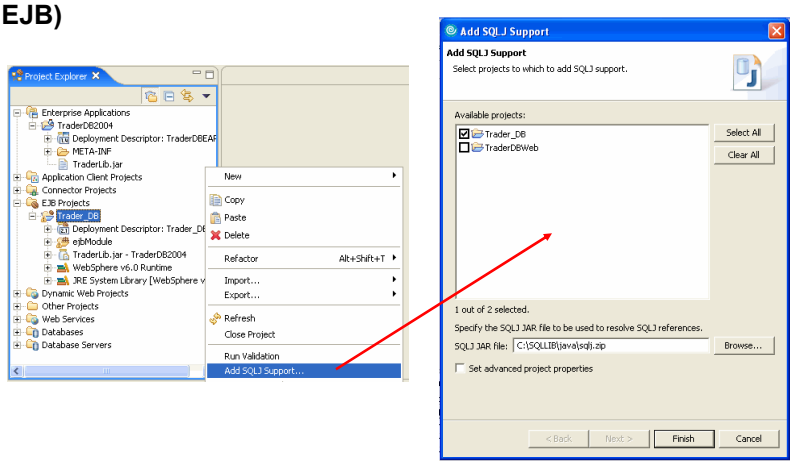


ON DEMAND BUSINESS

Redbooks IBM

## SQLJ support needs to be added to the project

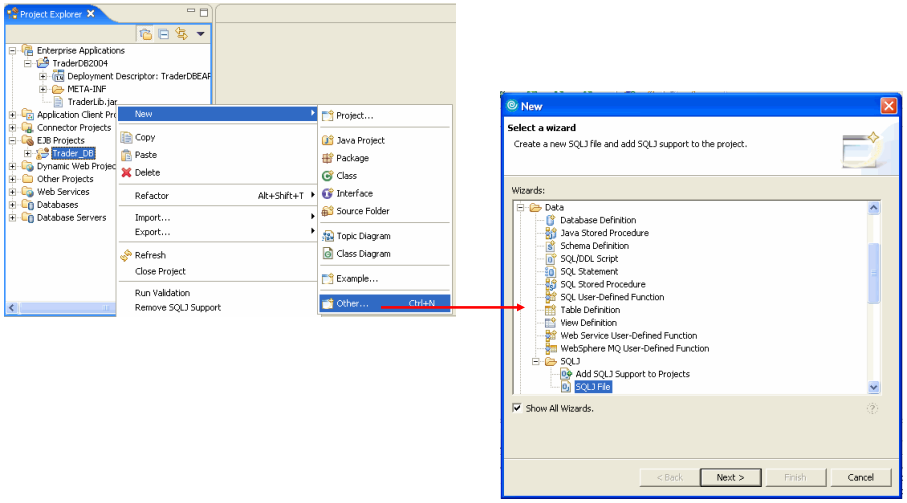
- **SQLJ support needs also to be added to the project (Web or EJB)**




 ON DEMAND BUSINESS

Redbooks IBM

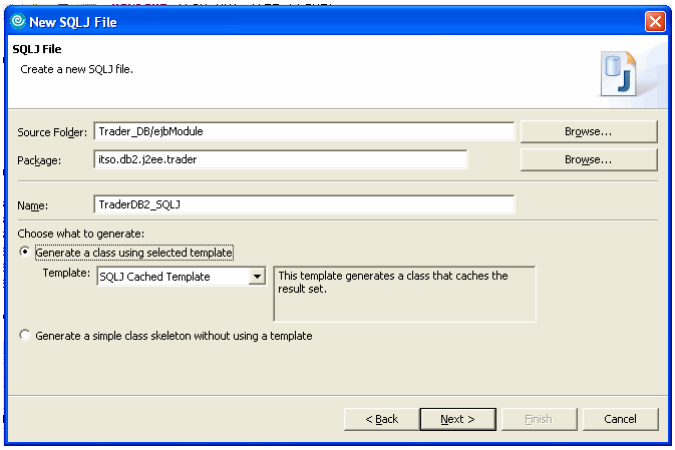
## Adding an SQLJ file to the Project




 ON DEMAND BUSINESS

Redbooks IBM

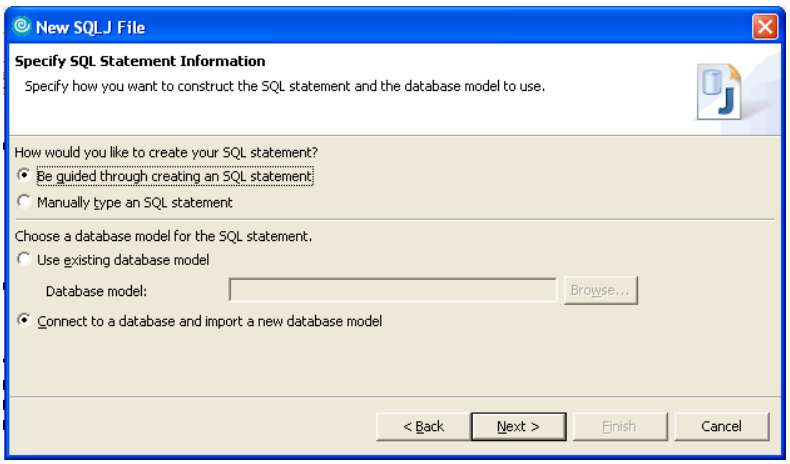
## Creating an SQLJ File - 1




 ON DEMAND BUSINESS

Redbooks IBM

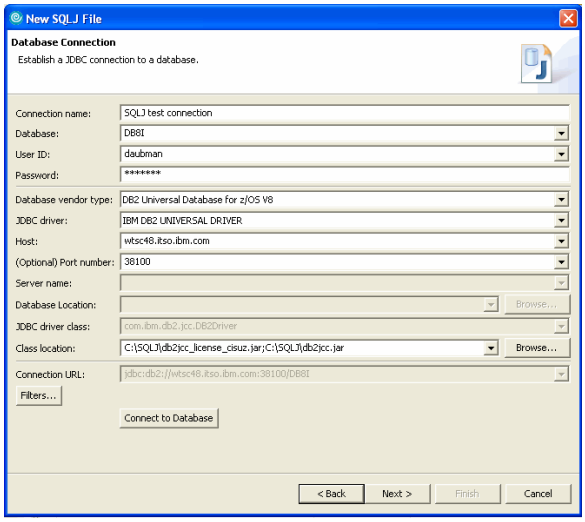
## Creating and SQLJ File - 2




 ON DEMAND BUSINESS

Redbooks IBM

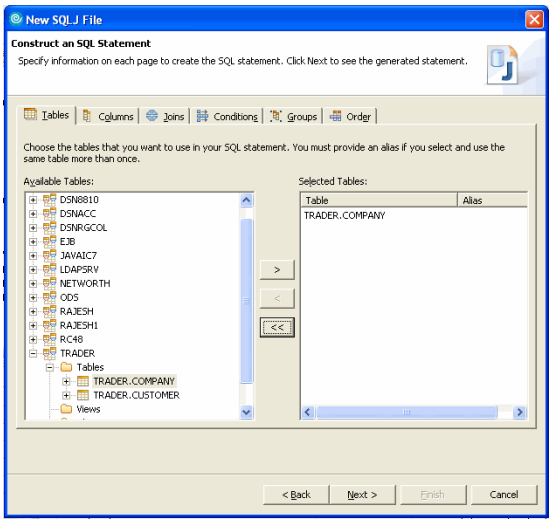
## Creating and SQLJ File - 3




 ON DEMAND BUSINESS

Redbooks IBM

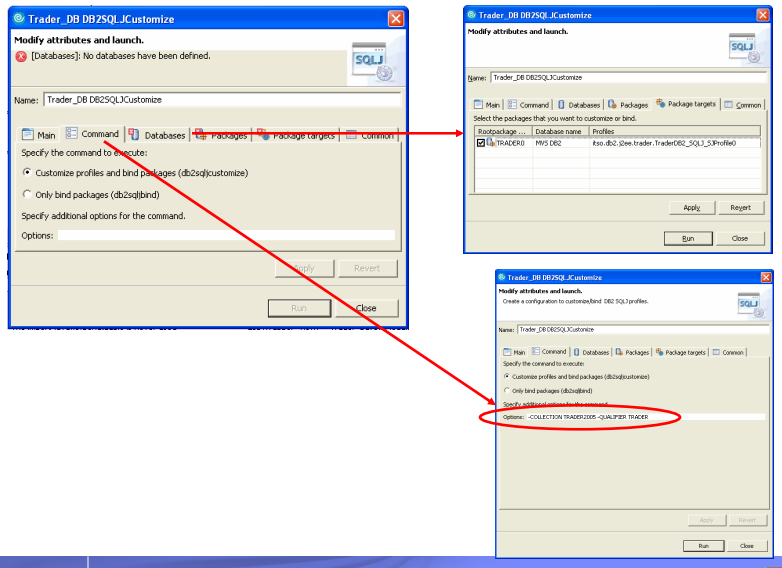
## Constructing an SQL Statement in RAD




 ON DEMAND BUSINESS

Redbooks IBM

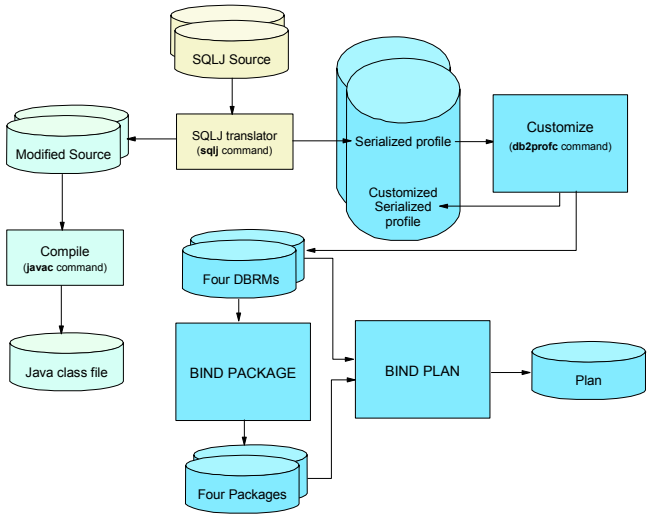
## SQLJ Customization in RAD




 ON DEMAND BUSINESS

Redbooks IBM

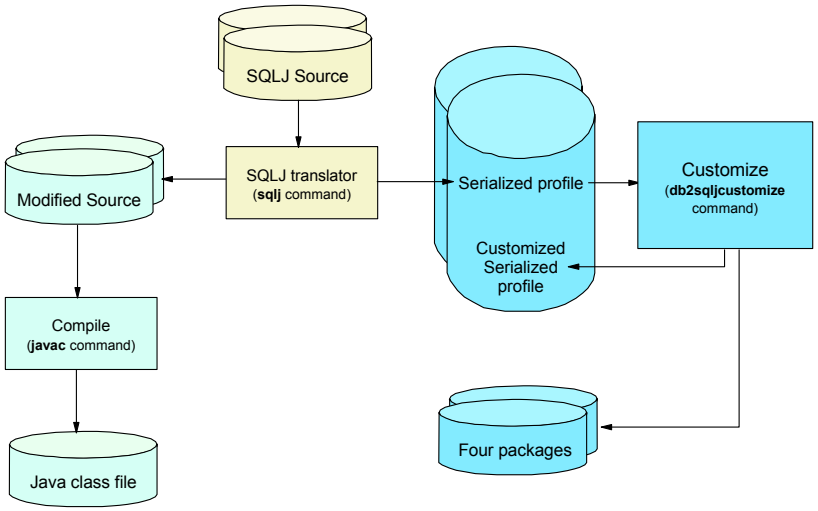
## SQLJ steps on z/OS- using legacy driver




 ON DEMAND BUSINESS

Redbooks IBM

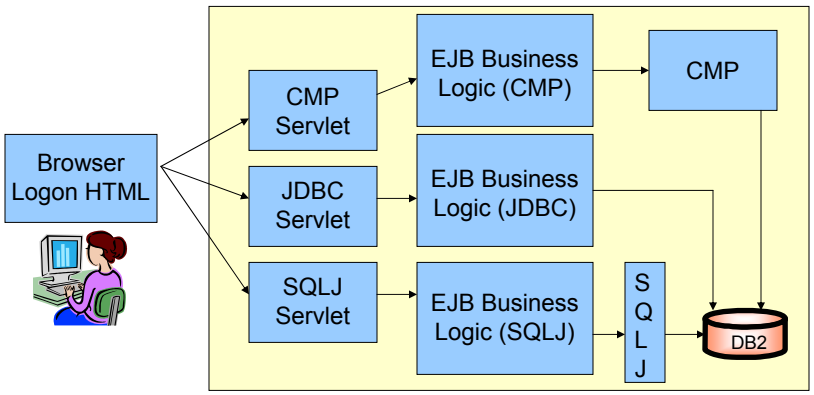
## SQLJ steps on z/OS- using legacy driver




 ON DEMAND BUSINESS

Redbooks IBM

## Trader sample application layout




 ON DEMAND BUSINESS

Redbooks IBM

## Agenda

- Java Database Connectivity (JDBC)
- SQLJ
- Accessing IMS DB through JDBC
- Java Record IO (JRIO)

 ON DEMAND BUSINESS

Redbooks IBM

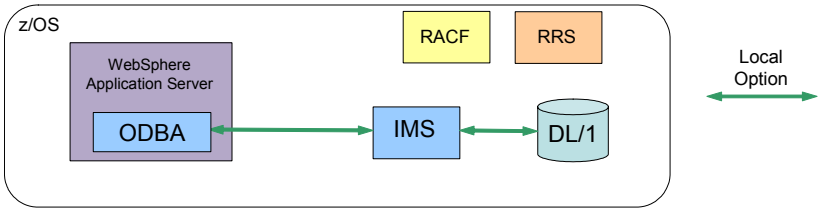
## Accessing IMS DB using JDBC - Tasks

- Configuring the WAS servant region to use ODBA
- Installing the IMS JDBC resource adapter
- Configuring the IMS JDBC J2C connection factory
- Installation verification with sample application
- Problem determination

 ON DEMAND BUSINESS

Redbooks IBM

## Topology




 ON DEMAND BUSINESS

Redbooks IBM

## Configuring the WAS servant region for ODBA

- **Create a DRA member with name DFSxxxx0, where xxxx is the DRAName in the IMS JDBC J2C connection factory.**
- **Add the dataset with this DRA member to the STEPLIB of the servant startup JCL.**
- **Also, add the data sets with ODBA modules and IMS Java modules to this STEPLIB.**

 ON DEMAND BUSINESS

Redbooks IBM

## Sample DRA member

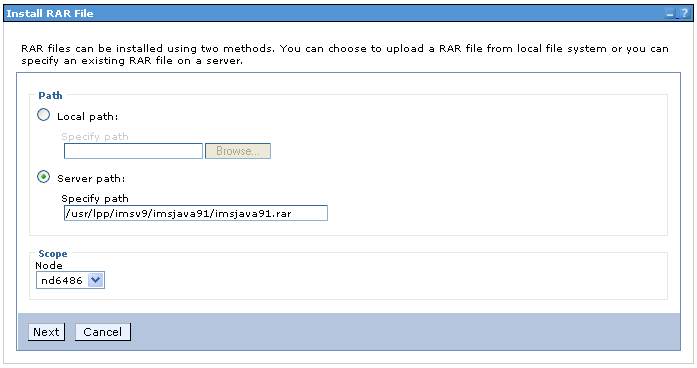
```
DFSIMSA0 CSECT
         DFSPRP DSECT=NO,                                          X
               FUNCLV=1,                  CCTL FUNCTION LEVEL      X
               DDNAME=CCTLDD,             DDN FOR CCTL RESLIB DYNALOC X
               DSNAME=IMS910A.SDFSRESL,   DSN FOR CCTL RESLIB      X
               DBCTLID=IM4B,              NAME OF DBCTL REGION     X
               USERID=,                   NAME OF USER REGION      X
               MINTHRD=001,               MINIMUM THREADS          X
               MAXTHRD=005,               MAXIMUM THREADS          X
               TIMER=60,                  IDENTIFY TIMER VALUE - SECS X
               FPBUF=010,                 FP FIXED BFRS PER THREAD X
               FPBOF=010,                 FP OVFLW BFRS PER THREAD X
               CNBA=010,                  FP FIXED NBA BFRS PER CCTL X
               SOD=T,                     SNAP DUMP CLASS          X
               AGN=IVP                    APPLICATION GROUP NAME
         END
```

 ON DEMAND BUSINESS

Redbooks IBM

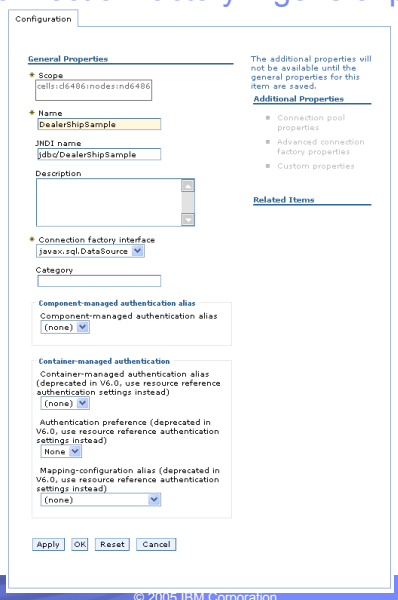
## Installing the IMS JDBC resource adapter

- **imsjava91.rar**
- **In the admin console, under “Resources”, click “Resource adapters” and then “Install RAR”.**




 ON DEMAND BUSINESS

## IMS JDBC J2C connection factory – general properties





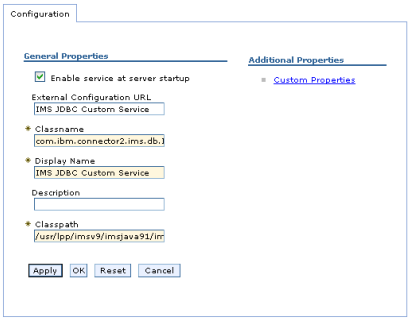

## IMS JDBC J2C connection factory – custom properties

| Name | Value | Description | Required |
|---|---|---|---|
| DatabaseViewName | samples.dealership.AUTPSB11DatabaseView | Fully qualified name of the database view subclass | false |
| DRAName | IMSA | The DRA name of the IMS to connect to | false |
| TransactionResourceRegistration | dynamic | Type of transaction resource registration (enlistment). This value must be "dynamic" (deferred) for this resource adapter. | false |

Total 3

- **DRAName – must match the ID parameter of the DataStore statement in the IMS Connect configuration member.**
- **DatabaseViewName – the fully qualified DLIDatabaseView subclass name**

## Defining a custom service in WAS for IMS JDBC – Part 1

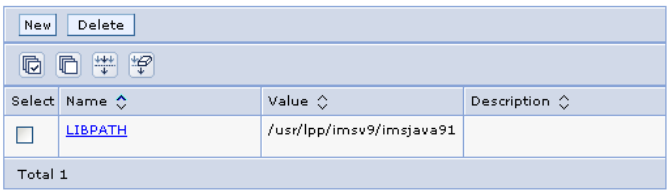



- **Classname**
  - com.ibm.connector2.ims.db.IMSJdbcCustomService
- **Classpath**
  - /usr/lpp/imsv9/imsjava91/imsjava.jar
- **Enable service at server startup enabled**



## Defining a custom service in WAS for IMS JDBC – Part 2

- **In *Server Infrastructure -> Java and Process Management*, click *Process Definition*.**
- **Select servant region and click on *Environment Entries.***
- **Add an new variable for LIBPATH with a value of */usr/lpp/imsv9/imsjava91***
  - this provides access to native libraries
- **Restart the server.**



Redbooks IBM

## Agenda

- **Java Database Connectivity (JDBC)**
- **SQLJ**
- **Accessing IMS DB through JDBC**
- **Java Record IO (JRIO)**


ON DEMAND BUSINESS

Redbooks IBM

## Module 4

## Web services


ON DEMAND BUSINESS

Redbooks IBM

## Agenda

- **Technical overview**
  - **Web services in WAS V6.01 for z/OS**
  - **Developing a Web service in RAD**
  - **Web services between WebSphere and CICS**
  - **Web services between WebSphere and IMS**
  - **Web services Between WebSphere and DB2**

 ON DEMAND BUSINESS

Redbooks IBM

## What are Web services?

*Web services are self-contained, modular applications that can be described, published, located, and invoked over a network.*

