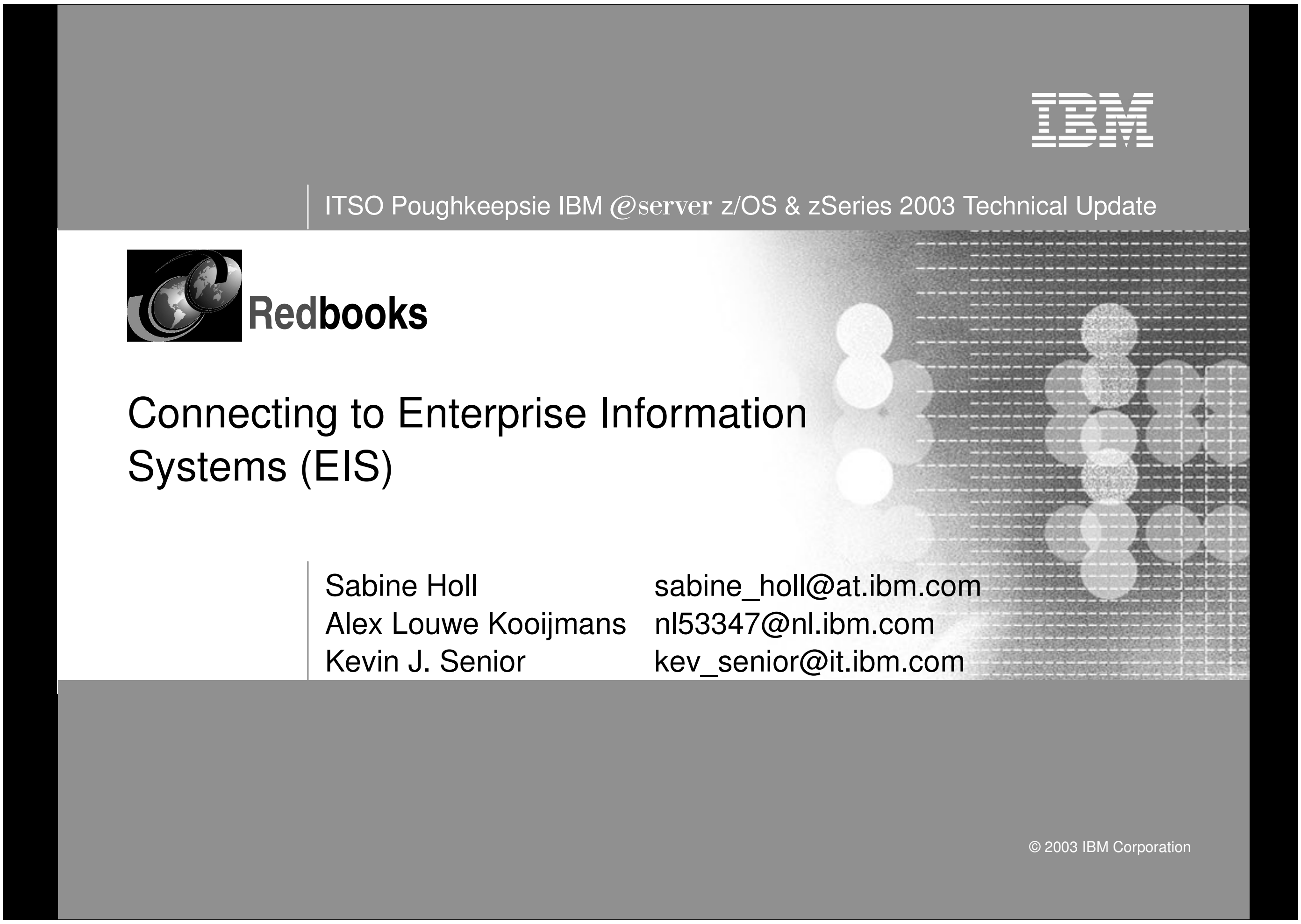
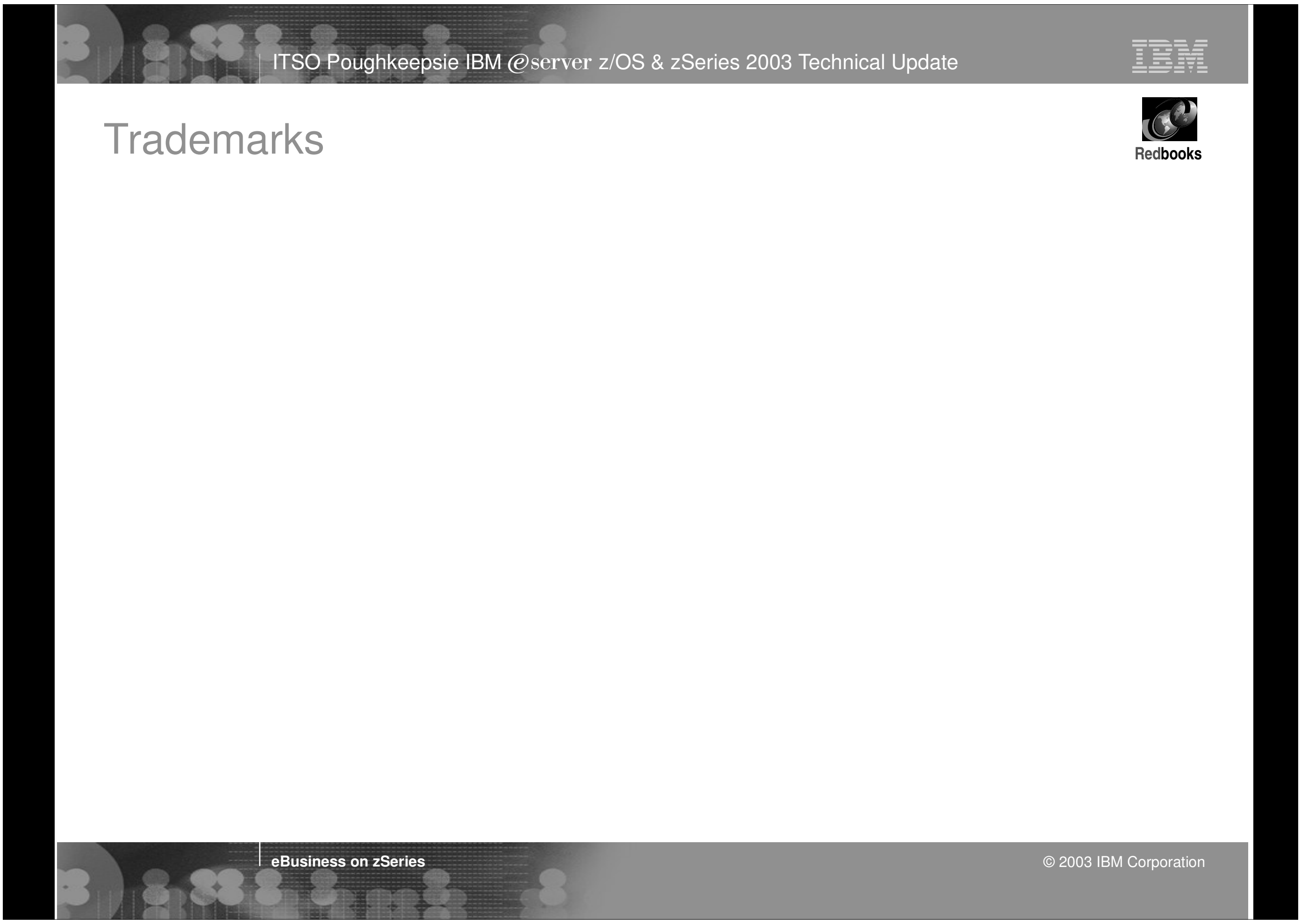