 ON DEMAND BUSINESS

Redbooks IBM

## Web services Properties

- **Web services are self-contained.**
  - On the client side, no additional software is required. A programming language with XML and HTTP client support is enough to get you started. On the server side, merely an HTTP server and a SOAP server are required.
- **Web services are self-describing.**
  - Using WSDL all the information required to implement a Web service as a provider or invoke a Web service as a requester are provided.
- **Web services can be published, located, and invoked across the Web.**
  - This technology uses established lightweight Internet standards such as HTTP. It leverages the existing infrastructure.
- **Web services are modular.**
  - Simple Web services can be aggregated to more complex ones, either using workflow techniques or by calling lower-layer Web services from a Web service implementation. Web services can be chained together to perform higher-level business functions. This shortens development time and enables best-of-breed implementations.
- **Web services are language-independent and interoperable.**
  - The client and server can be implemented in different environments. Theoretically, any language can be used to implement Web service clients and servers.
- **Web services are inherently open and standard-based.**
  - XML and HTTP are the major technical foundation for Web services. A large part of the Web service technology has been built using open-source projects. Therefore, vendor independence and interoperability are realistic goals.
- **Web services are loosely coupled.**
  - Traditionally, application design has depended on tight interconnections at both ends. Web services require a simpler level of coordination that allows a more flexible reconfiguration for an integration of the services in question.
- **Web services provide programmatic access.**
  - The approach provides no graphical user interface; it operates at the code level. Service consumers have to know the interfaces to Web services but do not have to know the implementation details of services.
- **Web services provide the ability to wrap existing applications.**

 ON DEMAND BUSINESS

Redbooks IBM

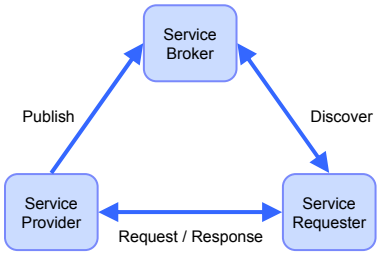
## Web services Core Standards

- **Simple Object Access Protocol (SOAP)**
  - Stateless
  - In principle one-way, but request/response possible too
  - SOAP 1.2 is the most current version
  - http://www.w3.org/2000/xp/Group/
- **Web Services Description Language (WSDL)**
  - Defines endpoints, operations and messages
  - Protocol independent
  - http://www.w3.org/TR/wsdl
- **Extended Markup Language (XML)**
  - http://www.w3.org/XML
- **Universal Description, Discovery, and Integration (UDDI)**
  - Defines means to publish and to discover Web services
  - Optional

 ON DEMAND BUSINESS

Redbooks IBM

## Web services Components and Operations

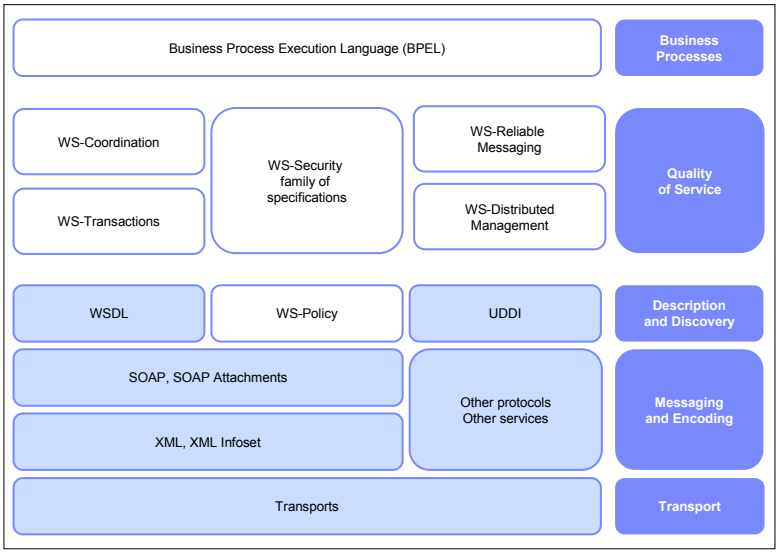



- The *service provider* creates a Web service and possibly publishes its interface and access information to the service broker.
- The *service broker* (also known as *service registry*) is responsible for making the Web service interface and implementation access information available to any potential service requestor.
- The *service requestor* binds to the service provider in order to invoke one of its Web services having optionally located entries in the broker registry using various find operations.

 ON DEMAND BUSINESS

Redbooks IBM

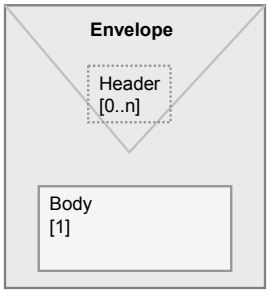
## Web services standards




 ON DEMAND BUSINESS

Redbooks IBM

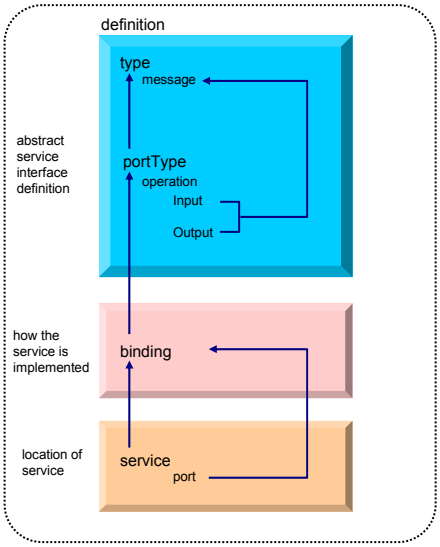
## SOAP envelope




 ON DEMAND BUSINESS

Redbooks IBM

## WSDL Elements and Relationships




 ON DEMAND BUSINESS

Redbooks IBM

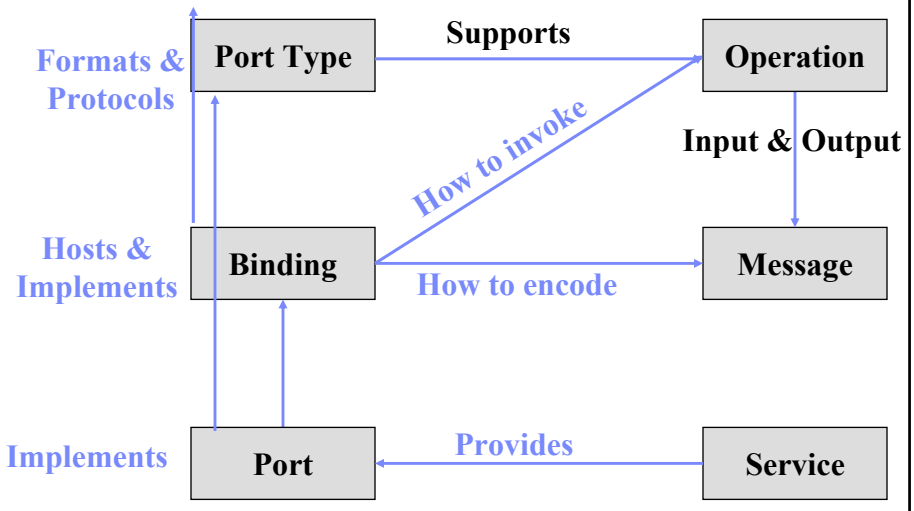
## WSDL: Fundamental Concepts

- **PortTypeOperation**
  - Designates a specific function offered by a service
  - Defines the abstract interface to the service
  - Associated with one or more operations
- **Message**
  - Defines the information exchanged at the time an operation is executed (input and output)
  - Made of one or more "parts"
- **Part**
  - Elemental piece of information in a message (an individual "parameter" or "return value")
- **Binding**
  - Defines the mapping to the specific implementation
    - mapping operations to the implementation specifics (a method call in Java, a CICS transaction, ...)
    - data type mapping and data conversion
- **Port**
  - Identifies the actual location of the service (a URL, a Java class, ....)
  - Each Port has one and one only binding
  - Each binding is associated with a single PortType
  - A PortType can have multiple bindings and be associated with multiple ports

 ON DEMAND BUSINESS

Redbooks IBM

## Ingredients Of WSDL




 ON DEMAND BUSINESS

## WSDL: Typical Java Binding

- **Service implemented by a Java class**
- **Operations map to individual Java methods on the class**
- **Messages map to the method parameters**
  - Input parameters are a sequence of parts
  - Output parameters are a sequence of parts
- **Parts map to a single input or output parameter**
- **Faults map to Java exceptions**



## Agenda

- **Technical overview**
- **Web services in WAS V6.01 for z/OS**
- **Developing a Web service in RAD**
- **Web services between WebSphere and CICS**
- **Web services between WebSphere and IMS**
- **Web services Between WebSphere and DB2**

Redbooks IBM

## J2EE 1.4 Requirements for Web Services

- **JAX-RPC 1.1 (JSR 101 1.1)**
- **WSEE (JSR 109 1.1)**
- **WS-I Basic Profile 1.0**
  - XML 1.0
  - SOAP 1.1
  - WSDL 1.1
  - UDDI 2.0
- **SAAJ 1.2**
- **JAXR 1.0**


ON DEMAND BUSINESS

Redbooks IBM

## Changes in Web Services

| WebSphere 4.0 & 5.0 | WebSphere 5.02/5.1 | WebSphere 6.0 |
|---|---|---|
| **Apache SOAP**<br>▪ The programming model, deployment model and engine<br><br>**Proprietary APIs**<br>▪ Because Java standards for Web services didn't exist<br><br>**Not WS-I compliant** | **JAX-RPC (JSR-101) 1.0**<br>▪ New standard API for programming Web services in Java<br><br>**JSR-109 1.0**<br>▪ New J2EE deployment model for Java Web services<br><br>**SAAJ 1.1**<br><br>**WS-Security**<br>▪ Extensions added<br><br>**WS-I Basic Profile 1.0**<br>▪ Profile compliance<br><br>**UDDI4J version 2.0 (client)**<br><br>**Apache Soap 2.3 enhancements**<br><br>The engine is a new high performance SOAP engine supporting both HTTP and JMS | **JAX-RPC (JSR-101) 1.1**<br>▪ Additional type support<br>▪ xsd:list<br>▪ Fault support<br>▪ Name collision rules<br>▪ New APIs for creating Services<br>▪ isUserInRole()<br>**JSR-109 (WSEE) 1.1**<br>▪ Moved to J2EE 1.4 schema types<br>▪ Migration of web services client DD moving to appropriate container DDs<br>▪ Handlers support for EJBs<br>▪ Service endpoint interface (SEI) is a peer to LI/RI<br>**SAAJ 1.2**<br>▪ APIs for manipulating SOAP XML messages<br>▪ SAAJ infrastructure now extends DOM (easy to cast to DOM and use)<br>**WS-Security**<br>▪ WSS 1.0<br>▪ Following WS-I Security Profile<br>**WS-I Basic Profile 1.1**<br>▪ Attachments support<br>**WS-TX (WS transactions)**<br>▪ WS-AtomicTransaction<br>**JAXR 1.0**<br>**UDDI v3**<br>▪ Includes both the registry implementation and the client API library |


ON DEMAND BUSINESS

Redbooks IBM

## WebSphere's Support for J2EE 1.4 Web Services

- **Web Services Updates from WebSphere V5**
  - JAX-RPC 1.1 (previously JAX-RPC 1.0)
  - WSEE 1.1 (previously WSEE 1.0)
  - SAAJ 1.2 (previously SAAJ 1.1)
  - WS-I Basic Profile 1.1[1] (previously WS-I BP 1.0)
  - UDDI V3[2] (previously UDDI V2)
- **New Web Service**
  - JAXR 1.0

[1] Only WS-I Basic Profile 1.0 is required by J2EE 1.4

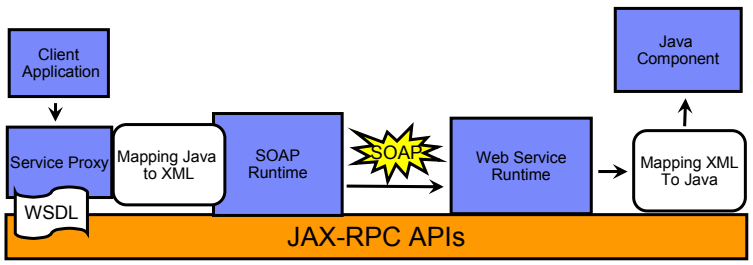
[2] Only UDDI V2 is required by J2EE 1.4

 ON DEMAND BUSINESS

Redbooks IBM

## JAX-RPC (JSR 101): Objectives

- **Java API for XML based Remote Procedure Call (JAX-RPC) formalizes the procedure for invoking Web services in an RPC-like manner**
- **Defines Client side APIs to access web service as well as server side requirements**
- **Defines mapping model between WSDL, XML and Java**
- **Defines a Handler model to the client and the server side to allow your custom code to intercept the request and the response**




 ON DEMAND BUSINESS

Redbooks IBM

## JAX-RPC (JSR 101) 1.1 Updates

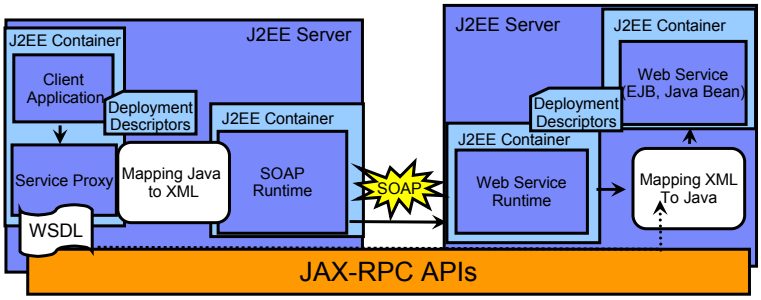
- **Additional type support**
- **xsd:list**
- **Fault support**
- **Name collision rules**
- **New APIs for creating Services**
- **isUserInRole()**

 ON DEMAND BUSINESS

Redbooks IBM

## Web Services for J2EE (JSR 109): Objectives

- **Specification for JAX-RPC Web Services in a J2EE environment**
- **Defines a standard deployment model for Web Services for J2EE components**
  - How a J2EE Server component can be described as a Web Service
  - How a J2EE Client component can be described for calling Web Services using JAX-RPC




 ON DEMAND BUSINESS

Redbooks IBM

## Web Services for J2EE (JSR 109) 1.1 Updates

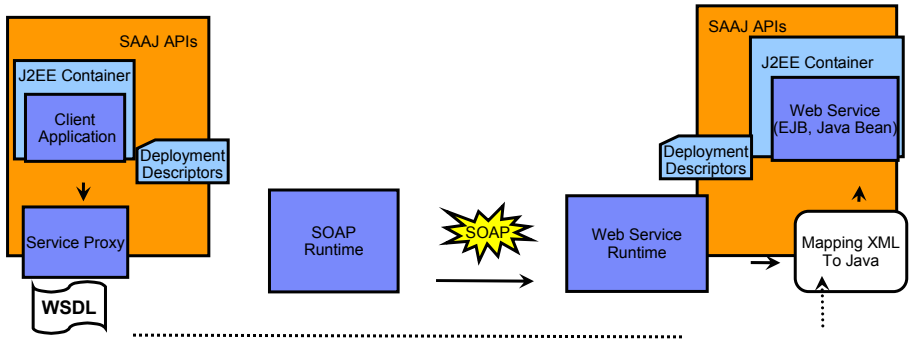
- **Moved to J2EE 1.4 schema types**
- **Migration of web services client DD moving to appropriate container DDs**
- **Event handler support for EJBs**
- **Service endpoint interface (SEI) is a peer to the Local Interface / Remote Interface**

 ON DEMAND BUSINESS

Redbooks IBM

## SOAP with Attachments API for Java (SAAJ)

- **SAAJ provides a standard set of APIs to send XML documents (including attachments) over the Internet**
- **Similar to JAX-RPC but requires additional effort on the client and server sides**

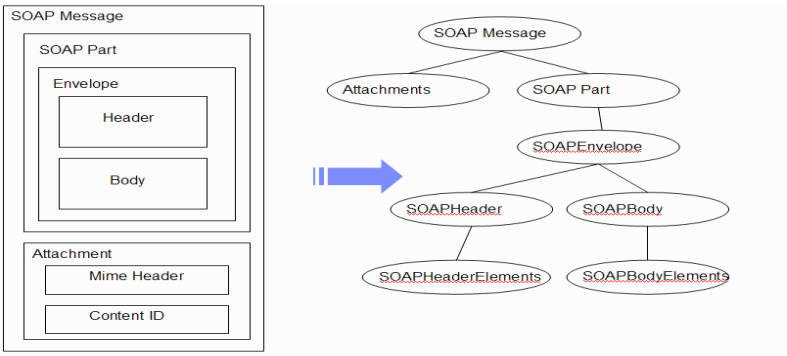



 ON DEMAND BUSINESS

Redbooks IBM

## SAAJ 1.2 Updates

- **APIs for manipulating SOAP XML messages**
- **SAAJ infrastructure now extends DOM**
  - easy to cast to DOM and use




 ON DEMAND BUSINESS

Redbooks IBM

## WS-I Updates

- **Organization promoting interoperability between Web Services**
- **Goal is to provide standardization of Web Services between different platforms, applications, and programming languages**
- **Defines profiles**
  - WS-I Basic 1.0 Profile currently available
  - WS-I 1.1 Profile adds support for attachments

 ON DEMAND BUSINESS

Redbooks IBM

## UDDI V3 Overview

- **Includes**
  - Registry implementation
  - Client API library
- **Client UDDI v3 API different than JAXR**
  - Exposes more native UDDI v3 functionality not available in JAXR

 ON DEMAND BUSINESS

Redbooks IBM

## UDDI V3 Support in WebSphere V6.0

- **Mandatory parts of the UDDI v3 specification**
  - v3 inquiry, publish and security APIs
- **Some optional parts of the v3 specification**
  - v1 and v2 inquiry and publish APIs
  - Ownership transfer (intra-node custody transfer)
- **Additional functionality**
  - Graphical user interface for inquiry and publication
  - Admin/management interface and GUI
  - UDDI v3 client for Java
  - V2 EJB interface, and UDDI4J, for backwards compatibility

 ON DEMAND BUSINESS

Redbooks IBM

## WebSphere's Support for J2EE 1.4 Web Services

- **Web Services Updates from WebSphere V5**
  - JAX-RPC 1.1 (previously JAX-RPC 1.0)
  - WSEE 1.1 (previously WSEE 1.0)
  - SAAJ 1.2 (previously SAAJ 1.1)
  - WS-I Basic Profile 1.1[1] (previously WS-I BP 1.0)
  - UDDI V3[2] (previously UDDI V2)
- **New Web Service**
  - JAXR 1.0

[1] Only WS-I Basic Profile 1.0 is required by J2EE 1.4

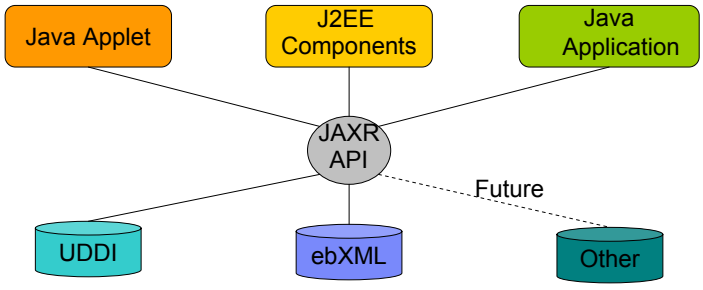
[2] Only UDDI V2 is required by J2EE 1.4

 ON DEMAND BUSINESS

Redbooks IBM

## What is JAXR?

- **A standard API for Java clients to access diverse registries**
- **Provides a union of best features of dominant registry specs (particularly UDDI and ebXML)**

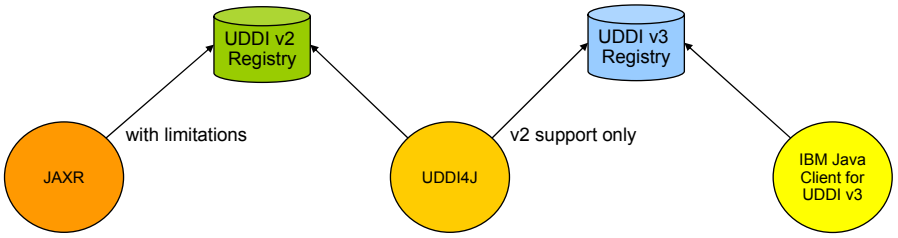



 ON DEMAND BUSINESS

Redbooks IBM

## JAXR compared to UDDI Java APIs

- **There are some limitations in the mapping of the JAXR Information Model to UDDI**
- **UDDI4J maps precisely to UDDI v2 and therefore provides a more complete API capability**
- **IBM Java Client for UDDI v3 maps precisely to UDDI v3**
  - JAXR is for UDDI v2 only, no mapping to UDDI v3 APIs
- **JAXR aims to provide portability across different XML Registries**




 ON DEMAND BUSINESS

Redbooks IBM

## WS-Security Overview

- **Provides enhancements to SOAP messaging with**
  - Message authentication via XML Digital Signature
  - Message confidentiality via XML Encryption
    - J2EE role-based authorization
  - Message integrity via XML Canonicalization

 ON DEMAND BUSINESS

## WS-Security

- **WS-Security is a message level standard defined how to secure SOAP messages, using**
  - XML Digital Signature:
    - Digitally sign the SOAP XML document, providing integrity, authenticity, and signer authentication
  - XML Encryption:
    - Process for encrypting data and representing the result in XML providing confidentiality
  - XML Canonicalization:
    - provides normalized XML document that can be digitally signed and verified
  - Credential propagation through security tokens
  - Applies to SOAP/HTTP and SOAP/JMS

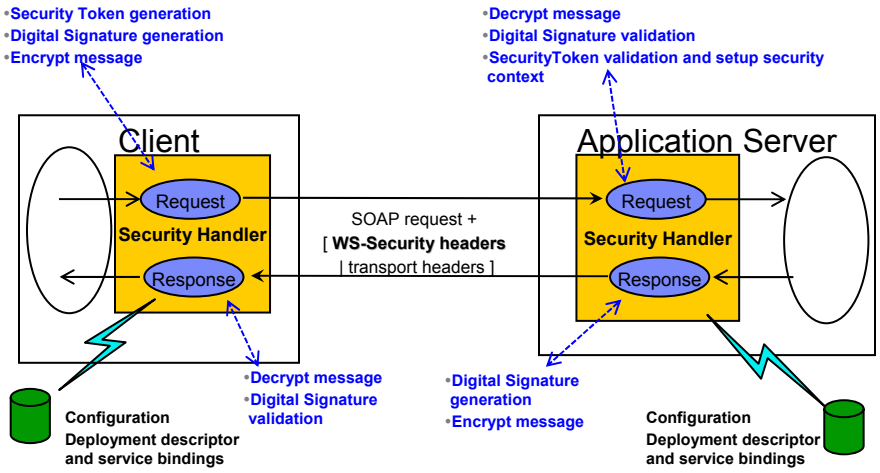


## WebSphere Support for Security Specifications

- **WebSphere V5.02 and V5.1**
  - WS-Security Draft 13
  - Username Token Profile Draft 0.2
  - X.509 BST Profile Draft 0.4
- **WebSphere V6.0**
  - WS-Security 1.0
  - Username Token Profile 1.0
  - X.509 BST Profile 1.0
- **Also based on specifications for XML Digital Signature and XML Encryption**

Redbooks IBM

## WS-Security High Level Architecture




 ON DEMAND BUSINESS

Redbooks IBM

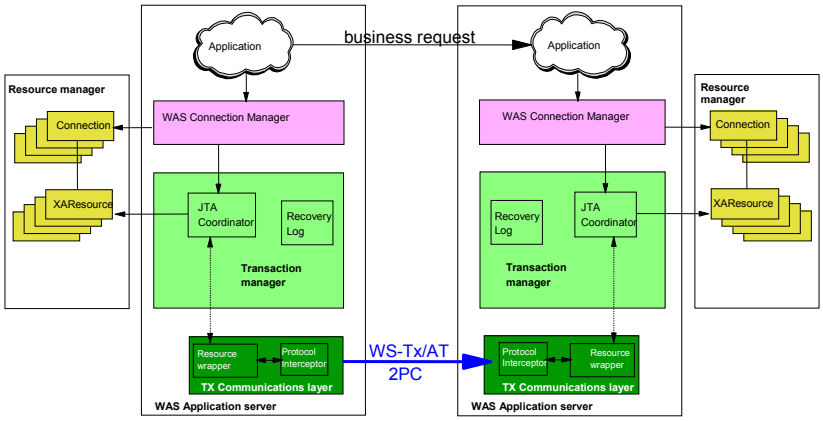
## WS-AtomicTransaction

- **Similar to IIOP transaction propagation**
- **WS-Address Endpoints exchanged as part of transaction context for protocol messages to be exchanged**
  - registration, commit coordination, etc…
- **Extensions to the DD model (via extensions)**
  - Settings to determine
    - Whether or not to propagate outbound transaction context
    - Whether to accept inbound transaction context

 ON DEMAND BUSINESS

Redbooks IBM

## WS-AtomicTransaction (continued)




 ON DEMAND BUSINESS

Redbooks IBM

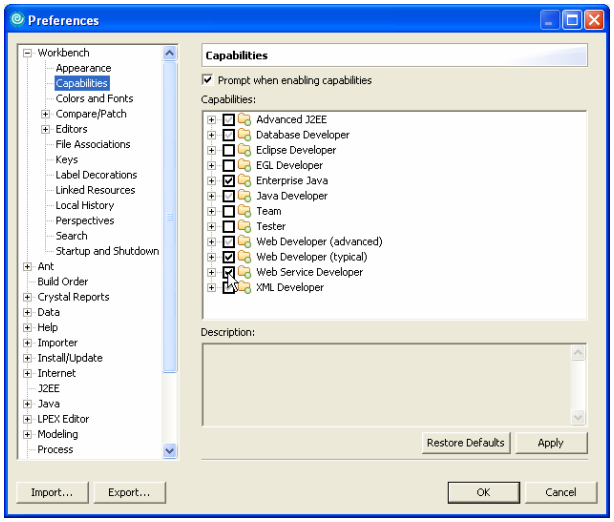
## Agenda

- **Technical overview**
- **Web services in WAS V6.01 for z/OS**
- **Developing a Web service in RAD**
- **Web services between WebSphere and CICS**
- **Web services between WebSphere and IMS**
- **Web services Between WebSphere and DB2**

 ON DEMAND BUSINESS

Redbooks IBM

## Enable Web Service Developer Capability in RAD

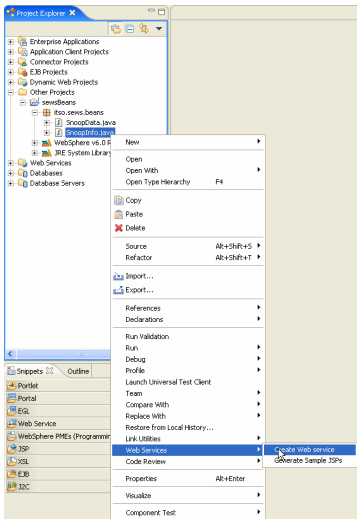




ON DEMAND BUSINESS

Redbooks IBM

## Create a Web Service out of an Existing Java Bean - 1

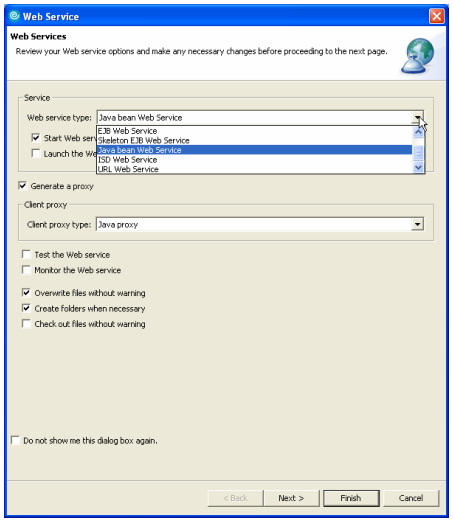




ON DEMAND BUSINESS

Redbooks IBM

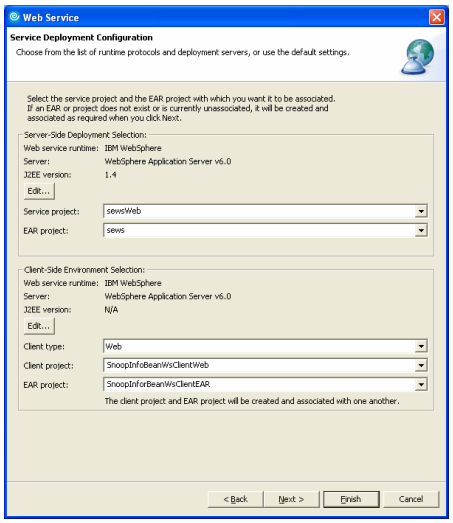
## Create a Web Service out of an Existing Java Bean - 2




 ON DEMAND BUSINESS

Redbooks IBM

## Create a Web Service out of an Existing Java Bean - 3

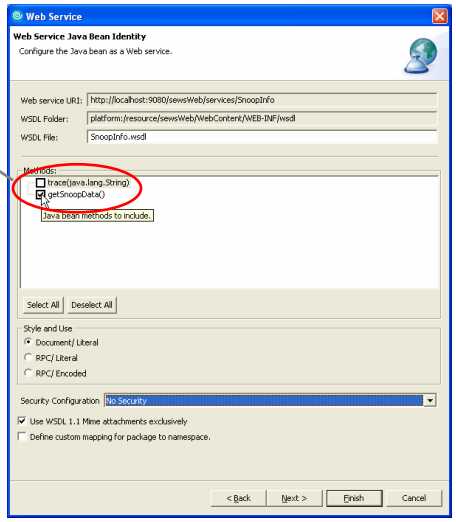



 ON DEMAND BUSINESS

Redbooks IBM

## Create a Web Service out of an Existing Java Bean - 4

Methods to be Web service enabled

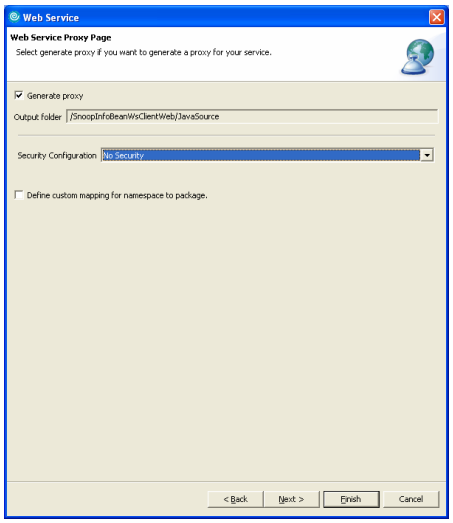





ON DEMAND BUSINESS

Redbooks IBM

## Create a Web Service out of an Existing Java Bean - 5



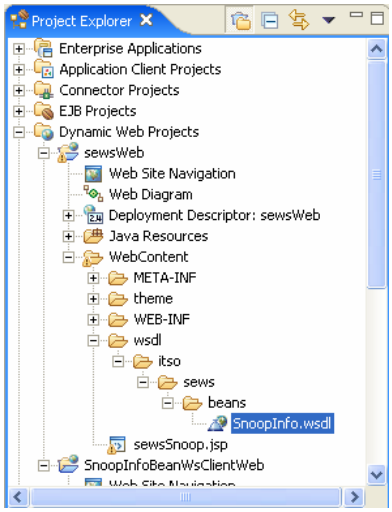



ON DEMAND BUSINESS

Redbooks IBM

## Where is the WSDL?

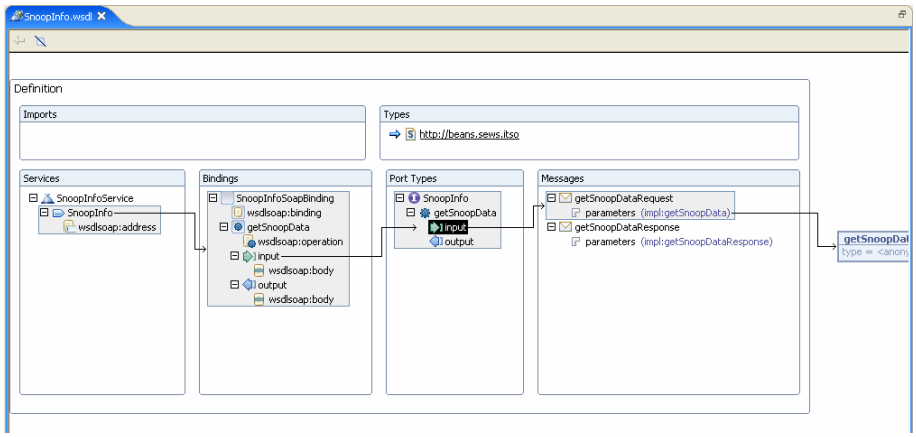
- **Resulting WSDL for the Web service enabled Bean can be found under Dynamic Web projects**
- **This WSDL is key for any client accessing this Web service**
  - i.e. a CICS client accessing this Web service over SOAP/HTTP
    - You then import this generated WSDL to create a mapping.




 ON DEMAND BUSINESS

Redbooks IBM

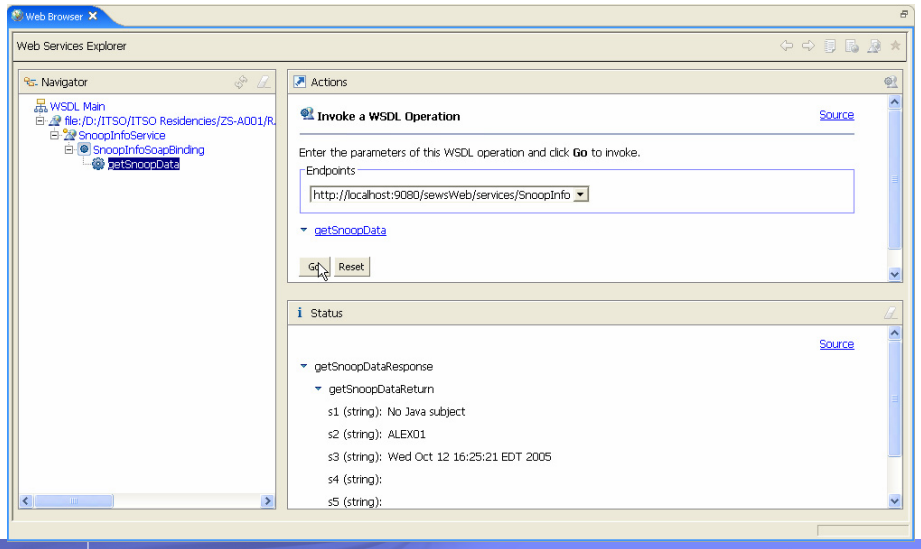
## WSDL – Graphical Representation




 ON DEMAND BUSINESS

Redbooks IBM

## Testing a Web Service in RAD

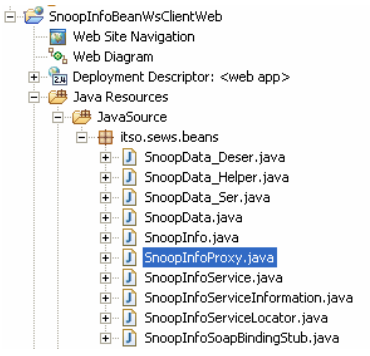



 ON DEMAND BUSINESS

Redbooks IBM

## Web Service Proxy

- **Besides the WSDL, another result of generating a Web service is a *Web service proxy*.**
- **This proxy can be used at a later time in any Java application to invoke the Web service.**




 ON DEMAND BUSINESS

Redbooks IBM

## Agenda

- **Technical overview**
- **Web services in WAS V6.01 for z/OS**
- **Developing a Web service in RAD**
- **Web services between WebSphere and CICS**
- **Web services between WebSphere and IMS**
- **Web services Between WebSphere and DB2**

 ON DEMAND BUSINESS

Redbooks IBM

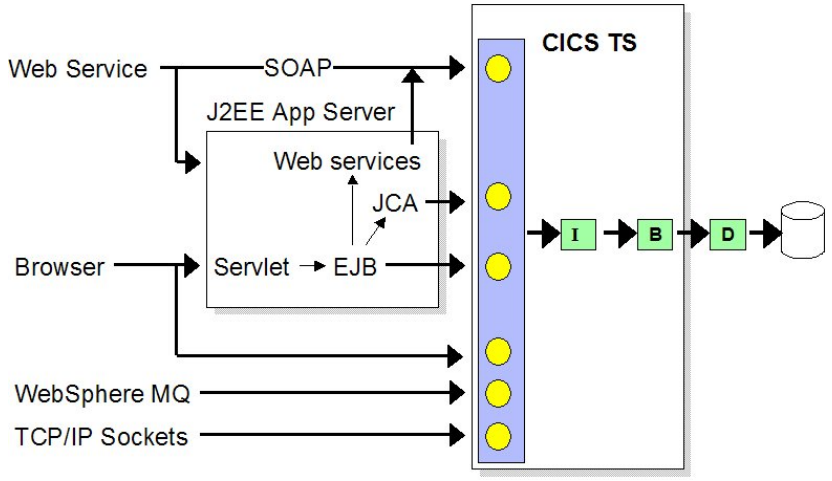
## WebSphere <-> CICS using SOAP - Overview

- **CICS and WebSphere can be both consumer and producer of a Web service**
- **CICS TS 3.1 has new SOAP support built-in**
  - For consuming a (remote) Web service from an existing or new CICS program
  - For Web service-fying an existing or new CICS program to become a Web service to the rest of the world
- **Supported protocols are HTTP and JMS, in both directions**

 ON DEMAND BUSINESS

Redbooks IBM

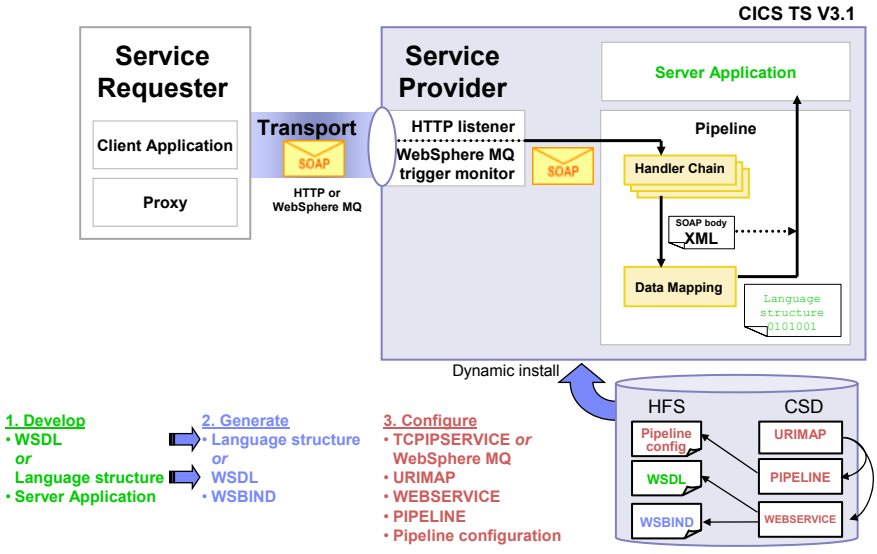
## Access to CICS - Options




 ON DEMAND BUSINESS

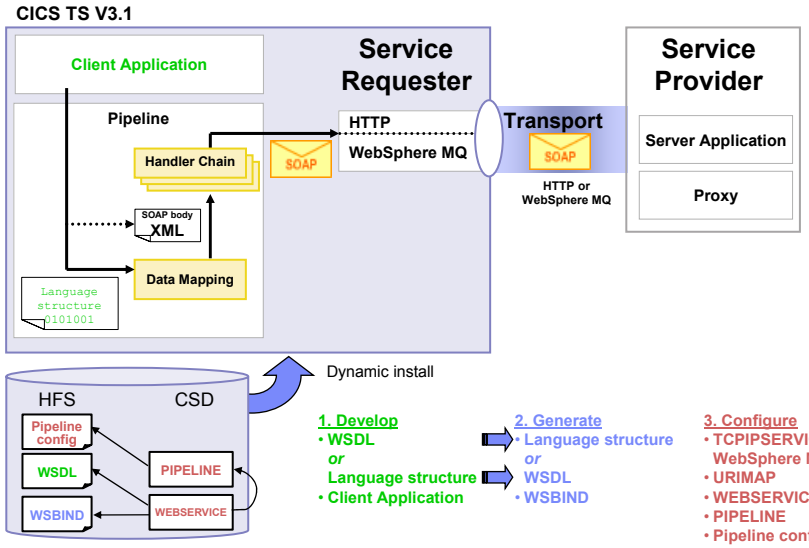
Redbooks IBM

## CICS as a Web service provider




 ON DEMAND BUSINESS

## CICS as a Web service requester

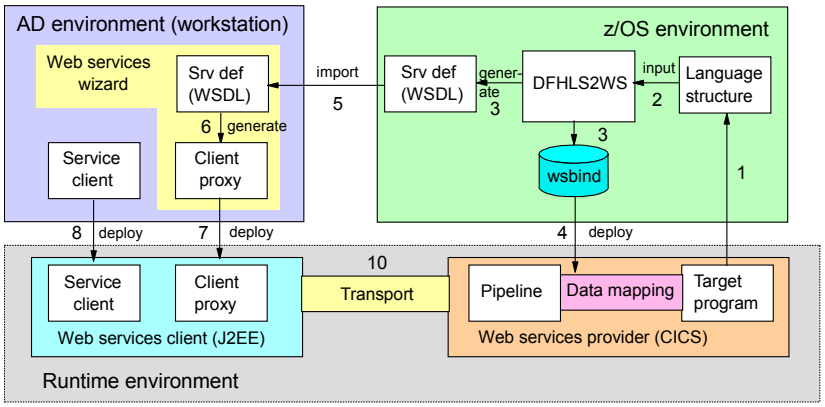





## CICS artifacts required to support Web Service

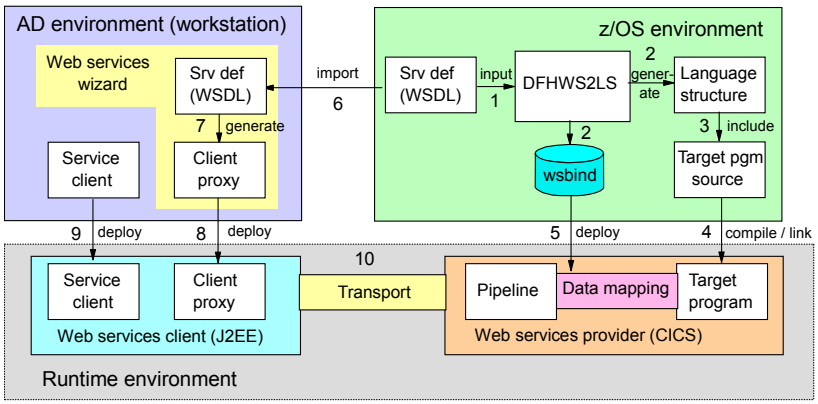
- **DFHPIPE**
  - Must be installed
- **TCPIP and TCPIP Service**
  - Should have been set up by the CICS sysprog
- **URIMAP**
  - Needed to define that a certain (HTTP) request is in fact a Web service request
- **Pipeline**
  - Need separate pipelines for incoming and outgoing Web services traffic
- **WEBS**
  - Is in fact the definition of a Web service provided or consumed by CICS