ITSO Poughkeepsie IBM @server z/OS & zSeries 2003 Technical Update

Redbooks

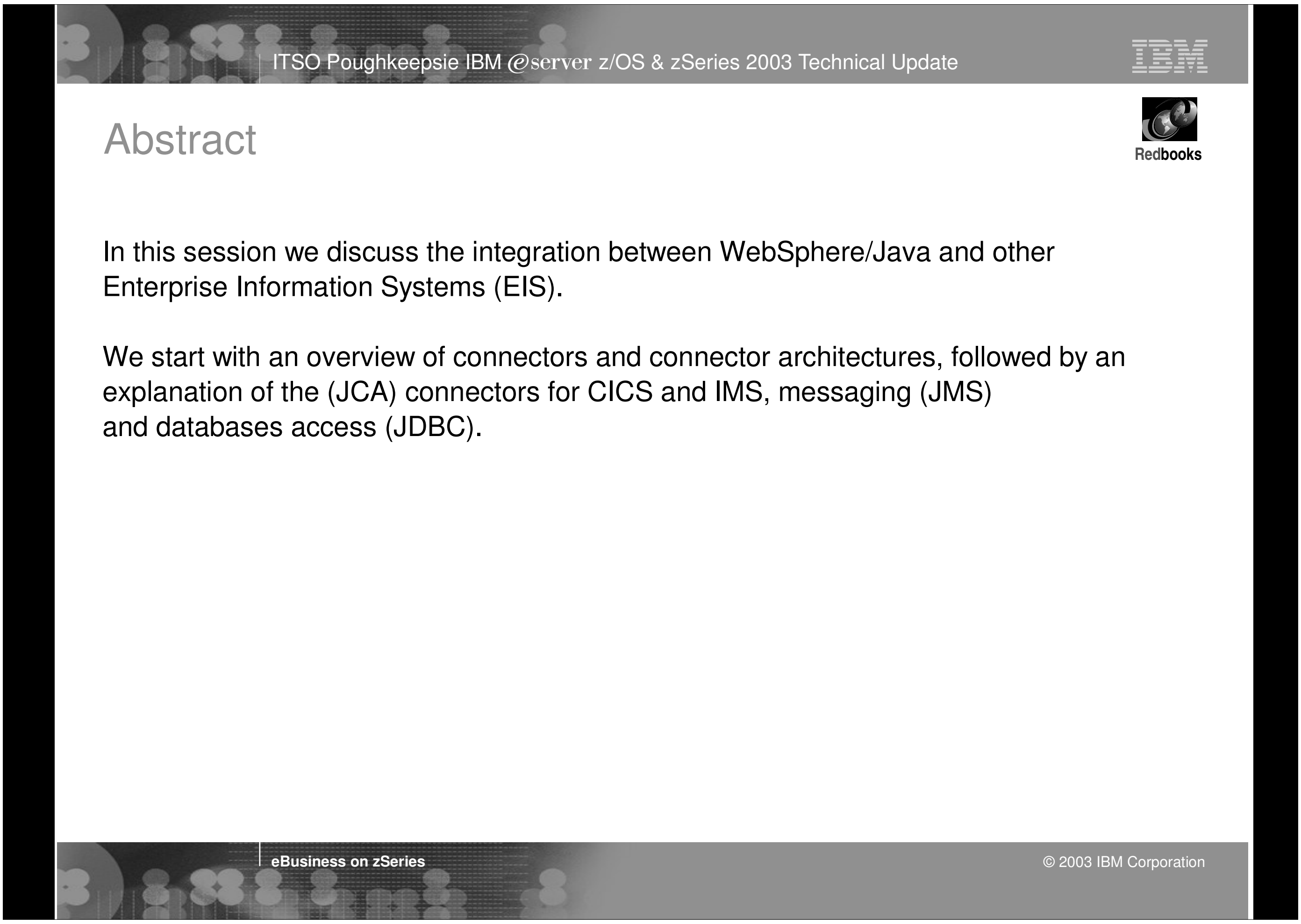
Connecting to Enterprise Information Systems (EIS)

Sabine Holl sabine_holl@at.ibm.com
Alex Louwe Kooijmans nl53347@nl.ibm.com
Kevin J. Senior kev_senior@it.ibm.com



Trademarks

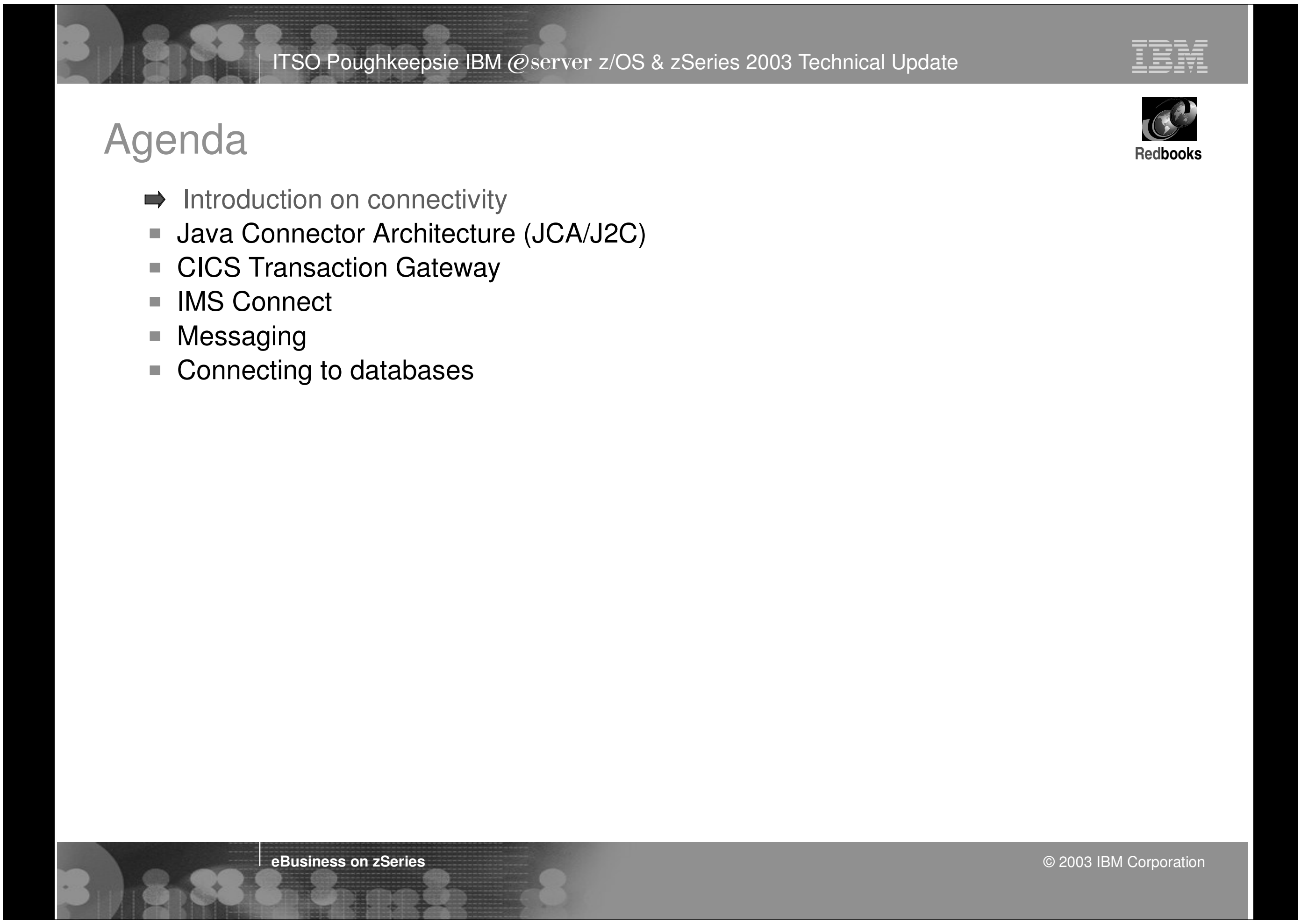
Redbooks



Abstract

In this session we discuss the integration between WebSphere/Java and other Enterprise Information Systems (EIS).

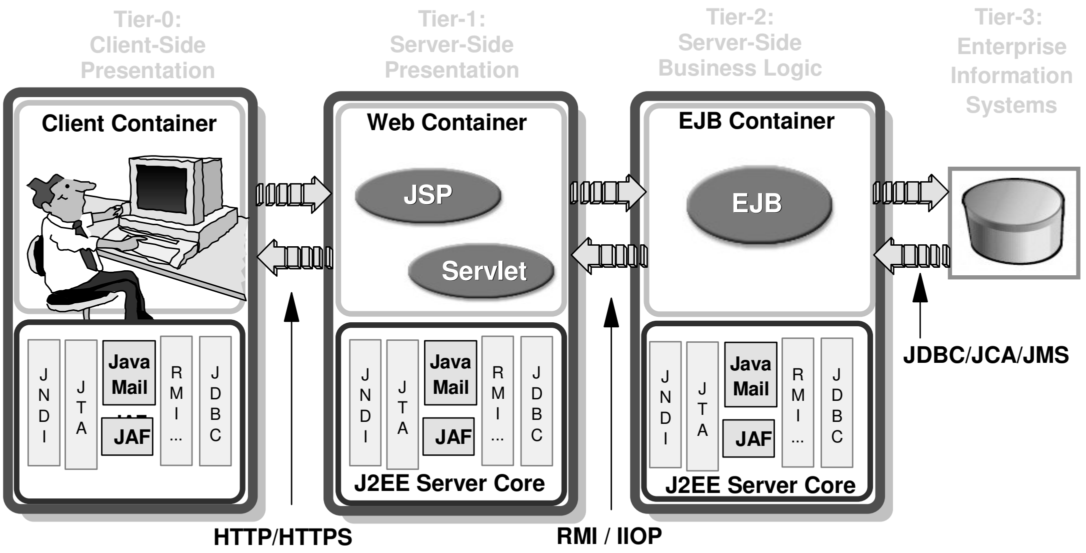
We start with an overview of connectors and connector architectures, followed by an explanation of the (JCA) connectors for CICS and IMS, messaging (JMS) and databases access (JDBC).



Agenda

- ➡ Introduction on connectivity
- Java Connector Architecture (JCA/J2C)
- CICS Transaction Gateway
- IMS Connect
- Messaging
- Connecting to databases

J2EE - The big picture



Tier-0:
Client-Side
Presentation

Tier-1:
Server-Side
Presentation

Tier-2:
Server-Side
Business Logic

Tier-3:
Enterprise
Information
Systems

Client Container

Web Container

EJB Container

JSP

Servlet

EJB

JNDI JTA Java Mail JAF RMI ... JDBC

J2EE Server Core

HTTP/HTTPS

RMI / IIOP

JDBC/JCA/JMS

The reason for connectors

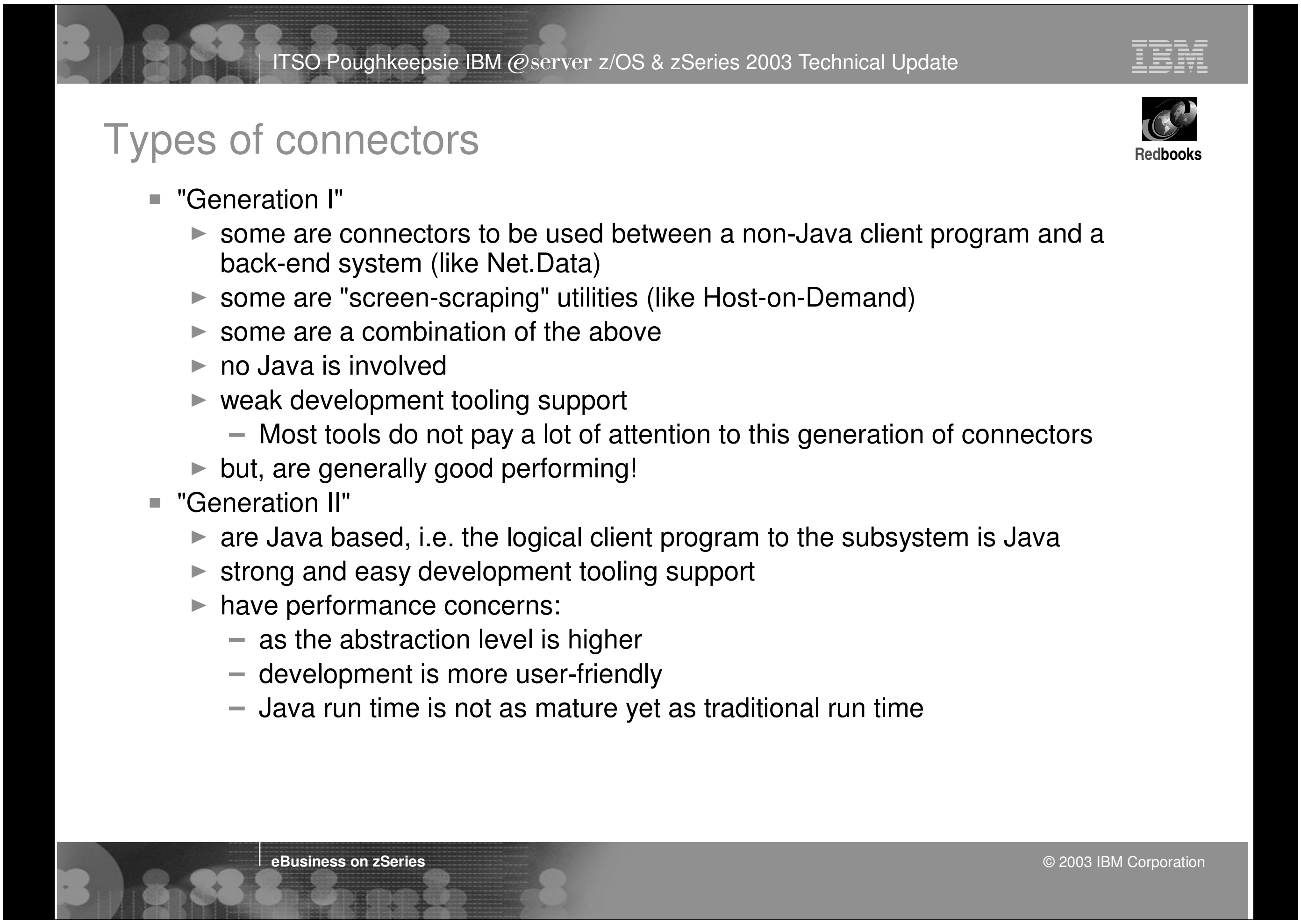
- Simplify integration of diverse EIS
  - standardized interface for client and resource portable across all compliant J2EE servers
  - facilitate scalable architectures
- Through the resource adapters connectors provide *Quality of Service* features transparently to the client application
- Flexibility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