## CICS as Web Service Requestor – Bottom up Approach



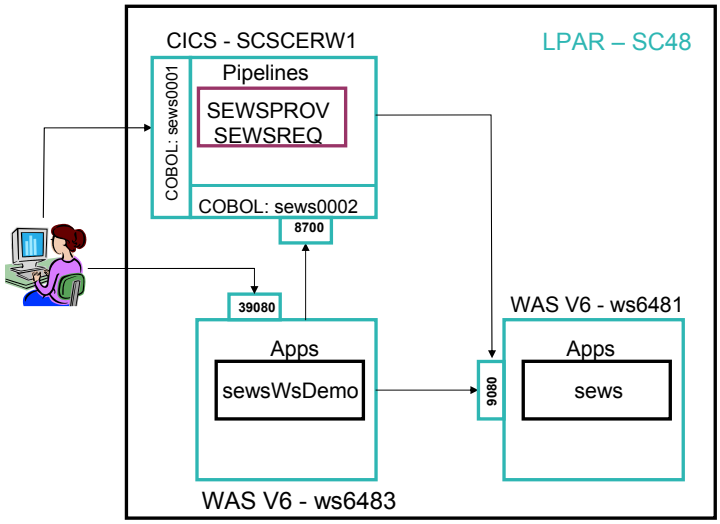



## CICS as Web Service Requestor – Top down Approach



Redbooks IBM

## Example using 2 WebSphere Cells and CICS and SOAP/HTTP




 ON DEMAND BUSINESS

Redbooks IBM

## Web service enable an (existing) CICS Program

- **Create bindings and WSDL**
- **Using a standard supplied JCL utility**
  - Called DFHLS2WS

```
//  JCLLIB ORDER=CICSTS31.CICS.SDFHINST
//*
//WS2LS     EXEC DFHLS2WS,
//    JAVADIR='java142s/J1.4',
//    USSDIR='cicsts31',
//    TMPFILE='cb',
//    PATHPREF=''
//INPUT.SYSUT1 DD *
PDSLIB=//EDMCAR.COPYLIB
LANG=COBOL
REQMEM=SEWSWP
RESPMEM=SEWSWR
PGMNAME=SEWS0002
URI=/service/snoopInfoCICS
PGMINT=COMMAREA
LOGFILE=/usr/cics/sews/wsprov/wsbind/sewsWsCics.log
WSBIND=/usr/cics/sews/wsprov/wsbind/sewsWsCics.wsbind
WSDL=/usr/cics/sews/wsprov/wsdl/sewsWsCics.wsdl
*/
```

 ON DEMAND BUSINESS

Redbooks IBM

## DFHLS2WS Parameters

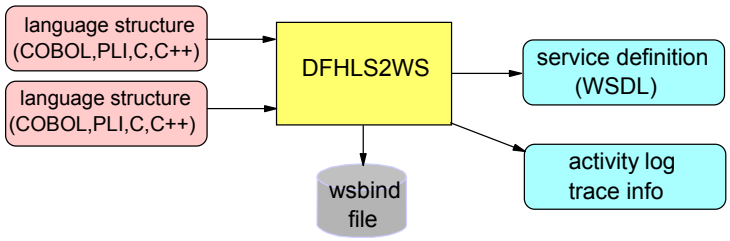
| Input Parameter | Purpose |
|---|---|
| PDSLIB | Name of a PDS which contains the request and response data structures |
| LANG | Identifies the programming language the generated code should be in |
| PGMINT | Indicates whether the program is passed a COMMAREA or a container |
| REQMEM | Name of PDS member that contains data structure that maps the request sent to the Web Service |
| RESPMEM | Name of PDS member that contains the data structure that maps the reply sent back in response |
| PGMNAME | This is the name of the program in CICS that will be invoked to process the Web Service requests when it is received by CICS |
| URI | This is the URI that will become part of the default end point address where the Web Service is located |
| LOGFILE | Location where log information from the execution of this program will be written |
| WSBIND | Location and name of file where the binding information will be written |
| WSDL | Location and name of the WSDL file that will be created by running this JCL |

 ON DEMAND BUSINESS

Redbooks IBM

## DFHLS2WS

- **Input**
  - Programming language data structures
    - In COBOL or PL/I or C or C++
    - Interface to the program can be COMMAREA or CHANNEL
- **Output**
  - Web service description (WSDL)
  - Web services bind file
  - Log file




 ON DEMAND BUSINESS

Redbooks IBM

## Next steps….

- **Once you have run the CICS utility, you will get a WSDL file as output**
- **Move the WSDL to your workstation and import it into a project in RAD**
  - The CICS generated WSDL file is in EBCDIC
- **Change the SOAP address location to the TCP/IP address and portnumber of the CICS server providing the Web service**
- **Generate a client application in RAD using the WSDL**
- **Now, you can deploy this application to WebSphere and invoke the Web service in CICS (using SOAP/HTTP)**


ON DEMAND BUSINESS

Redbooks IBM

## Invoking a Web service from CICS COBOL - API

```
EXEC CICS INVOKE WEBSERVICE(WS-WEBSERVICE-NAME)
          CHANNEL(WS-CHANNELNAME)
          URI(WS-ENDPOINT-Uri)
          OPERATION(WS-OPERATION)
          RESP(RESP)  RESP2(RESP2)
```

| Keyword | Purpose | Value |
|---|---|---|
| WEBSERVICE | Name of the Web Service resource that CICS will use to marshall and demarshal the request. | sews |
| CHANNEL | Name of the channel to pass to CICS. | sewsChl |
| URI | End point where the Web Service is located | default value is : http://localhost:9080/sewsWeb/services/SnoopInfo |
| OPERATION | The operation to be invoked in the Web Service | getSnoopData |


ON DEMAND BUSINESS

Redbooks IBM

## Calling a remote Web service from CICS - steps

- **Retrieve the WSDL describing the Web service you want to call**
  - i.e. the SnoopInfo.wsdl we explained earlier
- **Upload it (in ASCII) to z/OS and place it in a directory**
- **Run the supplied CICS utility DFHWS2LS with the WSDL as input**

```
//  JCLLIB ORDER=CICSTS31.CICS.SDFHINST
//*
//WS2LS     EXEC DFHWS2LS,
//    JAVADIR='java142s/J1.4',
//    USSDIR='cicsts31',
//    TMPFILE='cb',
//    PATHPREF=''
//INPUT.SYSUT1 DD *
PDSLIB=//EDMCAR.COPYLIB
LANG=COBOL
PGMINT=CONTAINER
REQMEM=SEWSRQ
RESPMEM=SEWSRS
LOGFILE=/usr/cics/sews/wsreq/wsbind/sews.log
WSBIND=/usr/cics/sews/wsreq/wsbind/sews.wsbind
WSDL=/usr/cics/sews/wsreq/wsdl/sews.wsdl
BINDING=SnoopInfoSoapBinding
*/
```

Use those to build the response and request in the COBOL program

 ON DEMAND BUSINESS

Redbooks IBM

## DFHWS2LS Parameters

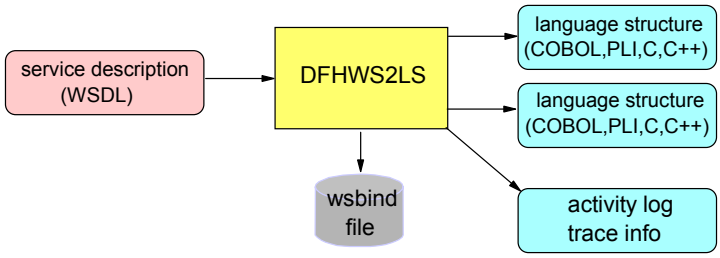
| Input Parameter | Purpose |
|---|---|
| PDSLIB | Name of a PDS into which will be saved generated code |
| LANG | Identifies the programming language the generated code should be in |
| PGMINT | Indicates whether the program is passed a COMMAREA or a container |
| REQMEM | Name of PDS member that will store the generated code that will map the data structure passed out on the request |
| RESPMEM | Name of PDS member that will store the generated code that will map the data structure returned from the Web Service call |
| LOGFILE | Location where log information from the execution of this program will be written |
| WSBIND | Location and name of file where the binding information will be written |
| WSDL | Location and name of the WSDL file that describes the Web Service |
| Binding | Is the name of the binding element in the WSDL file to use |
| PDSLIB | Name of a PDS into which will be saved generated code |

 ON DEMAND BUSINESS

Redbooks IBM

## DFHWS2LS

- **Input**
  - Web service description (WSDL)
- **Output**
  - High level language data structures
    - In COBOL or PL/I or C or C++
    - Interface to the program can be COMMAREA or CHANNEL
  - Web services bind file
  - Log file




 ON DEMAND BUSINESS

Redbooks IBM

## Agenda

- **Technical overview**
- **Web services in WAS V6.01 for z/OS**
- **Developing a Web service in RAD**
- **Web services between WebSphere and CICS**
- **Web services between WebSphere and IMS**
- **Web services Between WebSphere and DB2**

 ON DEMAND BUSINESS

Redbooks IBM

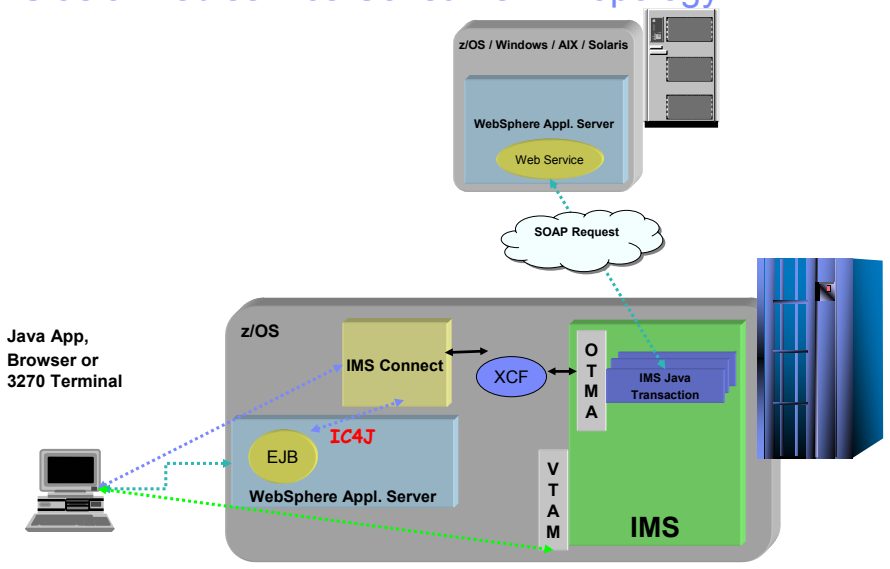
## WebSphere <-> IMS using Web services

- **No native SOAP listener in IMS**
- **IMS SOAP Gateway available soon**
  - Wraps Web service calls
  - Runs on the distributed platforms
- **IMS does not provide out of the box support to call Web services**
  - But you can develop a Java IMS transaction that acts as a Web service consumer



Redbooks IBM

## IMS as a Web service Consumer – Topology



Redbooks IBM

## IMS Requirements for SOAP Java client

- **Java FMID must be installed and IVP must have been succesfully executed**
- **A new transaction code must be defined for the Java client**
- **A PSB is required with LANG=JAVA**
- **An HFS directory is required to store/upload Java classes from the development environment**
- **DFSJMP in the PROCLIB must be tailored for the environment**
- **The JCL for a Java Message Processing (JMP) region must be tailored**
  - XPLINK parameter must be set to Y when using JDK 1.4.1
- **The Java PROCLIB members for the Environment and Master JVM must be tailored for use with a JDK level 1.4.1 or higher.**

 ON DEMAND BUSINESS

Redbooks IBM

## Developing an IMS Java SOAP client

- **Create a project/package for the new application**
- **Import the WSDL of the service to be called into RAD**
  - Make sure the address/location of the Web service reflects the TCP/IP address and portnumber of the WAS server
- **Import the IMS Java class (imsjava.jar) files into RAD (from z/OS) and those to the project classpath**
- **Create a basic IMS Java transaction**
  - A minimum of 2 Java classes is required
    - One for input/output message definitions
    - One for the Web service invocation code

 ON DEMAND BUSINESS

Redbooks IBM

## IMS Java Transaction calling a Web Service

```
public class IMSJavacallsWebService extends IMSApplication {
    public IMSJavacallsWebService()  {}
    public void doBegin() throws DLIException, IMSException  {
      IMSMessageQueue messageQueue = new IMSMessageQueue();
      IMSJavaMessage inputMessage = new IMSJavaMessage();
      IMSJavaMessage outputMessage = new IMSJavaMessage();
                messageQueue.getUniqueMessage(inputMessage);
                do {
                        try {
                                        SnoopInfoProxy proxy = new SnoopInfoProxy();
                                        SnoopData output = proxy.getSnoopData();
                                        outputMessage.setString("Message", "S1: " + output.getS1());
                                        messageQueue.insertMessage(outputMessage);
                                        outputMessage.setString("Message", "S2: " + output.getS2());
                                        messageQueue.insertMessage(outputMessage);
                                        outputMessage.setString("Message", "S3: " + output.getS3());
                                        messageQueue.insertMessage(outputMessage);
                                        outputMessage.setString("Message", "S4: " + output.getS4());
                                        messageQueue.insertMessage(outputMessage);
                                        outputMessage.setString("Message", "S5: " + output.getS5());
                                        messageQueue.insertMessage(outputMessage);
                                        outputMessage.setString("Message", "S6: " + output.getS6());
                                        messageQueue.insertMessage(outputMessage);
                        } catch (Exception e) {
                                                System.out.println("\nCaught exception is: " + e);
                                                e.printStackTrace();
                                                outputMessage.setString("Message", "\nCaught exception is: " + e);
                                                messageQueue.insertMessage(outputMessage);
                        }
        IMSTransaction.getTransaction().commit();
                } while (messageQueue.getNextMessage(inputMessage));
    }

    public static void main(String args[])  {
             IMSJavacallsWebService test = new IMSJavacallsWebService();
        test.begin();
    }
}
```

 ON DEMAND BUSINESS

Redbooks IBM

## Running the Java SOAP client in IMS

- **Add WebSphere V6 for z/OS runtime jars to the IMS classpath**
  - Using -Djava.ext.dirs parameter of the master and worker JVM config. Member
  - Do not try this with the WAS V6 distributed jars as this will not work due to codepage problems
- **Add BBOLOAD dataset of WAS V6 to the STEPLIB of the IMS Java region**
- **Edit the master JVM PROCLIB member, to include:**
  - imsjava.jar
  - the jar file containing the Java transaction

 ON DEMAND BUSINESS

Redbooks IBM

## IMS PROCLIB – Master JVM Member

```
-Dibm.jvm.shareable.application.class.path=>
/u/denis/imsjava_transaction.jar
-Dibm.jvm.trusted.middleware.class.path=>
/usr/lpp/imsv9/imsjava91/imsjava.jar
-Djava.ext.dirs=/usr/lpp/java/J1.4/lib/ext:>
/usr/lpp/zWebSphereAL1/V6R0/lib:>
/usr/lpp/zWebSphereAL1/V6R0/installedChannels
-Xinitacsh4M
-Xinitsh4M
-Xinitth8M
-Xmaxf0.6
-Xmine8M
-Xminf0.3
-Xms8M
-Xmx64M
-Xoss400k
```

 ON DEMAND BUSINESS

Redbooks IBM

## IMS PROCLIB – Worker JVM Member

```
-Djava.ext.dirs=/usr/lpp/java/J1.4/lib/ext:>
/usr/lpp/zWebSphereAL1/V6R0/lib:>
/usr/lpp/zWebSphereAL1/V6R0/installedChannels
-Xmaxf0.6
-Xminf0.3
-Xms128M
-Xmx1024M
-Xoss400k
-Xinitth512M
```

 ON DEMAND BUSINESS

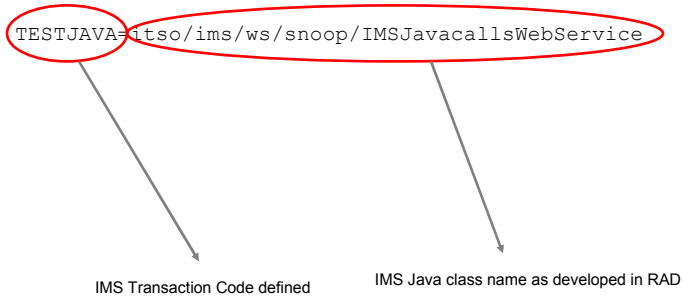
## IMS PROCLIB – USS Settings

```
PATH=/usr/lpp/java/J1.4/bin:.:/bin:>
/u/denis
LIBPATH=/usr/lpp/imsv9/imsjava91:>
/usr/lpp/java/J1.4/bin:>
/usr/lpp/java/J1.4/bin/classic:>
/usr/lpp/zWebSphereAL1/V6R0/lib
```



## Mapping IMS Transaction Code and Java Class

- **Mapping kept in DFSJVMAP member of IMS PROCLIB**



Redbooks IBM

## IMS Stage1 and PSB Definition for Java Transaction

```
APPLCTN PSB=TESTJAVA PGMTYPE=TP,SCHDTYP=PARALLEL
            TRANSACT CODE=TESTJAVA,MODE=SNGL,EDIT=(ULC),
X
                MSGTYPE=(MULTSEG,RESPONSE,1),MAXRGN=0,
X
                PARLIM=0,PROCLIM=(10,100)
          SPACE 2
```

STAGE1

```
ALTERPCB PCB     TYPE=TP,MODIFY=YES
          PSBGEN LANG=JAVA,PSBNAME=TESTJAVA
          END
```

PSB

 ON DEMAND BUSINESS

Redbooks IBM

## Agenda

- **Technical overview**
- **Web services in WAS V6.01 for z/OS**
- **Developing a Web service in RAD**
- **Web services between WebSphere and CICS**
- **Web services between WebSphere and IMS**
- **Web services Between WebSphere and DB2**

 ON DEMAND BUSINESS

Redbooks IBM

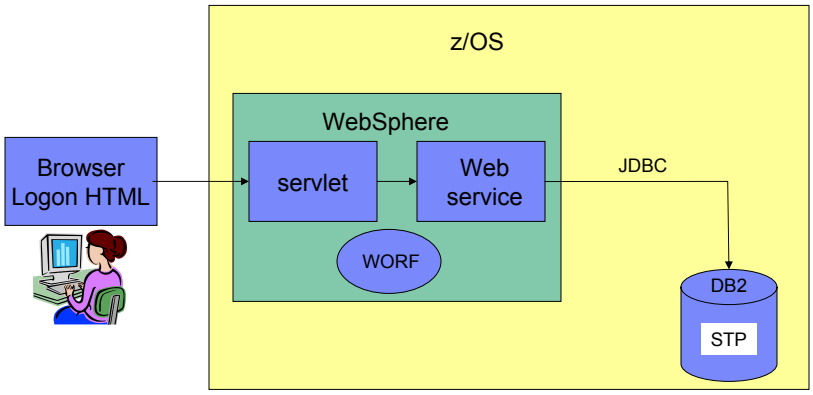
## WebSphere <-> DB2 using Web services

- **DB2 Stored Procedures can be called as a Web service from a J2EE application in WebSphere**
  - Using a framework in WebSphere called “WORF”
    - Wraps a JDBC call to a stored procedure into a Web service
  - DB2 does not provide a “native” SOAP listener
- **Web services in WebSphere can be called from DB2 using a set of user defined functions**
  - Works from any DB2 program

 ON DEMAND BUSINESS

Redbooks IBM

## Calling a Stored Procedure - Sample Topology




 ON DEMAND BUSINESS

Redbooks IBM

## Web Services Object Runtime Framework (WORF) for DB2

- **WORF is a framework using:**
  - Simple Object Access Protocol (SOAP) 2.2 or later
  - Document Access Definition Extension (DADX)
    - A DADX document specifies a Web Service using a set of operations that are defined by SQL statements or XML Extender Document Access Definition (DAD) documents.
- **WORF is part of DB2 V8.1 and later and must be installed into WebSphere on z/OS**
  - WORF is not part of WAS for z/OS and only supported on z/OS as of WAS V5.1
  - The worf.jar file must be placed in the working directory of the application server
    - For example, /SC48/WebSphereAL1/V6R0/BS01/AppServer/lib/ext/
  - JavaMail and Java Beans Activation Framework libraries must be available in WAS too
    - mail.jar and activation.jar