Quality of Service (QoS)

- 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

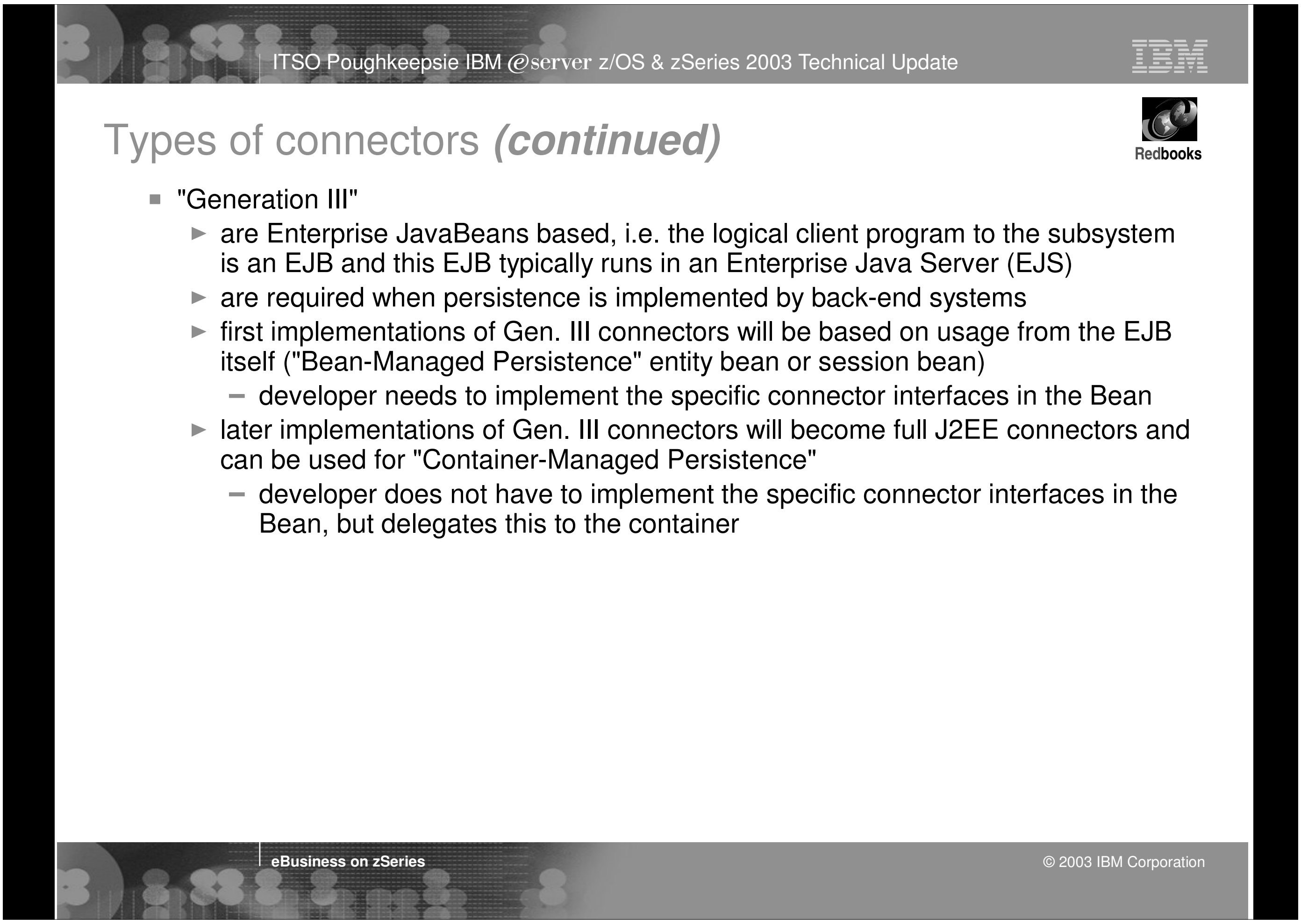
Connectors, what are they?

- Objectives:
  - make the communication protocol / mechanism between a client application and a subsystem transparent to the application developer
  - provide portability of the client application
- Most connectors can be used:
  - locally -> client and target subsystem reside on the same system
  - remote -> client resides on another computer than the target subsystem
- 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 introducing the ~~IBM Common Connector Framework (CCF)~~ and Java Connector Architecture (JCA)



Types of connectors

- "Generation I"
  - some are connectors to be used between a non-Java client program and a back-end system (like Net.Data)
  - some are "screen-scraping" utilities (like Host-on-Demand)
  - some are a combination of the above
  - no Java is involved
  - weak development tooling support
    - Most tools do not pay a lot of attention to this generation of connectors
  - but, are generally good performing!
- "Generation II"
  - are Java based, i.e. the logical client program to the subsystem is Java
  - strong and easy development tooling support
  - have performance concerns:
    - as the abstraction level is higher
    - development is more user-friendly
    - Java run time is not as mature yet as traditional run time



Types of connectors *(continued)*

- "Generation III"
  - are Enterprise JavaBeans based, i.e. the logical client program to the subsystem is an EJB and this EJB typically runs in an Enterprise Java Server (EJS)
  - are required when persistence is implemented by back-end systems
  - first implementations of Gen. III connectors will be based on usage from the EJB itself ("Bean-Managed Persistence" entity bean or session bean)
    - developer needs to implement the specific connector interfaces in the Bean
  - later implementations of Gen. III connectors will become full J2EE connectors and can be used for "Container-Managed Persistence"
    - developer does not have to implement the specific connector interfaces in the Bean, but delegates this to the container

J2EE and connectivity

Redbooks

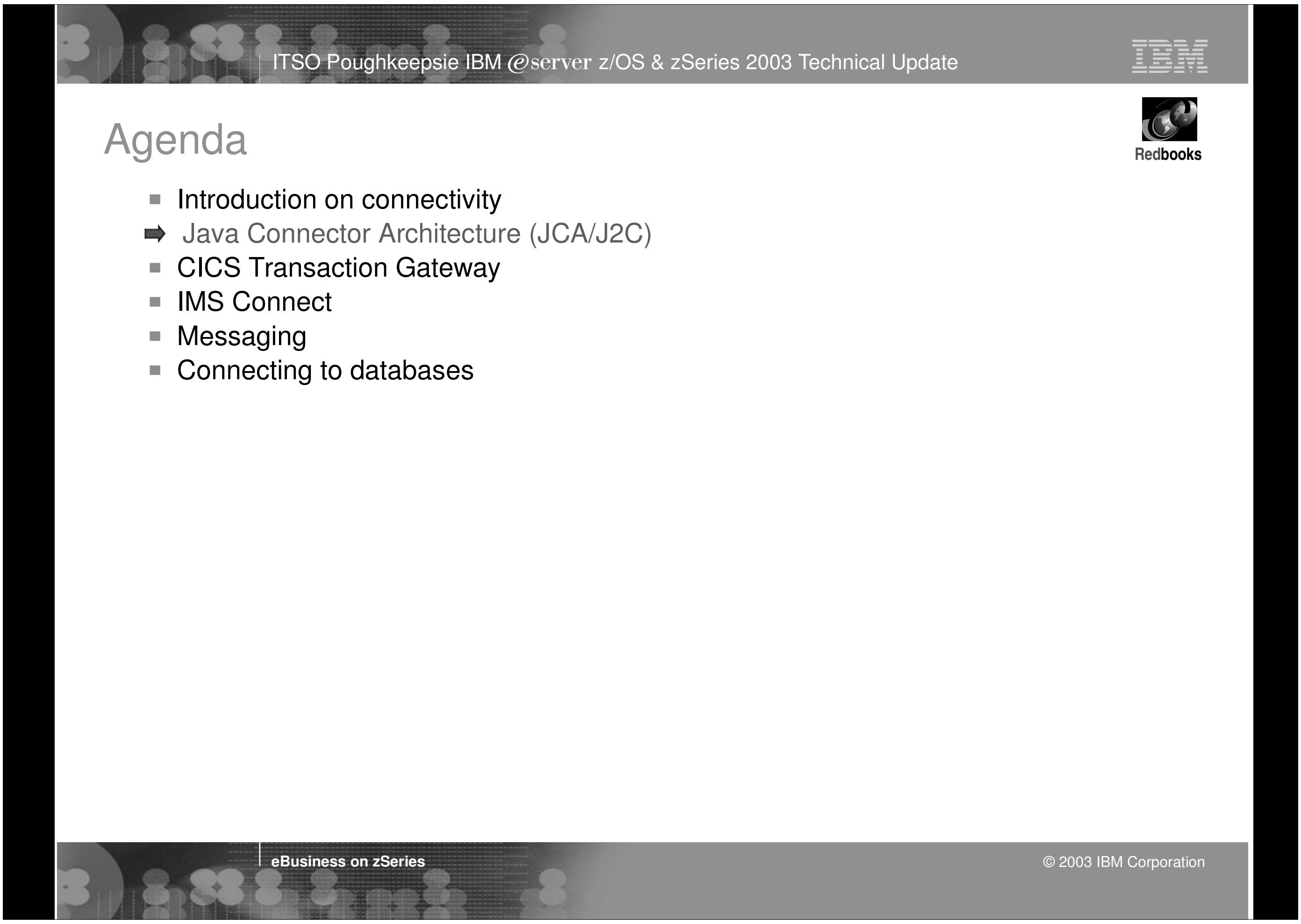
- Java 2 Platform Enterprise Edition (J2EE) defines different ways of accessing back ends:
  - Java DataBase Connectivity (JDBC)
    - for relational databases only
    - well, not entirely true, also IMS DB can be accessed through JDBC APIs
  - Java Message Service (JMS)
    - for message oriented middleware
  - Java Connector Architecture (JCA)
    - for Enterprise Information Systems
    - transaction processors, ERP systems, legacy DB

Supported integration on z/OS

Redbooks

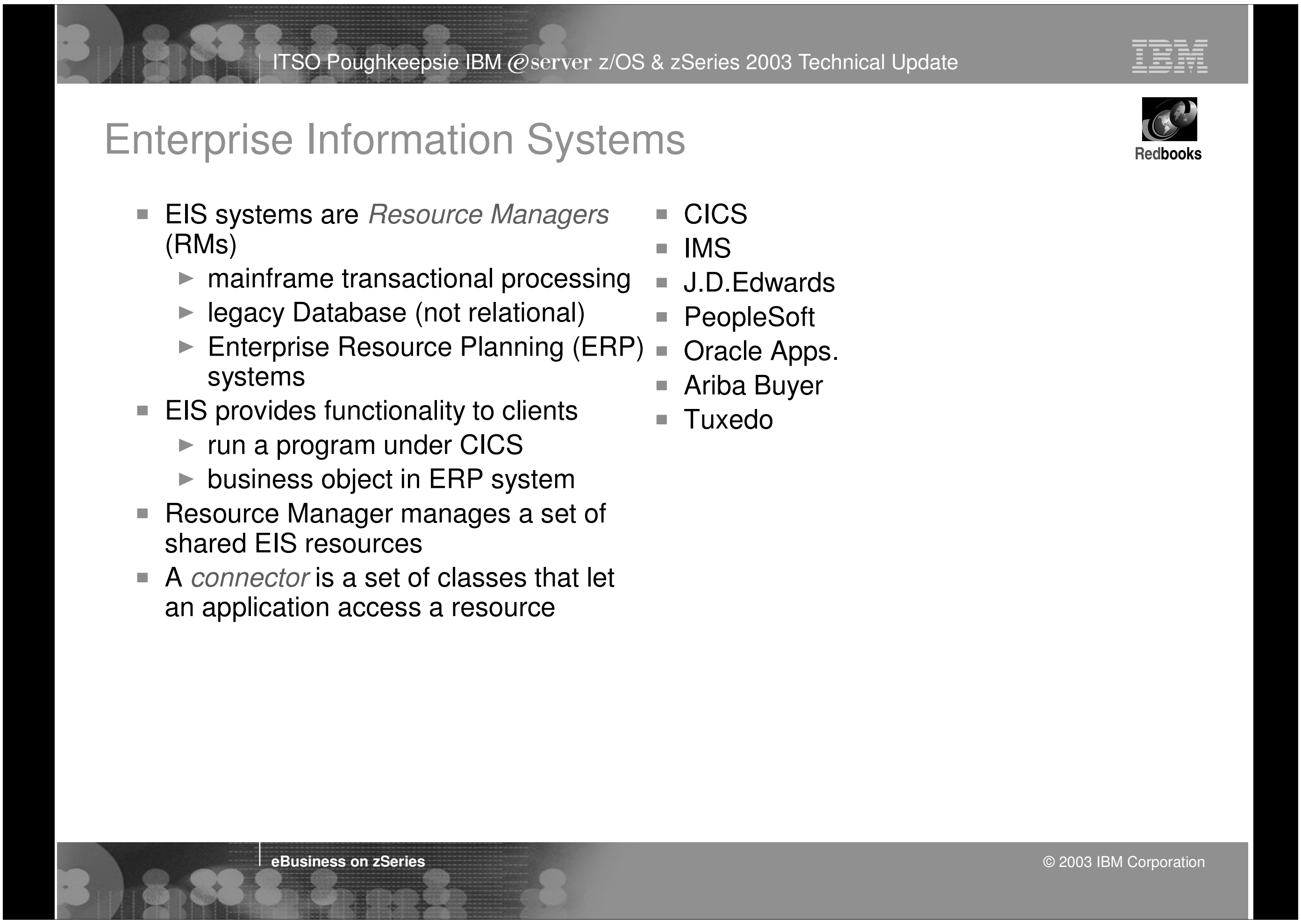
	Function	Available/planned	Direct access from:
DB2 on z/OS	▪ JDBC (Type 2 driver) ▪ JDBC (Type 4 driver) ▪ SQLJ	▪ Available from IBM ▪ Available from IBM and vendors (HiT, HOB, Merant) ▪ Available on z/OS	▪ Local Java clients ▪ Remote and local Java clients ▪ Local Java clients and remote Java clients through DB2 Connect
Oracle on z/OS	▪ JDBC (Type 4 driver)	▪ Available from Oracle	▪ Local and remote Java clients
CICS	▪ CICS Transaction Gateway V5 ▪ IIOP to access CICS EJBs ▪ SOAP for CICS	▪ Available ▪ Available in TS 2.2 ▪ Downloadable from Web	▪ Local and remote Java clients ▪ Java clients ▪ z/OS EJBs (session)
IMS	▪ IMS Connect V2.1	▪ Available	▪ Local and remote Java clients ▪ z/OS EJBs (session)
MQSeries	▪ MQSeries classes for Java ▪ JMS classes for MQ ▪ MQSeries Bindings for OS/390	▪ Available in MQ product ▪ JMS provider in WAS V5 ▪ Available as support pac	▪ Remote Java/MQ clients ▪ Local and remote Java/MQ clients ▪ Only local Java clients
VSAM files	▪ Java Record I/O	▪ Available (incl. in JDK)	▪ Only from Java on z/OS
SAF (RACF etc.)	▪ SAF classes	▪ Available (incl. in JDK)	▪ Only from Java on z/OS
C/C++ modules	▪ JNI	▪ Available (incl. in JDK)	▪ Only from Java on z/OS
ASM, COBOL, PL/I	▪ JNI	▪ Available (incl. in JDK)	▪ Only from Java on z/OS