 ON DEMAND BUSINESS

Redbooks IBM

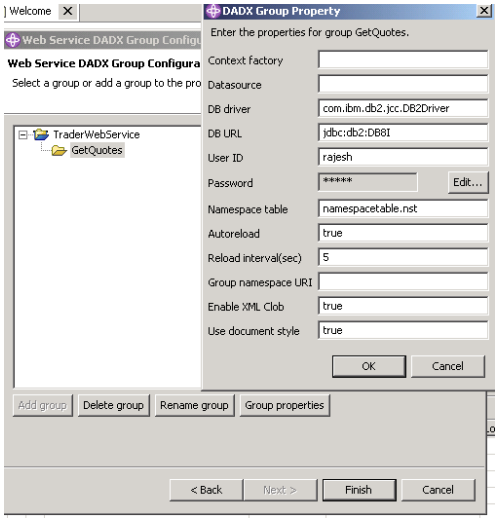
## Developing a Web Service to access DB2 – DADX File

- **Key in developing a Web Service accessing DB2 is a so-called "DADX file"**
  - Create a "Web service DADX group configuration".
    - The creation of a Web service DADX group configuration can be reached from File → New → Other → Web Services
  - Create the actual DADX file
    - The creation of a DADX file can be reached from File → New → Other → Web Services
    - The DADX file can be created "live" using a connection to a stored procedure in DB2 on z/OS, based on the properties set in in the Web service DADX group configuration
    - The DADX file can also be created manually in WSAD or RAD

 ON DEMAND BUSINESS

Redbooks IBM

## Web Service DADX Group Configuration - Properties




 ON DEMAND BUSINESS

Redbooks IBM

## DADX File – manually provided

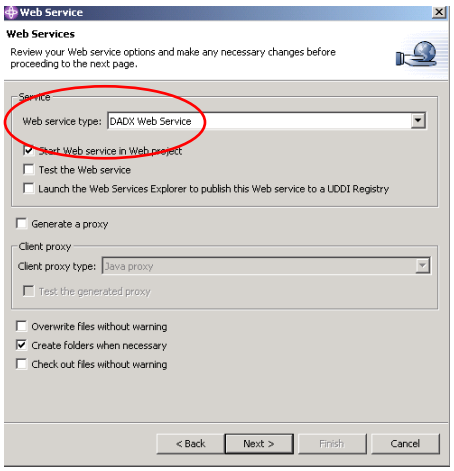
```
<?xml version="1.0" encoding="UTF-8"?>
<dadx:DADX xmlns:dadx="http://schemas.ibm.com/db2/dxx/dadx"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<dadx:result_set_metadata  name="resultSet1Metadata" rowName=
"resultSet1MetadataRow">
         <dadx:column name="COMPANY"
                  type="CHAR"
                  nullable="false"
                  as="COMPANY" />
.................specifying all fields.....................
         </dadx:result_set_metadata>
         <dadx:operation name="GETQUOTES">
         <dadx:documentation xmlns="http://www.w3.org/1999/xhtml">
         <![CDATA[
         ]]>
         </dadx:documentation>
                  <dadx:call>
                           <dadx:SQL_call>
                           <![CDATA[
                           CALL RAJESHPR.GETQUOTES()
                           ]]>
                           </dadx:SQL_call>
                  <dadx:result_set name="resultSet1" metadata="resultSet1Metadata"/>
                  </dadx:call>
         </dadx:operation>
</dadx:DADX>
```

 ON DEMAND BUSINESS

Redbooks IBM

## Developing a Web Service to access DB2 – The Web Service

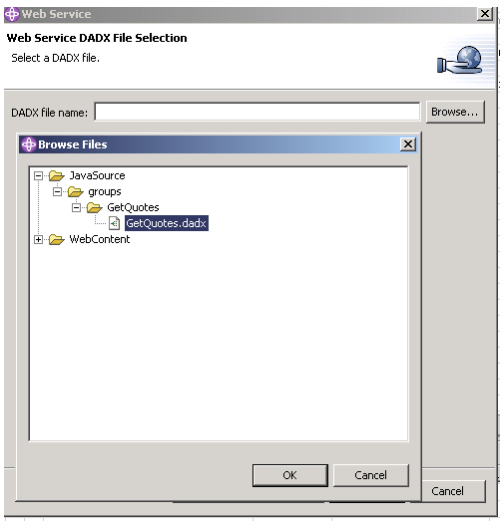
- **Once the DADX file is created, it will be used in the generation of the Web service**
  - Select File → New → Web services → Web service




 ON DEMAND BUSINESS

Redbooks IBM

## DADX File Selection when creating the Web Servive




 ON DEMAND BUSINESS

Redbooks IBM

## Calling a Web service from DB2

- **User Defined Functions (UDFs) allow you to consume Web services in SQL statements**
- **The SQL statements for calling the Web service can be used from a variety of environments, such as SPUFI, stored procedures or (batch) programs**
- **Upon receipt of the results from a Web service call, the SOAP envelope is stripped and the remaining XML document can be processed in a variety of ways:**
  - Insert into or update a table directly
  - Send to the XML extender for decomposing
  - Forward to an MQ queue

 ON DEMAND BUSINESS

Redbooks IBM

## Prereqs and setup for DB2 as Web Service Consumer

- **Prereqs:**
  - z/OS 1.2 or later
  - IBM XML Toolkit Version 1.6 for z/OS
  - The UDFs must run in a WLM-Managed address space
- **Setup:**
  - The job DSNTIJWS (provided in the SDSNSAMP data set) can be used to issue the CREATE FUNCTION statements for the Consumer UDFs. This job will need to be customized to fit the parameters of your system. The UDFs are installed under DB2XML schema.
  - Ensure that the XML Toolkit is added to the STEPLIB of the startup procedure for the WLM Environment used for the UDFs.

 ON DEMAND BUSINESS

Redbooks IBM

## Web Service Consumer Format

```
DB2XML.SOAPHTTPV(IN endpoint_url VARCHAR(256),
IN soap_action VARCHAR(256),
IN soap_body VARCHAR(3072))
RETURNS VARCHAR(3072)

DB2XML.SOAPHTTPV(IN endpoint_url VARCHAR(256),
IN soap_action VARCHAR(256),
IN soap_body CLOB(1M))
RETURNS VARCHAR(3072)

DB2XML.SOAPHTTPC(IN endpoint_url VARCHAR(256),
IN soap_action VARCHAR(256),
IN soap_body VARCHAR(3072))
RETURNS CLOB(1M)

DB2XML.SOAPHTTPC(IN endpoint_url VARCHAR(256),
IN soap_action VARCHAR(256),
IN soap_body CLOB(1M))
RETURNS CLOB(1M)
```

- ***endpoint_url***
  - The URL of the Web service
- ***soap_action***
  - Is used as part of the SOAP request. Depending on the Web server, this parameter can be optional. If it is required, the required value can be found in the WSDL of the Web service.
- ***soap_body***
  - The operation name and the parameters to the Web service. The data in this parameter should be well-formed XML.

 ON DEMAND BUSINESS

Redbooks IBM

## Calling a Web Service from DB2 - Example

```
SELECT
    DB2XML.SOAPHTTPV(
'http://wtsc48.itso.ibm.com:29080/TraderWebService/GetQuotes/GetQuotes.d
adx/SOAP', 'http://tempuri.org/GetQuotes/GetQuotes.dadx',
'<GETQUOTES xmlns="http://tempuri.org/GetQuotes/GetQuotes.dadx"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<COMPANY>IBM</COMPANY></GETQUOTES>') FROM SYSIBM.SYSDUMMY1;
```

**returns:**

```
<GETQUOTESResponse xmlns="http://tempuri.org/GetQuotes/GetQuotes.dadx"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<resultSet1><resultSet1Metadata xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<resultSet1MetadataRow> <COMPANY>IBM                </COMPANY>
<SHARE_PRICE>163.0</SHARE_PRICE>
<UNIT_VALUE_7DAYS>157.0</UNIT_VALUE_7DAYS>
<UNIT_VALUE_6DAYS>156.0</UNIT_VALUE_6DAYS>
<UNIT_VALUE_5DAYS>159.0</UNIT_VALUE_5DAYS>
<UNIT_VALUE_4DAYS>161.0</UNIT_VALUE_4DAYS>
<UNIT_VALUE_3DAYS>160.0</UNIT_VALUE_3DAYS>
<UNIT_VALUE_2DAYS>162.0</UNIT_VALUE_2DAYS>
<UNIT_VALUE_1DAYS>163.0</UNIT_VALUE_1DAYS>
<COMM_COST_SELL>10</COMM_COST_SELL> <COMM_COST_BUY>15</COMM_COST_BUY>
</resultSet1MetadataRow> </resultSet1Metadata></resultSet1>
</GETQUOTESResponse>
```

 ON DEMAND BUSINESS

Redbooks IBM

## Example with parsing the XML Results

- **The following statement will also parse the XML document**

```
WITH SOAPOUTPUT(OUT) AS (
 SELECT
    DB2XML.SOAPHTTPV(
'http://wtsc48.itso.ibm.com:29080/TraderWebService/GetQuotes/GetQuotes.d
adx/SOAP', 'http://tempuri.org/GetQuotes/GetQuotes.dadx',
'<GETQUOTES xmlns="http://tempuri.org/GetQuotes/GetQuotes.dadx"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<COMPANY>IBM</COMPANY></GETQUOTES>')  FROM SYSIBM.SYSDUMMY1 )
SELECT DB2XML.EXTRACTCHAR(DB2XML.XMLVARCHAR(OUT),
'/GETQUOTESResponse/resultSet1/resultSet1Metadata/resultSet1MetadataRow/
COMPANY') from soapoutput
```

 ON DEMAND BUSINESS

Redbooks IBM

# Module 5

# J2EE Connector Architecture

 ON DEMAND BUSINESS

Redbooks IBM

## Agenda

- What is J2CA and why should I care
- Tooling and developing applications for CICS and IMS using J2CA
- Using J2CA to access CICS
- Using J2CA to access IMS

 ON DEMAND BUSINESS

Redbooks IBM

## IBM Transaction Servers

- Over 30 years and $1 Trillion invested in Applications ... IDC
- Over $1 trillion processed/day
- Over 30 billion transactions/day
- Most people use CICS
  - 85% penetration on S/390 + zSeries; 490 of top 500 IBM customers
- Among the highest revenue earning software products in the world


- Most Corporate Data is Managed by IMS
  - Over 95% of Fortune 1000 Companies use IMS
  - IMS Manages over 15 Million GBs of Production Data
  - $2.5 Trillion/day transferred through IMS by one customer
- Over 50 Billion Transactions a Day run through IMS
  - Serves Close to 200 Million Users a Day
  - Over 100 Million IMS Trans/Day Handled by One Customer
  - 120M IMS Trans/day, 7M per hour handled by another customer
  - 21,000 Trans/sec (near w1 Billion/day) with IMS Data/Queued sharing on a single processor with database updates
  - 6000 Trans/sec across TCP/IP to single IMS


 ON DEMAND BUSINESS

## Enterprise Modernization Products Today in IBM

| Networks | Improve | Transform | Innovate |
|---|---|---|---|
| Comms Server | HOD<br>HATS<br>Host Publisher<br>CICS 3270 Web Bridge | Host Publisher<br>HACL/HOD Beans<br>CICS TG<br>IMS Connect<br>SOAP for CICS<br>MQIAC<br>DPL and FEPI<br>Link3270 Bridge<br>WDS | WSAA<br>WSED<br>WSAD-IE<br>WDS<br>RAD<br>RD for z |

**Enterprise Modernization**

- IBM's EM landscape is evolving
- Products, functionality, and solutions overlap and compete
- Lack of consistency at technical & marketing level
- No migration through phases of EM
- Customer confusion on overlapping solutions - lack of EM roadmap



## The Reason for Connectors

- **Simplify integration of diverse EIS**
  - standardized interface for client and resource portable across all compliant J2EE servers
  - facilitate scalable architectures
- **Through the resource adapters connectors provide Quality of Service features transparently to the client application**
- **Flexibility**
  - one application server can support many resource adapters
  - many application servers can support a standard resource adapter

Redbooks IBM

## Qualities of Service, what are they?

- **In addition to simple access, the connector architecture should allow for Quality of Service (QoS) features:**
  - connection creation and destruction
  - connection pooling
  - transaction management
  - security handling
  - error tracing
  - logging
- **In a *managed connection* the application server provides QoS**
  - specified by J2EE standard
- **In a *non-managed* connection the application manages QoS**

 ON DEMAND BUSINESS

Redbooks IBM

## Connectors, what are they?

- **Objectives**
  - make the communication protocol / mechanism between a client application and a subsystem transparent to the application developer
  - provide portability of the client application
- **Some connectors can be used**
  - Locally
    - client and target subsystem reside on the same system
    - some form of optimization in the communication
  - Remote
    - client resides on another partition or machine than the target subsystem
    - communication is taking place over a protocol such as TCP/IP
- **The usage of connectors in application programs is supported by a structured API and development tools**
- **Connectors are a key component in most e-business architectures on zSeries**
- **Using connectors is even made easier for the programmer by using the J2EE Connector Architecture (J2CA)**

 ON DEMAND BUSINESS

Redbooks IBM

## J2CA – Local vs. Remote Connections

| | Local connection | Remote connection |
|---|---|---|
| Performance | Does not include network overhad | Includes network overhed |
| Availability | All components on one LPAR, imposing some limitations on workload balancing. | Components can be distributed across LPARs or systems, allowing for a better workload balancing. |
| Scalability | Scaling achieved by duplicating all components on another LPAR | Scaling can be achieved by duplicating selective components on the same or another LPAR |
| Security | Thread identity (ACEE) can be used.<br>As no network traffic is involved, connections are les risky. | No thread identity.<br>Connections over the network are more risky and require more security measures. |
| Transactionality | 2-PC supported (RRS enforced) | Not always 2-PC possible. |

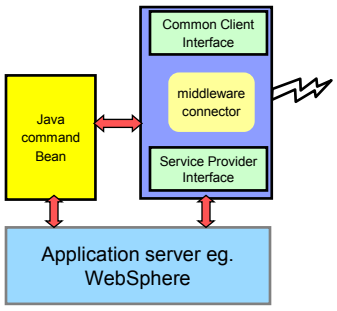
 ON DEMAND BUSINESS

Redbooks IBM

## What does J2EE Connector Architecture provide?

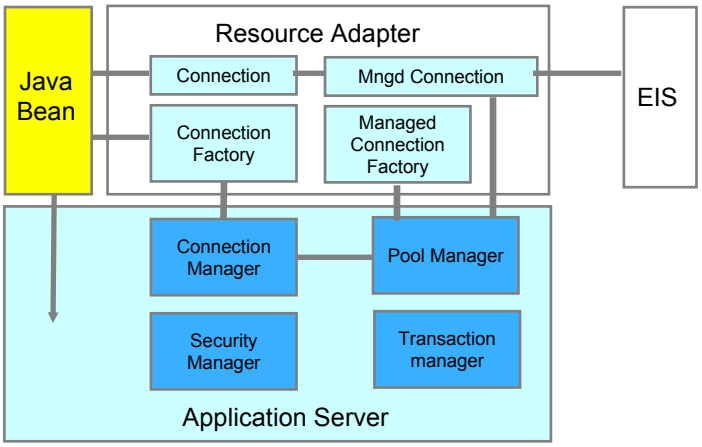
- **A common programming interface for Java clients**
  - make interfaces independent of resource, flow or data structure details
  - hide connection details from application programs
- **A common Service Provider Interface (SPI) for Application Servers**
  - for pooling, transactions & security
- **Portability**
  - across any JCA-compliant Application Server
- **Base for extensions**
  - e.g. exploiting XML and SOAP

- A ***Connector*** is generic runtime code, such as a J2EE architected connector, that transforms one calling interface into another
- An ***Adapter*** is runtime code, ***possibly generated by a tool***, that converts one data format to another (e.g. converts a bean format into a CICS COMMAREA)
- Many solutions will use ***both*** connectors and adapters



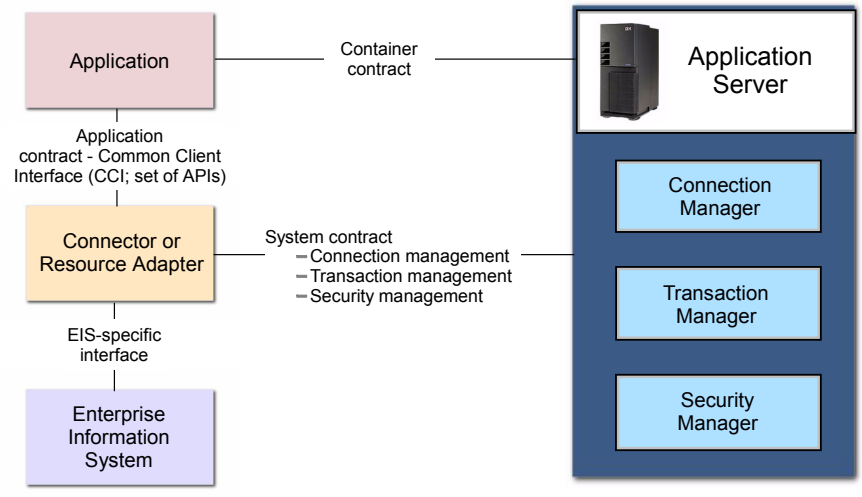

 ON DEMAND BUSINESS

## J2EE Connector Architecture (overview)



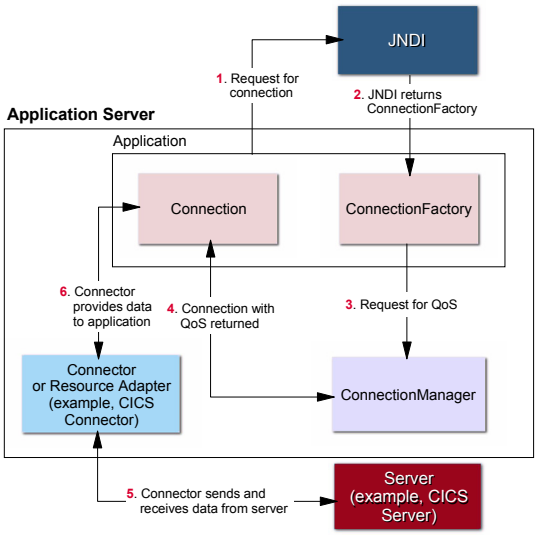



## J2CA Architecture



Redbooks IBM

## J2CA Architecture - Flow




 ON DEMAND BUSINESS

Redbooks IBM

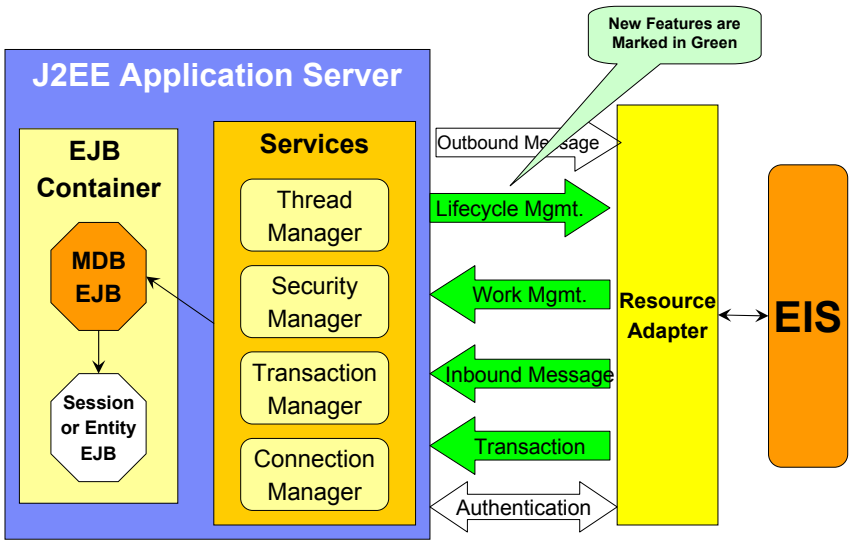
## JCA 1.5 – New Features

- **Inbound Communication**
  - Allows external EIS systems to communicate with the application components deployed on the Application Server
- **Transaction Inflow**
  - Allows a Resource Adapter to propagate an existing two phase transaction into an Application Server
- **Message Inflow**
  - Generic contract to plug-in a variety of Message Providers into an Application Server