* Status as per 10/2003



Agenda

- Introduction on connectivity
- ➡ Java Connector Architecture (JCA/J2C)
- CICS Transaction Gateway
- IMS Connect
- Messaging
- Connecting to databases



Enterprise Information Systems

- EIS systems are *Resource Managers* (RMs)
  - mainframe transactional processing
  - legacy Database (not relational)
  - Enterprise Resource Planning (ERP) systems
- EIS provides functionality to clients
  - run a program under CICS
  - business object in ERP system
- Resource Manager manages a set of shared EIS resources
- A *connector* is a set of classes that let an application access a resource

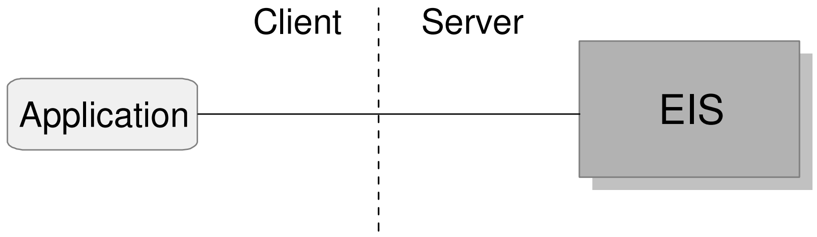
- CICS
- IMS
- J.D.Edwards
- PeopleSoft
- Oracle Apps.
- Ariba Buyer
- Tuxedo

J2EE Resource Manager usage

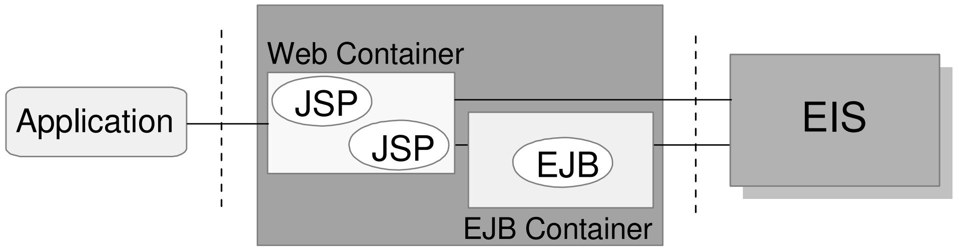
- J2EE prescribes a common model for deployment and use of resource managers by components
  - configure and deploy Connection Factory in naming directory (via tooling)
  - look up a Connection Factory in naming directory through JNDI
  - use the Connection Factory to acquire a Connection
  - use the Connection in a Resource Manager specific manner
  - close the Connection