 ON DEMAND BUSINESS

Redbooks IBM

## New Features in JCA 1.5

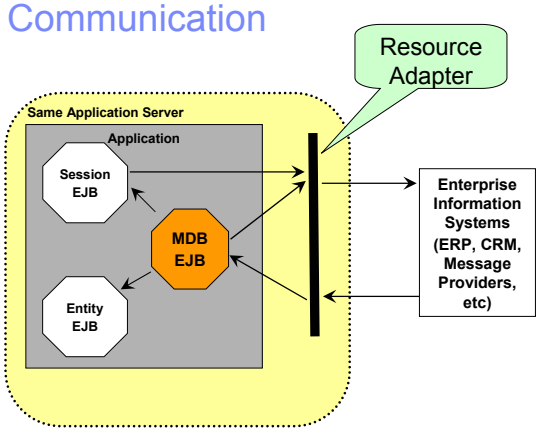



 ON DEMAND BUSINESS

Redbooks IBM

## Inbound and Outbound Communication

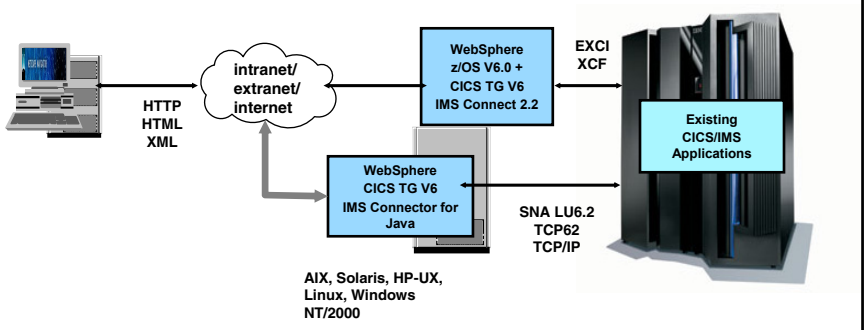
- **Introduces inbound and outbound communication between EIS and EJB Container**
  - JCA 1.0 supported only outbound communication (EJB to EIS)
- **Allows EIS to inject work into the J2EE container with a transactional context initiated by the EIS.**




 ON DEMAND BUSINESS

Redbooks IBM

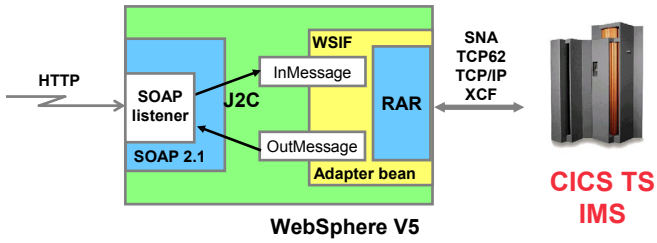
## Accessing CICS programs or IMS transactions




 ON DEMAND BUSINESS

Redbooks IBM

## Exploit CICS/IMS Applications as Web Services




- **Applications using WSAD-IE V5.x:**
  - Use WSIF (Web Services Invocation Framework) and SOAP 2.1 supplied with WebSphere App Server V5
  - WebSphere Studio generates Adapter Bean
- **Applications developed with RAD V6 do not use WSIF anymore!**
- **All apps use J2CA Connector provided by CTG and IMS Connect**

 ON DEMAND BUSINESS

Redbooks IBM

## Agenda

- **What is J2CA and why should I care**
- **Tooling and developing applications for CICS and IMS using J2CA**
- **Using J2CA to access CICS**
- **Using J2CA to access IMS**

 ON DEMAND BUSINESS

Redbooks IBM

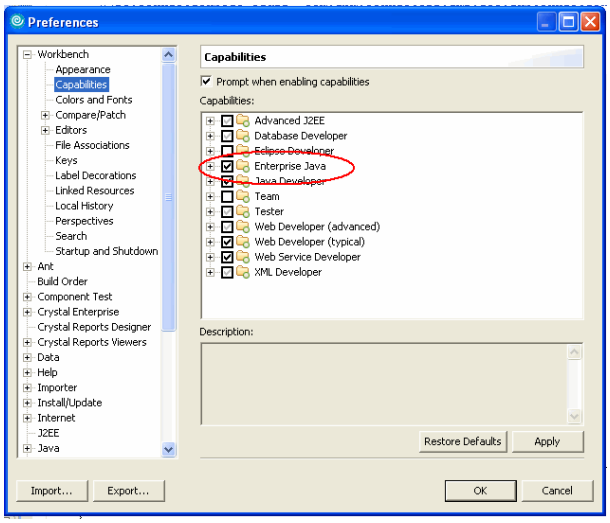
## Tooling overview

- **Most current tool for WebSphere Version:**
  - Rational Application Developer Version 6
  - Rational Software Architect Version 6
- **RAD Version 6 does not use WSIF anymore for generating J2CA applications**
- **Applications generated with WSAD-IE V5.1 work though when deployed to WebSphere V6**
  - WSIF framework is still available
- **New applications should be developed with RAD Version 6**

 ON DEMAND BUSINESS

Redbooks IBM

## Adding the J2CA feature to RAD




 ON DEMAND BUSINESS

Redbooks IBM

## Development process for a J2CA application in RAD

- **Create a J2C Java Bean**
  - Choice between the “ECI” (CTG) and “IMS” (IMS Connect) resource adapter
  - Logical JNDI name is specified in this step
- **Add a method to th J2C Bean**
- **Define input and output types for the method**
  - Represents the commarea
- **Generate Java classes for the input and output**
  - Import the COBOL copybook representing the commarea
  - Specify attributes like target codepage and platform
- **Set connector properties**

 ON DEMAND BUSINESS

Redbooks IBM

## Accessing an EIS using J2CA

- **Lookup a ConnectionFactory for the resource adapter**
- **Create a Connection object to the EIS using a ConnectionFactory object.**
  - The Connection object represents the connection to the EIS
  - Used for subsequent interactions with the EIS
  - Connection properties (like user name, password, etc.) can be passed via a ConnectionSpec object
- **Create an Interaction object**
  - The Interaction object describes the behavior or property of this specific interchange with the EIS
  - Interaction properties (like functionName,ItermName,interactionVerb) can be passed via an InteractionSpec
- **Create Record instances for transferring data in/out**
- **Perform the desired function (Interaction.execute(...))**
- **Process output and close the connection**

 ON DEMAND BUSINESS

Redbooks IBM

## Connections

- **Managed connections**
  - The application server handles all aspects of the connection. The application server handles the Quality of Service (QoS). This includes, for example, looking up a connection factory instance, getting an EIS connection, and finally closing the connection.
- **Non-managed connections**
  - In a non-managed application scenario, the application developer follows a similar programming model to the managed application scenario, but must handle all aspects of the connection within the application code.

 ON DEMAND BUSINESS

Redbooks IBM

## J2EE Connector security

- **It all depends on an extended deployment descriptor (xdd) tag:**
  - res-auth
- **res-auth = Container**
  - The container will pass the required identity and credential to the connector
    - User ID/password encoded in a J2C Container-managed alias or provided in Managed Connection Factory
  - If the application codes, for example, getConnection(userid,password), the passed identity and credential will be ignored
- **res-auth = Application**
  - The application is expected to provide the required identity and credential, for example, getConnection(userid,password)
  - If no userid is supplied J2C defaults to userid defined in J2C Application-managed alias

 ON DEMAND BUSINESS

Redbooks IBM

## Security: Thread Identity Support - z/OS Only

- **Thread identity associated with current thread used as connection owner**
- **Used only when res-auth = Container and no Container-managed alias specified**
- **Dependent on Resource Adapter and whether Local or Remote**
- **Only implemented by following resources adapters:**
  - IMS Connector for Java (local only)
  - CTG CICSECI Connector (local only)
  - IMS JDBC Connector
  - RRA DB2 390 Local JDBC provider
- **Synch-to-OS-thread**
  - Sets the OS task (thread) ACEE to the J2EE RunAs identity
  - Only implemented by following resources adapters:
    - IMS JDBC Connector
    - RRA DB2 390 Local JDBC provider

 ON DEMAND BUSINESS

Redbooks IBM

## RunAs: Sets the principal used for this connection

- **Caller**
  - Run this method with the identity of the user who instantiated me
- **Server (the default)**
  - Run this method with the identity of the server on which I was instantiated
- **Role**
  - Run this method with the RACF ID associated with this roles RACF EJBROLE/GEJBROLE profile
  - Role name is mapped to RACF ID via the APPLDATA property in the EJBROLE/GEJBROLE profile

 ON DEMAND BUSINESS

Redbooks IBM

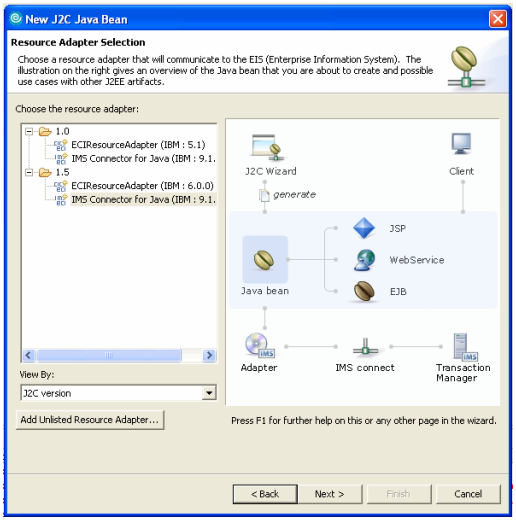
## Common Client Interface - sample

```
// JNDI lookup of a Connection Factory
InitialConext ic = new InitialContext();
ConnectionFactory connectionFactory = (ConnectionFactory)
ic.lookup("EIS");

// Create a Connectionspec object
ConnectionSpec connectionSpec = new ConnectionSpec(user,password);
// Create a Connection object using the above ConnectionSpec object
Connection connection =
connectionFactory.getConnection(connectionSpec);

// Create an InteractionSpec object
InteractionSpec interactionSpec = new InteractionSpec();
interactionSpec.setUserName("USERID");
interactionSpec.setPassword("PASSWORD");
// Create an interaction using the above InteractionSpec object
Interaction interaction = conn.createInteraction(interactionSpec);

// Execute the interaction object passing an input record and
outputRecord = interaction.execute(inputRecord);
```

 ON DEMAND BUSINESS

Redbooks IBM

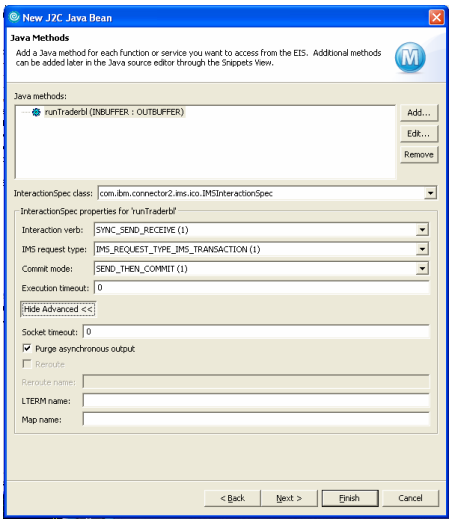
## Selecting the J2C Resource Adapater in RAD




 ON DEMAND BUSINESS

Redbooks IBM

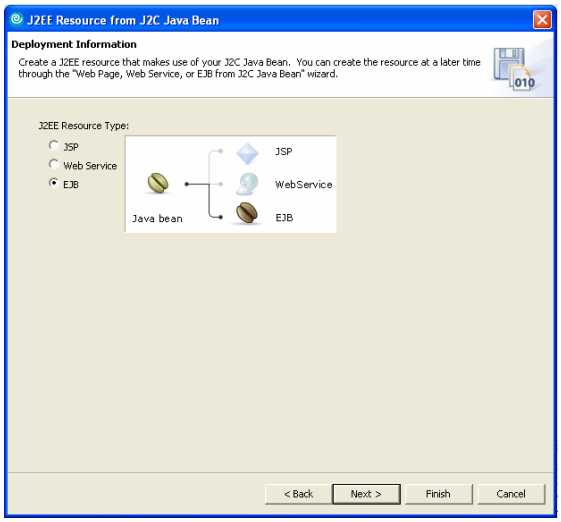
## Adding a Java method to a J2C Bean in RAD




 ON DEMAND BUSINESS

Redbooks IBM

## Creating a J2EE component using J2C Bean




 ON DEMAND BUSINESS

Redbooks IBM

## Agenda

- **What is J2CA and why should I care**
- **Tooling and developing applications for CICS and IMS using J2CA**
- **Using J2CA to access CICS**
- **Using J2CA to access IMS**

 ON DEMAND BUSINESS

Redbooks IBM

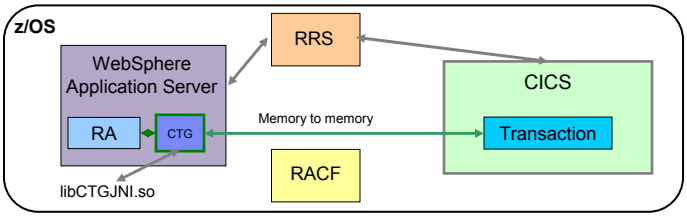
## Using J2CA to access CICS

- **Requires CICS Transaction Gateway**
  - Version 6 is the most current version
- **Can be used in various modes:**
  - *Local*, where WAS and CICS are on the same LPAR
    - This mode only requires the CTG libraries to be present
  - *Remote*, where WAS and CICS can be on the same or different LPARs
    - This mode requires the CTG daemon to be set up
- **2-PC is only possible when using *local* mode**
- **J2CA CTG applications developed with WSAD-IE still use WSIF and fine in WebSphere V6**
- **J2CA CTG applications developed with RAD V6 use new J2C framework**

 ON DEMAND BUSINESS

Redbooks IBM

## Accessing CICS using CTG in local Mode




- **RRS Transaction Coordinator for both WebSphere and CICS (2-PC Support)**
- **WebSphere and CICS must be on the same LPAR**
- **The CICS RRS connection is not involved in the WebSphere Global Transaction**

 ON DEMAND BUSINESS

Redbooks IBM

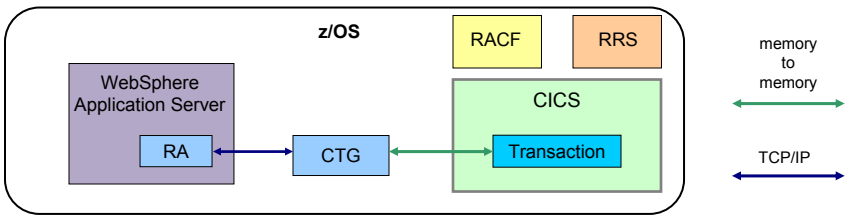
## Accessing CICS using CTG in local Mode - Attributes

- **Performance**
  - Best end-to-end performance
  - Uses less memory than external CICS TG
  - Takes advantage fo EXCI efficiencies
- **Security**
  - RACF can be used for this configuration
  - Thread Identity available for local resources via WebSphere
- **Scalability**
  - Vertical scaling with zSeries resources
- **Availability**
  - Single z/OS image with two points of failure
- **Transactionality**
  - Full two-phase commit


ON DEMAND BUSINESS

Redbooks IBM

## Remote connection to CTG daemon in the same LPAR




- **CICS RA in WebSphere configured in *remote* mode**
- **CTG runs as daemon on the same LPAR as WebSphere and CICS**
- **No 2-PC, as there is no RRS control between WebSphere and CICS**

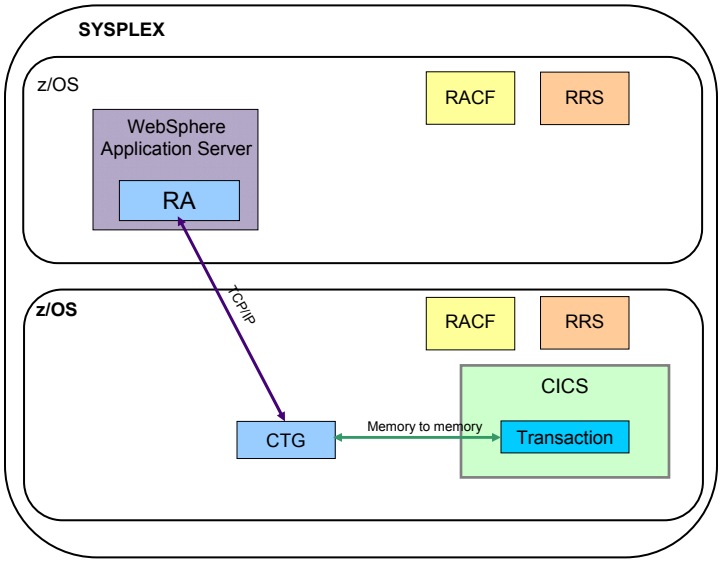

ON DEMAND BUSINESS

## Remote connection to CTG in same LPAR - attributes

- **Performance**
  - Increase path length due to network connection
  - Requests flow over TCP/IP
  - Uses addtional address space
- **Security**
  - JAAS Authentication Alias
- **Scalability**
  - Vertical scaling with zSeries resources
- **Availability**
  - Single z/OS image with three points of failure
- **Transactionality**
  - One-phase commit support



## Remote connection to CTG across LPARs

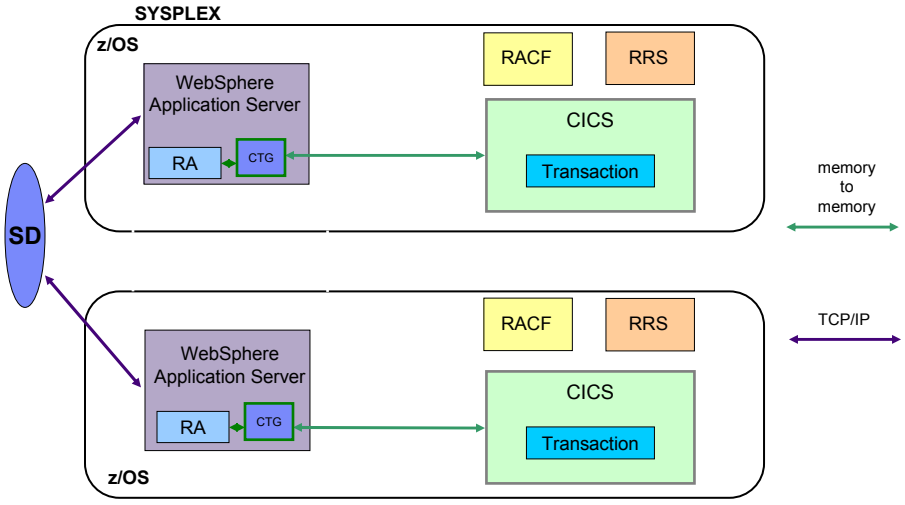


## Remote connection to CTG across LPARs - attributes

- **Performance**
  - Introduces one more address space and a second LPAR
  - Requests flow over TCP/IP across LPARs
  - Uses addtional address space
- **Security**
  - JAAS Authentication Alias
- **Scalability**
  - Horizontal scaling with additional LPARs for WebSphere or CICS
- **Availability**
  - Three points of failure in two z/OS images
  - Provides complete isolation between WebSphere and CICS
- **Transactionality**
  - Single-phase commit support



## Local CICS with increased availability



Redbooks IBM

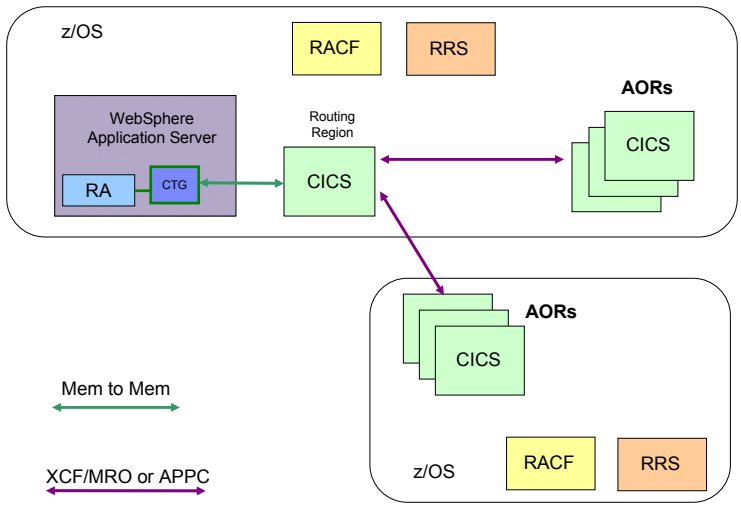
## Local CICS with increased availability - attributes

- **Performance**
  - Increase performance by adding multiple WebSphere+CICS LPARs
  - Sysplex Distributor and WLM will balance workload across LPARs
- **Availability**
  - Fail-over support of cluster IP address provided by SD
  - Non-disruptive quiesce of WebSphere CICS LPAR by SD redirecting IP traffic
  - "Storm-Drain" possibility must be evaluated
- **Security**
  - Easiest method to handle security between WebSphere and CICS – trust z/OS address space
  - Thread identity passed from WebSphere to CICS
- **Transactionality**
  - Complete two-phase commit is supported by RRS because CTG is in the WebSphere address space
- **Scalability**
  - Each WebSphere CICS LPAR can be independently vertically scaled (CPU, …)
  - Horizontal scaling possible by adding more LPARs and enlarging the WebSphere cluster