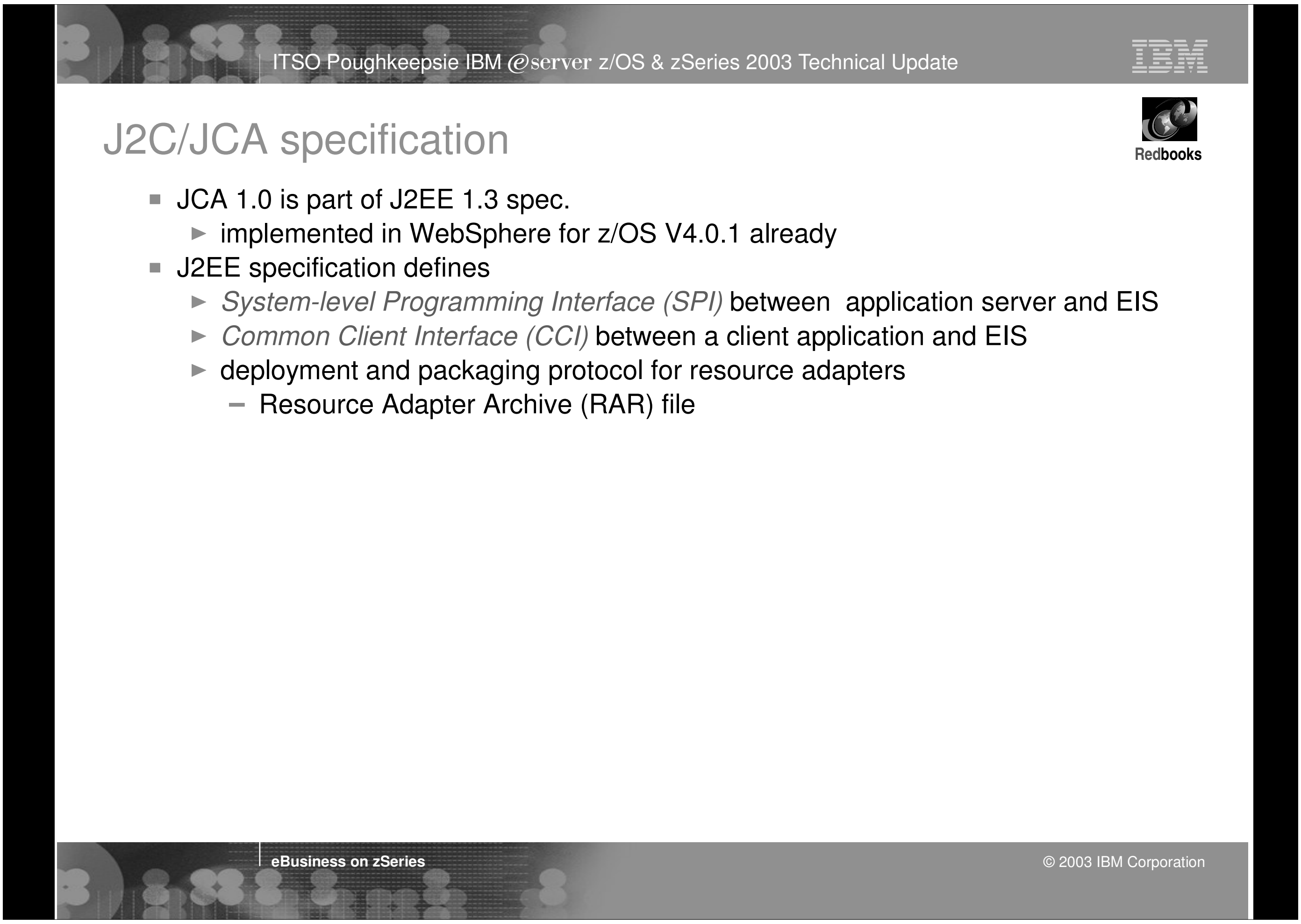
Managed and non-managed environments

Non-managed Environment



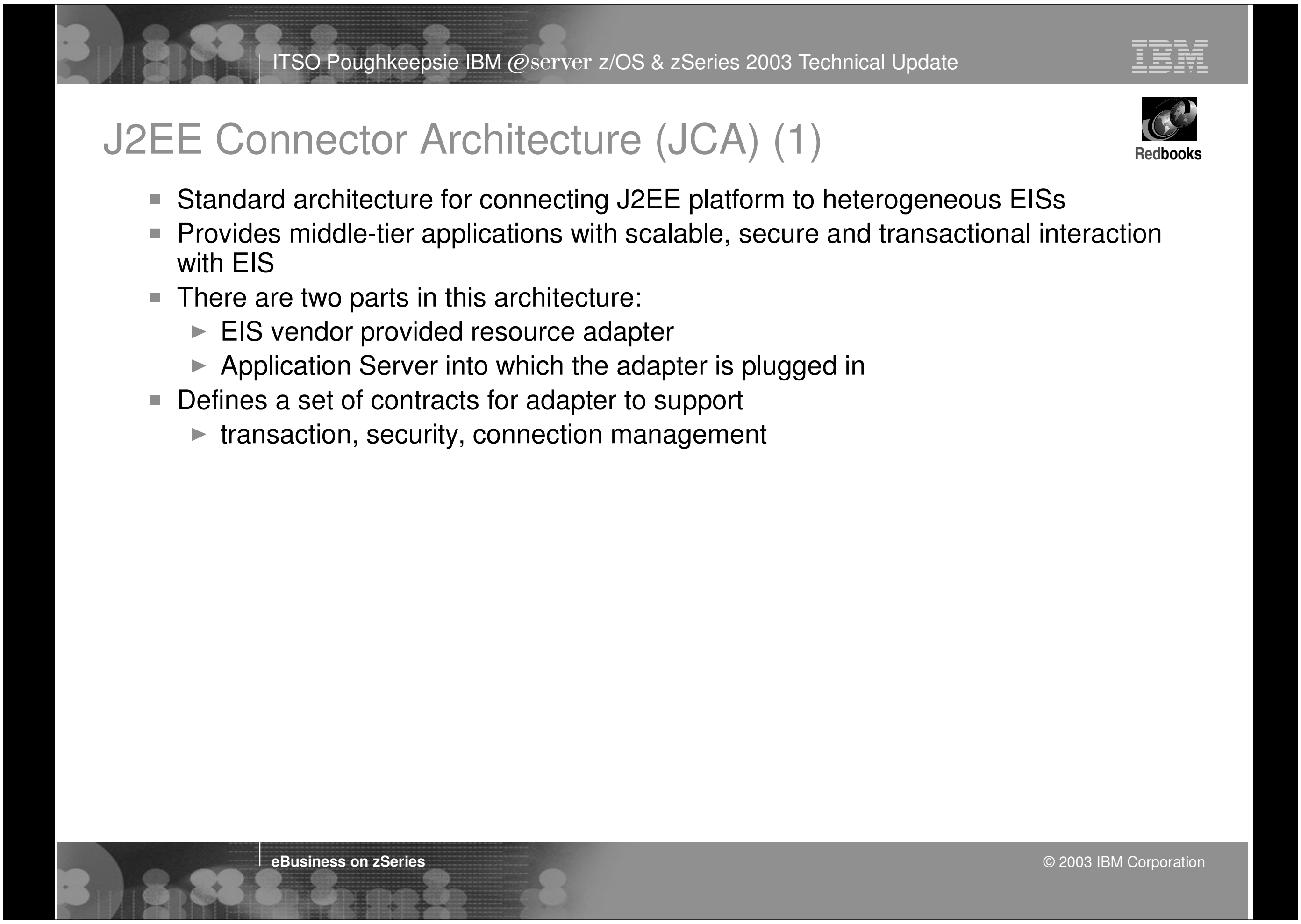
Managed Environment





J2C/JCA specification

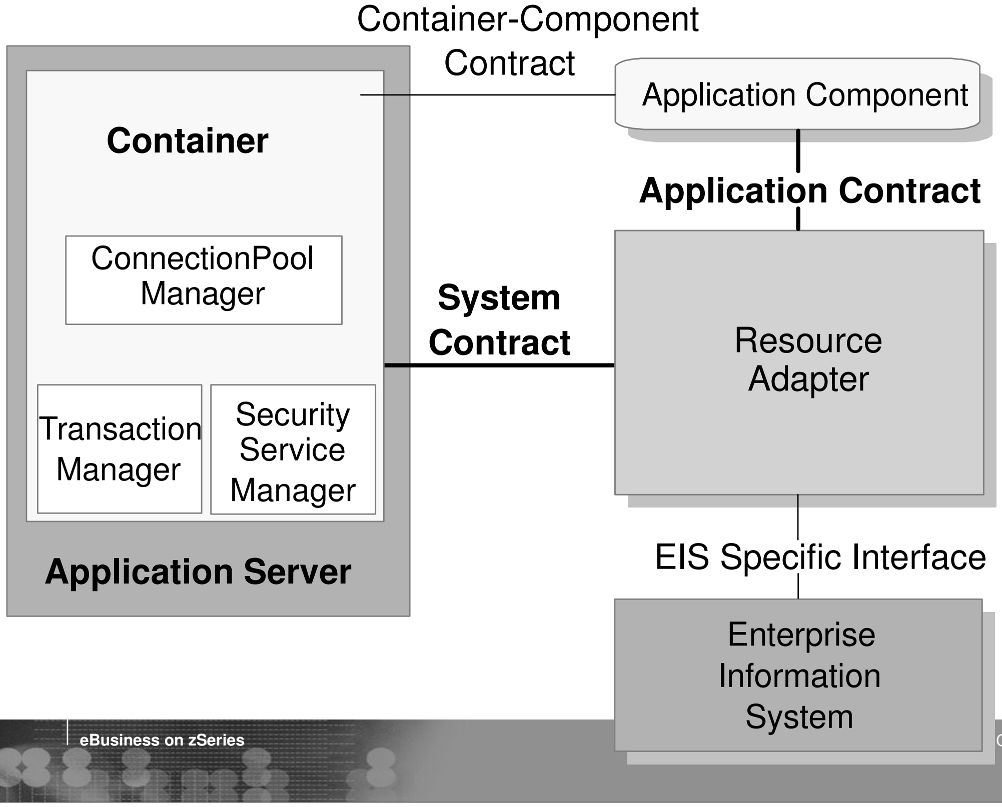
- JCA 1.0 is part of J2EE 1.3 spec.
  - implemented in WebSphere for z/OS V4.0.1 already
- J2EE specification defines
  - *System-level Programming Interface (SPI)* between application server and EIS
  - *Common Client Interface (CCI)* between a client application and EIS
  - deployment and packaging protocol for resource adapters
    - Resource Adapter Archive (RAR) file