 ON DEMAND BUSINESS

Redbooks IBM

## Switch local to remote CICS




 ON DEMAND BUSINESS

Redbooks IBM

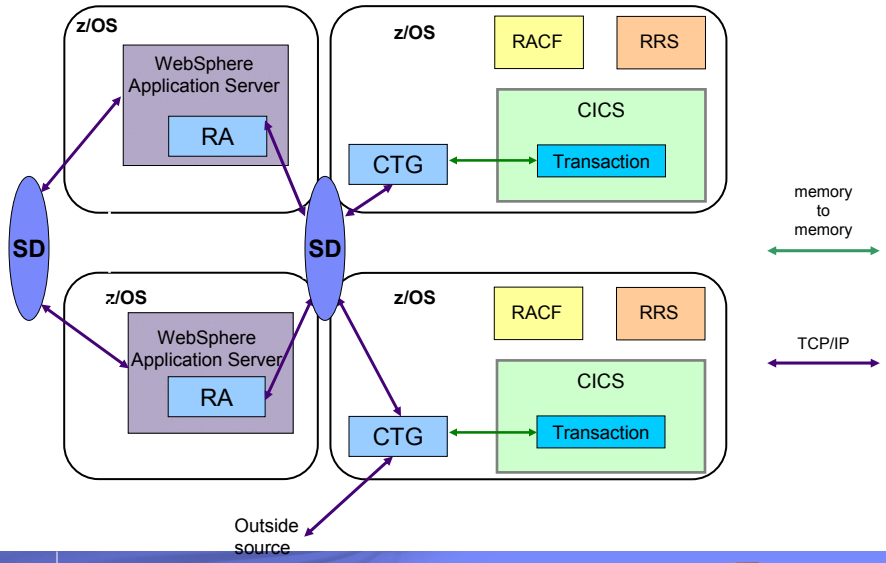
## Switch local to remote CICS - attributes

- **Performance**
  - WebSphere access to local CICS AOR for best performance
  - CICS routing program to multiple AORs with low system overhead
  - CICS traditional performance management
- **Availability**
  - Remote CICS can be accessed if local CICS is unavailable
  - CICSPlex/SM can provide enhanced routing and availability capabilities
  - ARM can restart CICS listener very quickly
- **Security**
  - Thread identity passed from WebSphere container to routing region
  - Once CICS is called, CICS security is used
- **Transactional attributes**
  - If CTG is local to WebSphere then full two-phase commit across all systems
  - If CTG in separate address space from WebSphere then single-phase commit with CICS as Last Participant
- **Scalability**
  - Local WebSphere CICS can be vertically scaled (CPU, memory, channels, …)
  - Remote CICS AOR can be vertically or horizontally scaled (more systems added)

 ON DEMAND BUSINESS

Redbooks IBM

## Remote CICS with availability




 ON DEMAND BUSINESS

Redbooks IBM

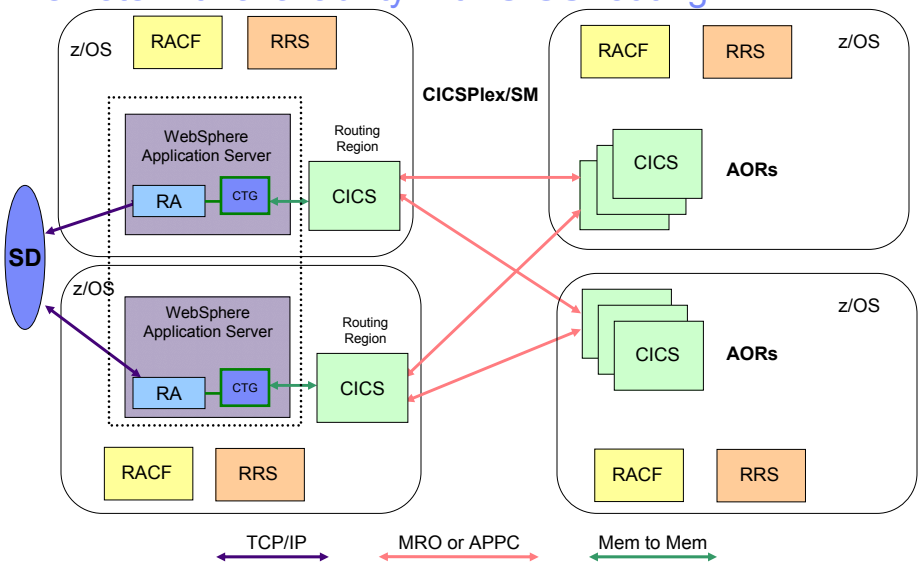
## Remote CICS with availability - attributes

- **Performance**
  - Not as good as local
  - Sysplex Distributor(SD) and WLM will balance workload across WebSphere LPARs
  - Single Cluster IP address
- **Availability**
  - No single point of failure
  - Availability improved with Sysplex Distributor
- **Security**
  - Security the same as remote connection case
- **Transactional capabilities**
  - Single-phase commit with CICS as Last Participant
  - If the application must handle affinity then WebSphere or CICS must handle since SD does not handle affinity
  - If communication goes down during a transaction, recovery may be routed to a different CTG that does not know about transactional state
- **Scalability**
  - Vertical scaling of each LPAR by increasing resources
  - Horizontal scaling by adding LPARs and resources to WebSphere or CICS regions
  - Sysplex Distributor and WLM can workload balance WebSphere LPARs to meet performance goals


ON DEMAND BUSINESS

Redbooks IBM

## Remote with availability with CICS routing





ON DEMAND BUSINESS

Redbooks IBM

## Remote with availability with CICS routing - Attributes

- **Performance**
  - WebSphere can use SD and WLM to meet LPAR performance goals
  - Implements CICS best practice solution for performance management
  - Takes advantage of WebSphere access to local CICS routing region for performance
  - Uses CICS routing program to access multiple AORs with low system overhead
- **Availability**
  - Provides greatest amount of availability due to no single point of failure
  - CICSPlex SM can provide enhanced routing and availability capabilities
- **Security**
  - Same security considerations as discussed in "Routing CICS local to remote"
- **Transactional capabilities**
  - If CICS TG is local to WebSphere then full two-phase commit across all systems
- **Scalability**
  - Local WebSphere or CICS can be vertically scaled (CPU, memory, channels, …)

 ON DEMAND BUSINESS

Redbooks IBM

## Installing the CICSECI J2CA Connector – task list

- **Optionally, define an authentication alias**
- **Install the Resource Adapter**
  - You can load the .rar file from z/OS or a local file
  - cicseci.jar
  - In case of V5 WSIF application, also place cicsecitools.jar file manually in CTG connector directory
- **Configure the connection factories**
  - Optionally, set the Component Managed Authentication alias
  - Choose between local (JNI) and remote (TCP/IP) connection factory, in the custom properties pane of the J2C Connection Factory

 ON DEMAND BUSINESS

Redbooks IBM

## Deploying a CTG J2C Application to WAS V6

- **TranName and TPNName can be set in either the custom properties of the J2C Connection factory or in the J2C Bean itself**
  - * @j2c.interactionSpec-property name="TranName" value="TRDT“

 ON DEMAND BUSINESS

Redbooks IBM

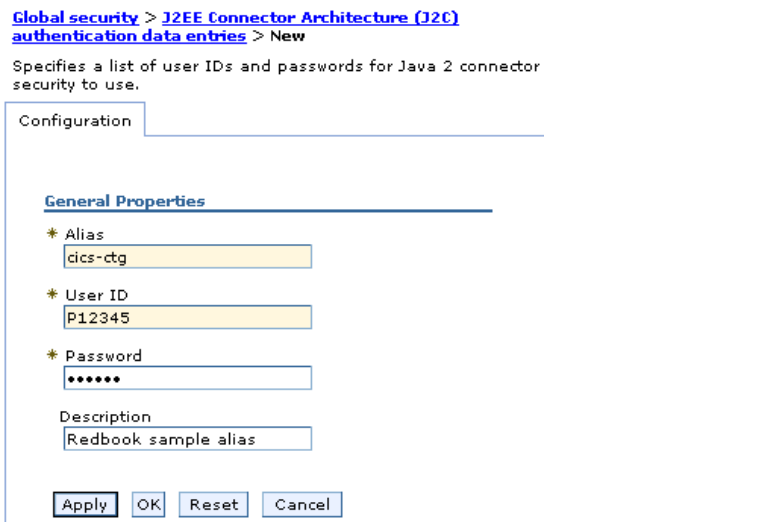
## Authentication Alias

- **A way to store a User ID and password in WebSphere**
  - Password stored encrypted in XML files
- **Authentication (res-auth)**
  - “Container”
    - User ID and password for connection provided by container
  - “Application”
    - User ID and password for connection provided by:
      - explicitly in the application
      - the J2C connection factory in the Component-managed Authentication Alias
- **Associating an Authentication Alias with a J2C Connector for container managed security is deprecated in WebSphere V6.01**
  - An alternative to this is to specify this in the “resources references” of the deployment descriptor of the application (see next page)

 ON DEMAND BUSINESS

Redbooks IBM

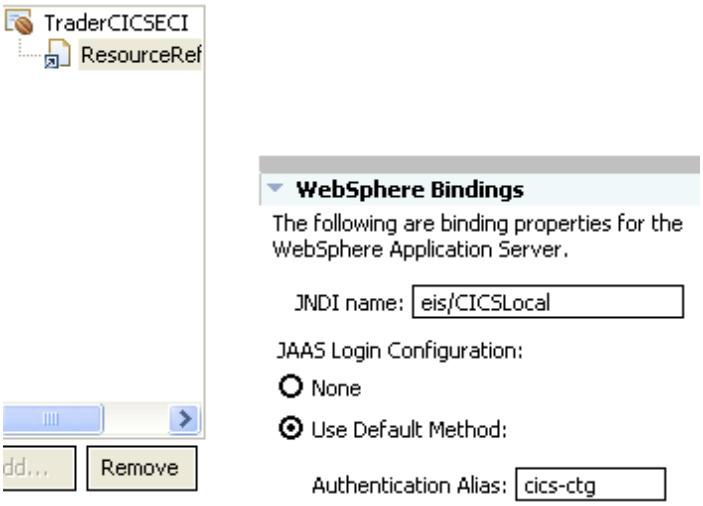
## Authentication alias in J2C Connection Factory Properties




 ON DEMAND BUSINESS

Redbooks IBM

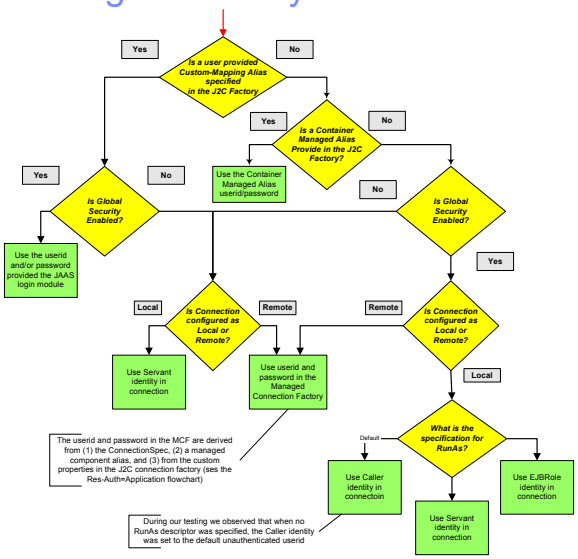
## Specifying JAAS Login in Deployment Descriptor




 ON DEMAND BUSINESS

Redbooks IBM

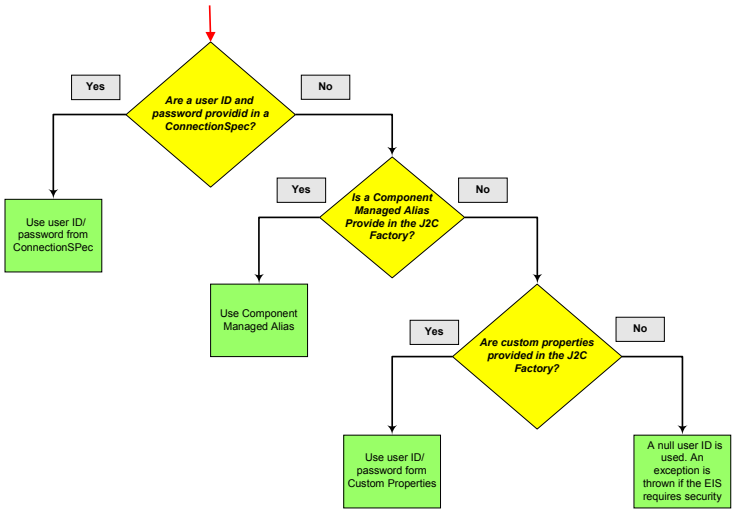
## Container managed security – decision tree




 ON DEMAND BUSINESS

Redbooks IBM

## Application managed security - decision tree




 ON DEMAND BUSINESS

Redbooks IBM

## Application managed security

- **JAAS login configuration parameters are disabled**
- **3 ways to pass the user ID and password to the CICS connection:**
  - Via the application (changing the meta-data)
  - Via authentication alias
  - Via custom properties

 ON DEMAND BUSINESS

Redbooks IBM

## Application managed security – via application (1)

- **Changing the meta-data of the implementation class of the J2C Bean, as follows:**

```
/**
 * @j2c.interactionSpec class="com.ibm.connector2.cics.ECIInteractionSpec"
 * @j2c.interactionSpec-property name="functionName" value="TRADERBL"
 * @j2c.interactionSpec-property name="commareaLength" value="500"
 * @j2c.interactionSpec-property name="replyLength" value="500"
 * @j2c.connectionSpec  class="com.ibm.connector2.cics.ECIConnectionSpec"
 * @j2c.connectionSpec-property  name="userName" argumentBinding="ivUserid"
 * @j2c.connectionSpec-property  name="password" argumentBinding="ivPassword"
 * @generated
 */
```

 ON DEMAND BUSINESS

Redbooks IBM

## Application managed security – via application (2)

- **Adding parameters for the user ID and password to the "run" method of the implementation class of the J2C Bean, as follows:**

```
public itso.cics.j2c.data.TraderCICSECICommarea runTrader
                    (itso.cics.j2c.data.TraderCICSECICommarea arg,
                                        String ivUserid, String ivPassword )
```

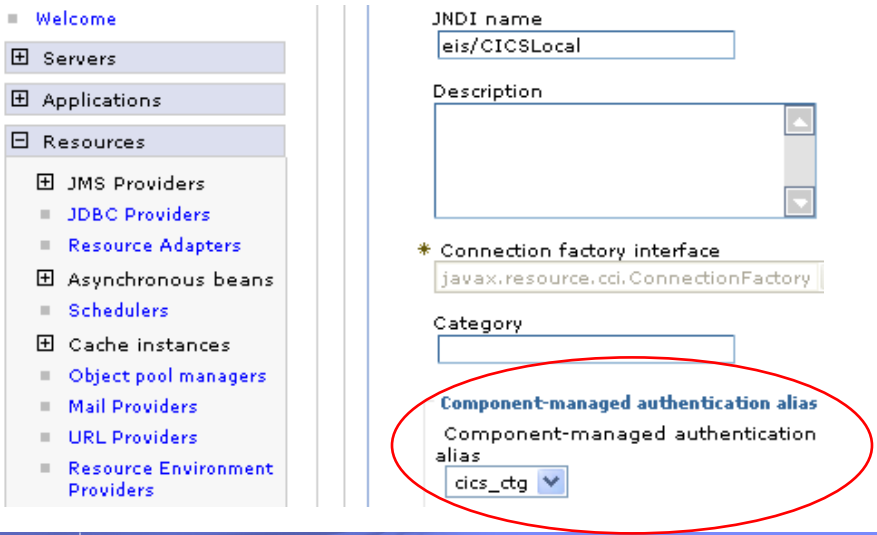
- **Adding parameters for the user ID and password to the interface of the J2C Bean, as follows:**

```
public itso.cics.j2c.data.TraderCICSECICommarea
                runTrader(itso.cics.j2c.data.TraderCICSECICommarea arg,
                                    String ivUserid, String ivPassword)
                throws javax.resource.ResourceException;
```

 ON DEMAND BUSINESS

Redbooks IBM

## Application managed security – via Component-managed authentication alias




 ON DEMAND BUSINESS

## Application managed security – via custom properties of the J2C Connection Factory

Preferences

| Name | Value | Description | Required |
|---|---|---|---|
| TPNName | | | false |
| ClientSecurity | | | false |
| ConnectionURL | local: | | false |
| KeyRingClass | | | false |
| KeyRingPassword | | | false |
| Password | | | false |
| PortNumber | 2006 | PortNumber | false |
| ServerName | SCSCERW1 | | false |
| ServerSecurity | | | false |
| TraceLevel | 1 | TraceLevel | false |
| TranName | | | false |
| UserName | | | false |

Total 12

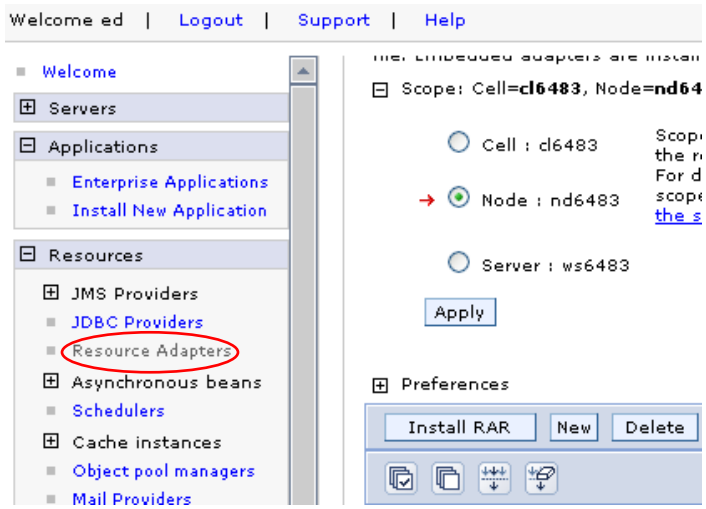


## Configuring an Optional Custom Principle Mapping Module

- **In a Container Managed connection, the behavior which selects the connection identity used to connect to CICS can be customized by providing custom JAAS application login module.**
  - The default principle mapping module does not use the J2EE Identity for a remote connection. A custom login module can change this behavior and use the current J2EE Identity as the connection identity.
  - Or a custom principle mapping module can provide any value for a connection identity and password
  - And requires the enabling of Global Security

Redbooks IBM

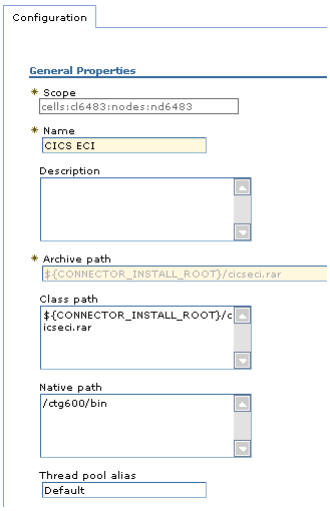
## Installing a Resource Adapter in WAS V6




 ON DEMAND BUSINESS

Redbooks IBM

## CTG J2CA Resource Adapter Configuration




 ON DEMAND BUSINESS

Redbooks IBM

## Custom Properties of J2C Connection Factory for *local* CTG

Preferences

| Name | Value | Description | Required |
|---|---|---|---|
| TPNName | | | false |
| ClientSecurity | | | false |
| ConnectionURL | local: | | false |
| KeyRingClass | | | false |
| KeyRingPassword | | | false |
| Password | | | false |
| PortNumber | 2006 | PortNumber | false |
| ServerName | SCSCERW1 | | false |
| ServerSecurity | | | false |
| TraceLevel | 1 | TraceLevel | false |
| TranName | | | false |
| UserName | | | false |

Total 12

 ON DEMAND BUSINESS

Redbooks IBM

## Custom Properties of J2C Connection Factory for *remote* CTG

Preferences

| Name | Value | Description | Requi |
|---|---|---|---|
| TPNName | | | false |
| ClientSecurity | | | false |
| ConnectionURL | tcp://wtsc48.itso.ibm.com | | false |
| KeyRingClass | | | false |
| KeyRingPassword | | | false |
| Password | | | false |
| PortNumber | 2006 | PortNumber | false |
| ServerName | | | false |
| ServerSecurity | | | false |
| TraceLevel | 1 | TraceLevel | false |
| TranName | | | false |
| UserName | | | false |

Total 12

 ON DEMAND BUSINESS

Redbooks IBM

## Using wsadmin to define a J2C connector

- **Optionally define an authentication alias**
- **Define a CICS Resource Adapter**
- **Define a CICS J2C Connection Factory to connect to a local CICS region**
  - optionally set the Component Managed Authentication alias
- **Run the script**
  - wsadmin.sh -f /tmp/Define-CICS-j2c.jacl
  - wsadmin.sh -user uid -password pwd -f /tmp/Define-CICS-j2c.jacl

 ON DEMAND BUSINESS

Redbooks IBM

## Agenda

- **What is J2CA and why should I care**
- **Tooling and developing applications for CICS and IMS using J2CA**
- **Using J2CA to access CICS**
- **Using J2CA to access IMS**

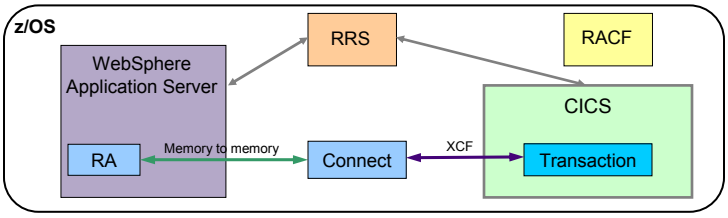
 ON DEMAND BUSINESS

## J2CA and IMS Connect - summary

- Connectivity from WebSphere to IMS using J2CA requires IMS Connector for Java and IMS Connect
- IMS Connector for Java (IC4J) is a Java application which runs within WebSphere on any platform and is installed as a Resource Adapter
- IMS Connect converts a request from IC4J and forwards the request to IMS using IMS's Open Transaction Management Architecture (OTMA) format.
- IMS Connect is a native z/OS application which listens on TCP/IP ports for remote requests over TCP/IP and XCF for local z/OS request.
- IMS Connect can authenticate a user ID/password with the local SAF database
- IMS Connect can authorize a local user's access to IMS Connect using SAF FACILITY resources.



## Accessing IMS using IMS Connect in local Mode



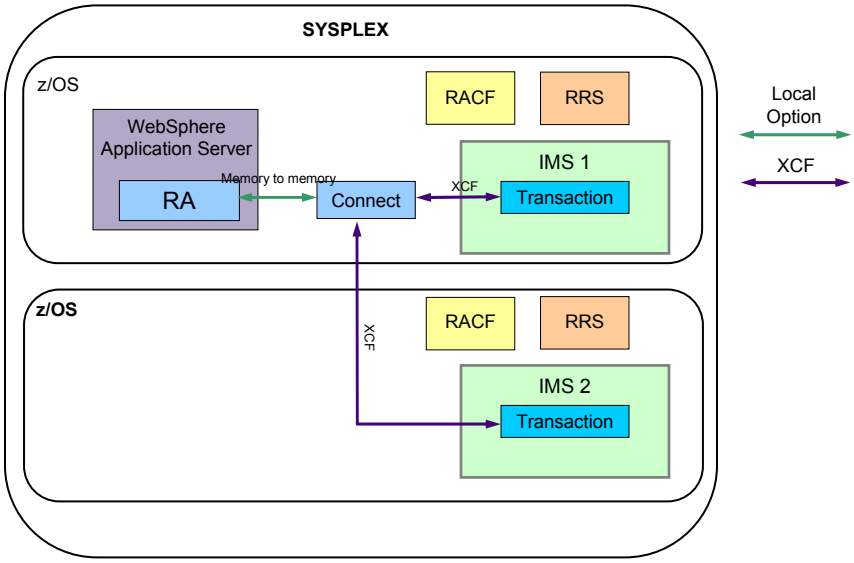

- Provides use of "Local" Option
- All components must run in same z/OS image
- Security with Thread ID
- 2-PC transaction using RRS

## Attributes of local Option

- **Performance**
  - Best option for performance
  - Uses a Program Call protocol
  - RRS performs better in local environment
- **Availability**
  - No failover provided
  - Consider ARM to decrease outage time
- **Security**
  - RACF is used for this configuration
  - Thread Identity available for local resources via Websphere
- **Transactionality**
  - RRS compliant for 2-PC
  - Uses WebSphere global transaction support
  - Conversational and non-conversational supported
- **Scalability**
  - No automatic growth



## Accessing IMS using IMS Connect in *remote* Mode



Redbooks IBM

## Accessing IMS using IMS Connect in *remote* Mode

- **Performance**
  - Single IMS Connect supports a very high transaction rate
  - Cost in using shared data queues
- **Availability**
  - Shared message queues for IMS and multiple IMS Connect address spaces
  - IMS Connect is a single point of failure
- **Security**
  - Thread Identity (RunAs) Identity for local connection
  - RACF setting in IMS Connect used to set authorization
  - User Message Exit can be implemented for security authentication
- **Transactionality**
  - RRS compliant for 2-PC in local LPAR
  - XA Compliant for 2-PC in remote LPAR
  - Uses WebSphere global transaction support
- **Scalability**
  - Shared queues can be used with multiple IMS LPARs

 ON DEMAND BUSINESS

Redbooks IBM

## IMS Connect Configuration

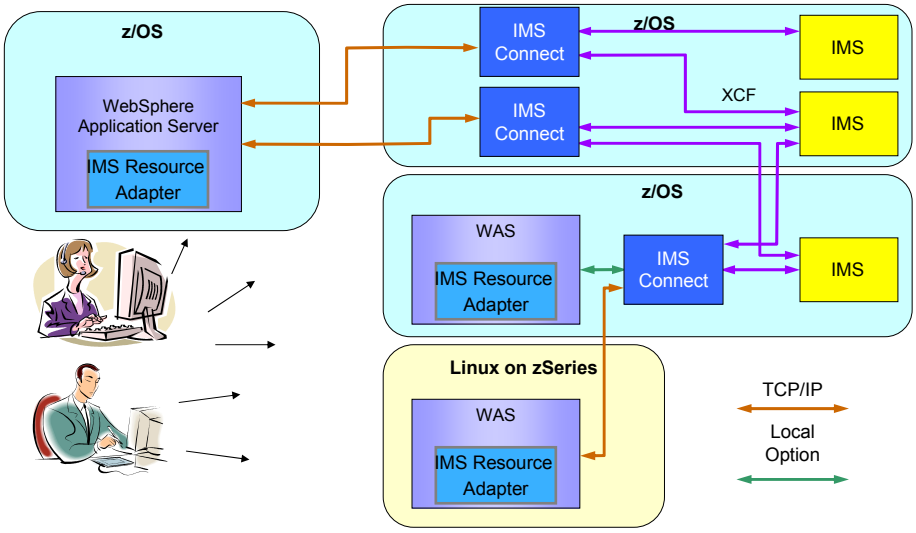
```
HWS (ID=IMSHWS,RACF=Y,RRS=Y)
TCPIPHOSTNAME=TCPIP,RACFID=RACFID,PORTID=(4000,LOCAL),
   SSLPORT=(4001),SSLENVAR=HWSSSL,EXIT=(HWSIMSO0))
DATASTORE (ID=HWS8,GROUP=OTMAGRP,MEMBER=HWSMEM,TMEMBER=OTMAMEM)
```

| | |
|---|---|
| **RACF** | Specifies whether or not the password and user ID provided by either the client application or a user exit routine are passed to RACF for authentication. |
| **RACFID** | A default RACF ID for exits to pass to OTMA for security checking if the RACF ID has not explicitly been set in the incoming message or by the user exit. |
| **SSLENVAR** | The member name of the SSL initialization file. |
| **SSLPORT** | Defines the Secure Socket Layer (SSL) ports. |

 ON DEMAND BUSINESS

Redbooks IBM

## WebSphere, IMS Connect and IMS




 ON DEMAND BUSINESS

Redbooks IBM

## Agenda

- **What is J2CA and why should I care**
- **Tooling and developing applications for CICS and IMS using J2CA**
- **Using J2CA to access CICS**
- **Using J2CA to access IMS**
- **Problem determination**

 ON DEMAND BUSINESS

Redbooks IBM

## WebSphere Problem Determination

- **Inspect SYSOUT for relevant error messages**
- **Trace Settings**
  - WebSphere
    - com.ibm.ejs.j2c.*=all=enabled
    - com.ibm.connector2.*=all=enabled
    - WAS.j2c=all=enabled
  - J2C Connection Factories
    - set trace level 3 in the connection factory
- **Controlling**
  - ras_trace_outputLocation=SYSPRINT BUFFER
  - Modify Commands:
    - F server,TRACETOSYSPRINT=YES | NO

 ON DEMAND BUSINESS

Redbooks IBM

## CICS J2C Problem Determination

- **Tracing Settings**
  - WebSphere
    - com.ibm.ejs.j2c.*=all=enabled
    - com.ibm.connector2.cics.*=all=enabled
    - WAS.j2c=all=enabled
  - J2C Connection Factories
    - set trace level 3 in the connection factory
  - CTG
    - CTG_JNI_TRACE set to a file
  - CICS
    - Use CEDF on the IRC Connection

 ON DEMAND BUSINESS

Redbooks IBM

## IMS J2C Problem Determination

- **Tracing Settings:**
  - WebSphere
    - com.ibm.ejs.j2c.*=all=enabled
    - com.ibm.connector2.ims.*=all=enabled
    - WAS.j2c=all=enabled
  - J2C Connection Factories
    - set trace level 3 in the connection factory
  - IMS Connect
    - Commands (operator replies)
      - Start - RECORDER ON
      - Stop - RECORDER OFF
    - Results written to dataset identified by DDname HWSRCORD

 ON DEMAND BUSINESS

Redbooks IBM

## Module 6

## J2EE connectivity using RMI over IIOP

 ON DEMAND BUSINESS

Redbooks IBM

## RMI-IIOP - Overview

- **J2EE's synchronous remote communication protocol**
- **Based on CORBA'S Internet Inter-Orb Protocol and Java's Remote Method Invocation (RMI)**
- **Both client and server must be fully J2EE supportive. Examples of scenarios are:**
  - A Java Windows client calling an EJB in a WebSphere server
  - A Java servlet in a WebSphere server calling an EJB in another WebSphere server
  - An EJB in a WebSphere server calling another EJB in another WbSphere server
  - A servlet or EJB in a WebSphere server calling an EJB in CICS
- **Is transactional (2-PC) and supports security**
- **RMI-IIOP plumbing is all generatered and practically invisible to the developer**

 ON DEMAND BUSINESS

Redbooks IBM

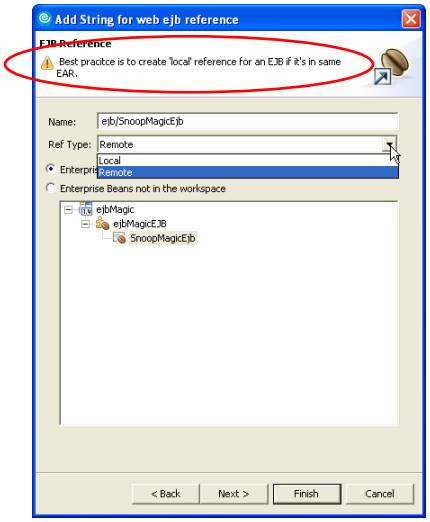
## Local scenario - Lookup of an EJB

- **Before an EJB can be called through RMI-IIOP, a lookup needs to be performed**
- **In the most simple case…**
  - …the servlet and EJB are in same EAR file and
  - …the entire application is deployed to one server
- **Web.xml in war file containing the servlet**
  - Has logical resource reference to the EJB
    - EJB has JNDI name associated with it
      - e.g. ejb/EJBSample
  - Logical EJB reference in web.xml
    - is mapped to JNDI name associated with EJB

 ON DEMAND BUSINESS

Redbooks IBM

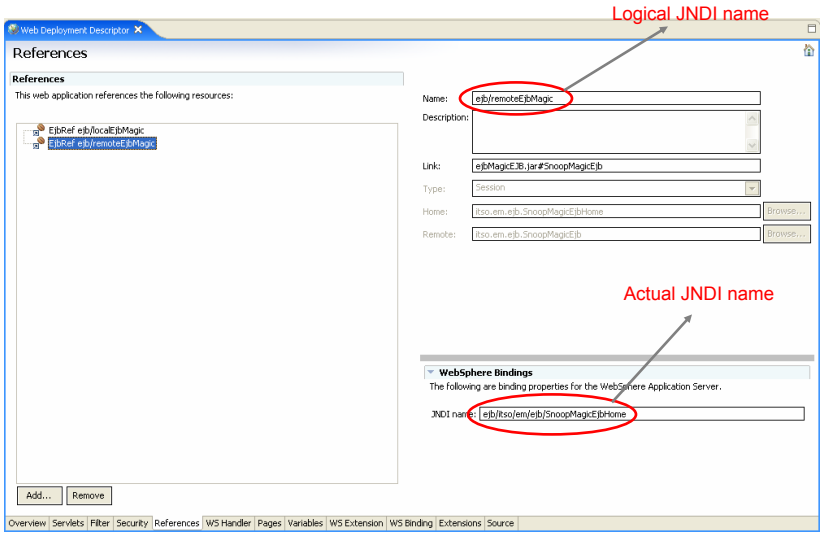
## Local vs. Remote References to EJBs




 ON DEMAND BUSINESS

Redbooks IBM

## References to EJBs from Web Deployment Descriptor




 ON DEMAND BUSINESS

Redbooks IBM

## Logical Lookup of an EJB

```
SnoopMagicEjbHome seHome = null;
SnoopMagicEjb se = null;
Object oHome = null;

Context ic = new InitialContext();
oHome = ic.lookup('java:comp/env/ejb/remoteEjbMagic');
seHome = ((SnoopMagicEjbHome)
         (javax.rmi.PortableRemoteObject.narrow(
         oHome,SnoopMagicEjbHome.class)));
se = seHome.create();
```

Refers to logical EJB reference set In deployment descriptor

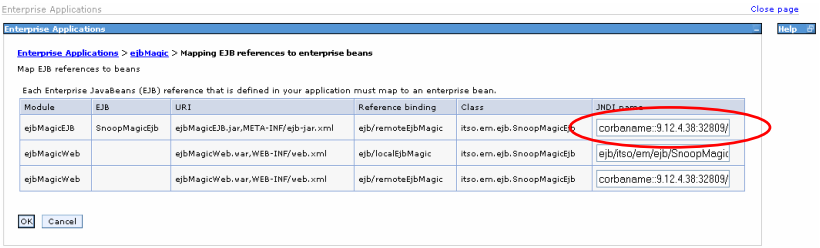
- **At deployment time the binding between the logical reference and actual JDNI name is set**

 ON DEMAND BUSINESS

Redbooks IBM

## Remote lookup of an EJB

- **When doing a lookup in a remote server, you enter a value such as**
  - `corbaname::9.12.4.38:32809/NameServiceServerRoot#cell/nodes/nd6483/servers/ws6483/ejb/itso/em/ejb/SnoopMagicEjbHome`
- **The format of this value is dictated by the way WebSphere stores its JNDI names**
- **You can set this value in the WebSphere admin console in Applications → Enterprise Applications → <your application> → Map EJB references to beans**




 ON DEMAND BUSINESS

Redbooks IBM

## Removing physical dependencies

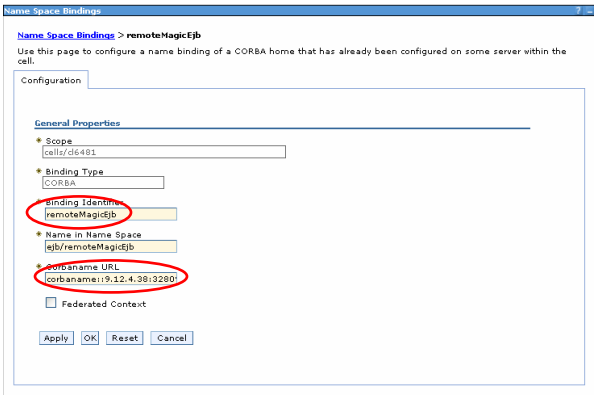
- **If you need to use qualified names…**
  - e.g.
    - `corbaname::wtsc59.itso.ibm.com:2804/NameServiceServerRoot#cell/nodes/nd4cd4sc59/servers/wd4nd4cd4sc59/ejb/EJBSample`
  - you will have node and server names in JNDI names
- **If things change…**
  - e.g. EJB is put into a different node
    - then need to update JNDI setting in the 'client' server
    - which is not ideal
  - You can handle this by setting up an EJB Name Space Binding on the Remote Server



Redbooks IBM

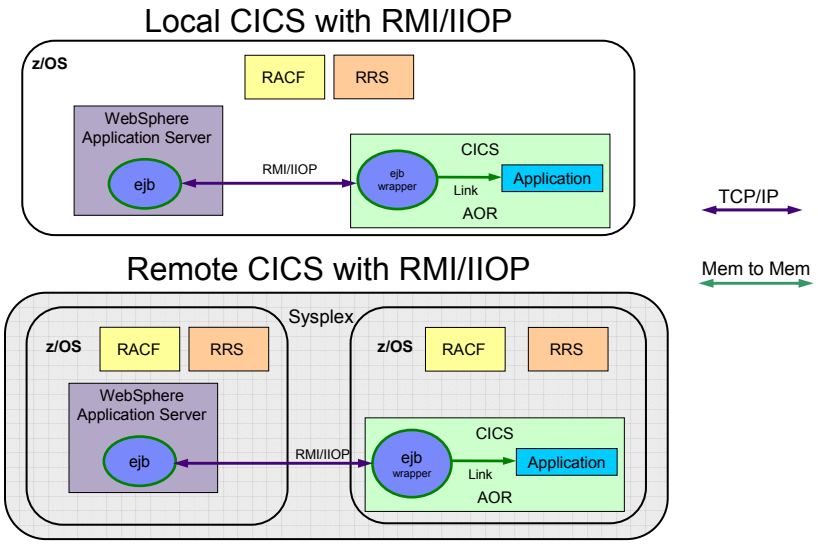
## Name Space Binding

- **Specify in Environment → Naming → Name Space Bindings**
- **Reference from application remains the same, i.e. will use a logical reference**



Redbooks IBM

## An RMI-IIOP Scenario with WebSphere and CICS




 ON DEMAND BUSINESS

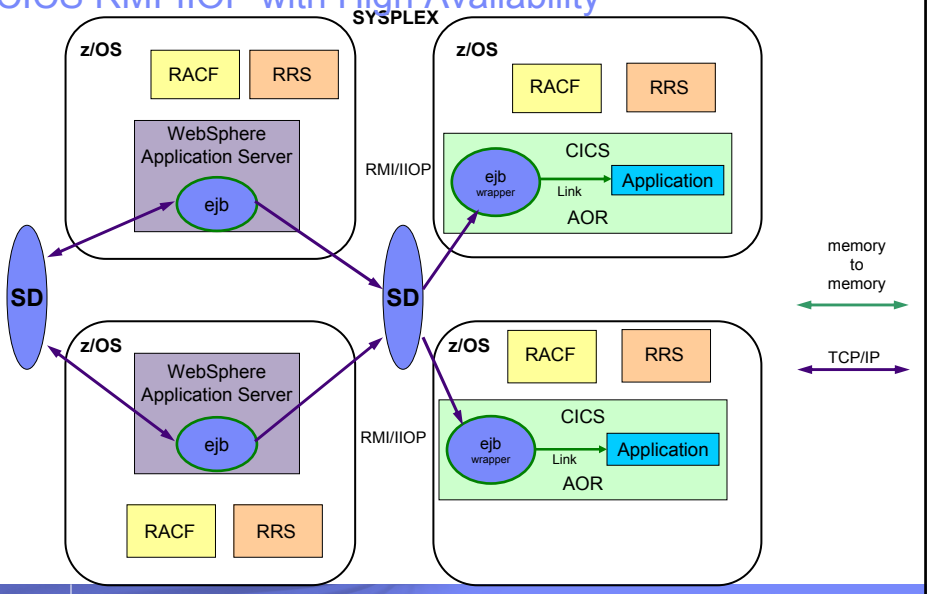
Redbooks IBM

## CICS access through RMI/IIOP - attributes

- **Performance**
  - Overhead of CICS EJB wrapper
  - Overhead of RMI over IIOP over TCP/IP
- **Availability**
  - Same considerations as local connection attributes
  - EJB component availability has to be considered against CTG availability
- **Security**
  - Consideration are similar to remote mode J2CA connection
- **Transactional attributes**
  - Full J2EE transactions between EJBs in WebSphere and CICS
  - Full two-phase commit between WebSphere and CICS
- **Scalability**
  - If on one LPAR then LPAR can be vertically scaled
  - If multiple LPARs then any component of the system can be horizontally scaled

 ON DEMAND BUSINESS

## CICS RMI-IIOP with High Availability






## CICS RMI/IIOP with availability - attributes

- **Performance**
  - Increased performance by adding multiple WebSphere and CICS LPARs
  - Sysplex Distributor and WLM will balance workload across WebSphere LPARs
  - Multiple AORs with IIOP listener
- **Availability**
  - No single point of failure
  - Availability improved with Sysplex Distributor
- **Security**
  - Security the same as remote connection case
- **Transactional attributes**
  - Full EJB transactions with two-phase commit
- **Scalability**
  - Vertical scaling of each LPAR by increasing resources
  - Horizontal scaling by adding LPARs and resources to WebSphere or CICS regions
  - Sysplex Distributor and WLM can balance WebSphere LPARs to meet performance goals


Redbooks
IBM
Thank you
Obrigado
너를 감사하십시요
_ 感谢你
Dankie
Bedankt
Danke
Merci
Hvala
© 2005 IBM Corporation
ON DEMAND BUSINESS