J2EE Connector Architecture (JCA) (1)

- Standard architecture for connecting J2EE platform to heterogeneous EISs
- Provides middle-tier applications with scalable, secure and transactional interaction with EIS
- There are two parts in this architecture:
  - EIS vendor provided resource adapter
  - Application Server into which the adapter is plugged in
- Defines a set of contracts for adapter to support
  - transaction, security, connection management

J2EE Connector Architecture (JCA) (2)



Container-Component Contract

Application Component

Application Contract

Container

ConnectionPool Manager

Transaction Manager

Security Service Manager

Application Server

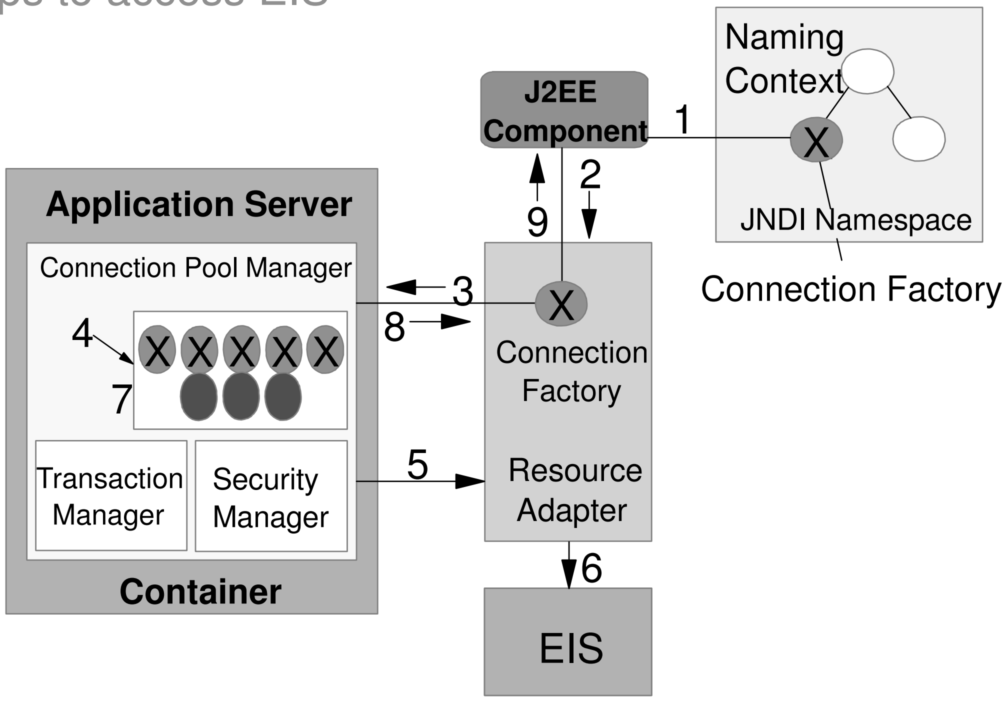
System Contract

Resource Adapter

EIS Specific Interface

Enterprise Information System

Steps to access EIS



Naming Context

JNDI Namespace

Connection Factory

J2EE Component

1

2

9

Application Server

Connection Pool Manager

4

7

3

8

Connection Factory

Resource Adapter

5

Transaction Manager

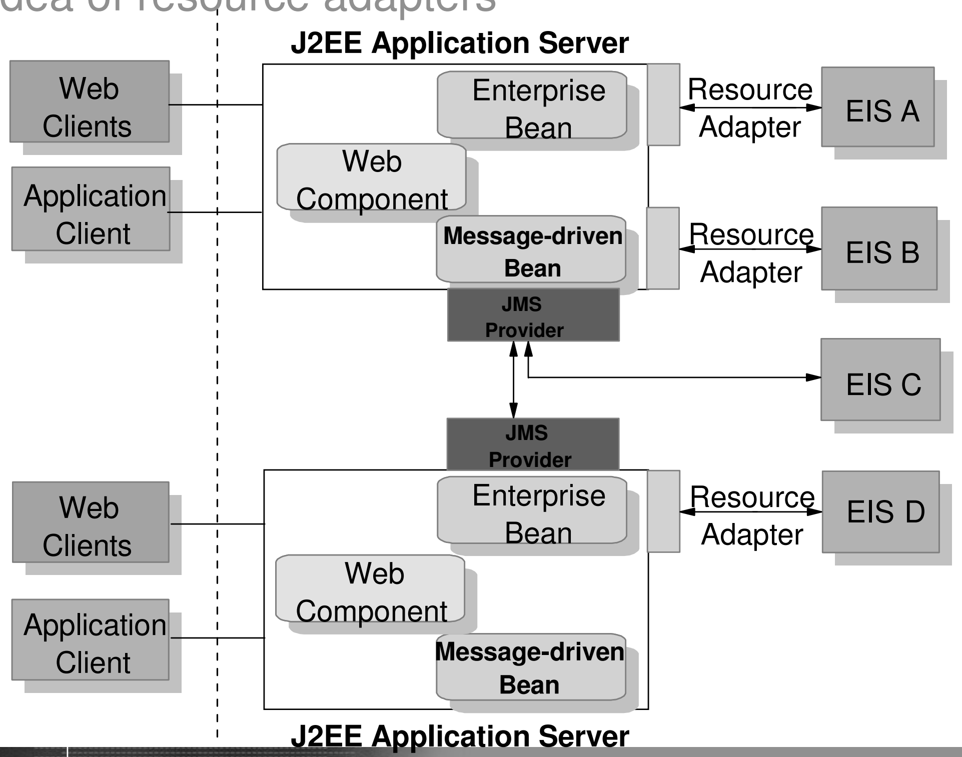
Security Manager

Container

6

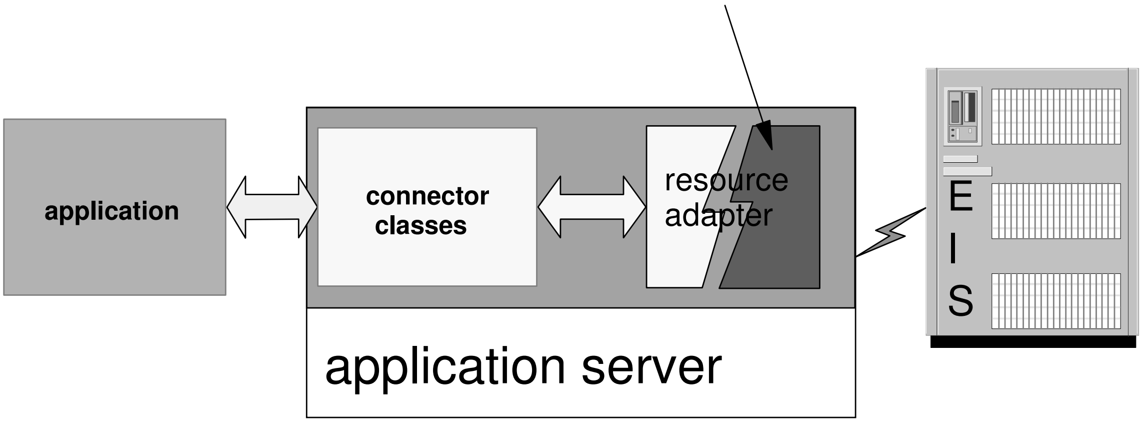
EIS

The idea of resource adapters



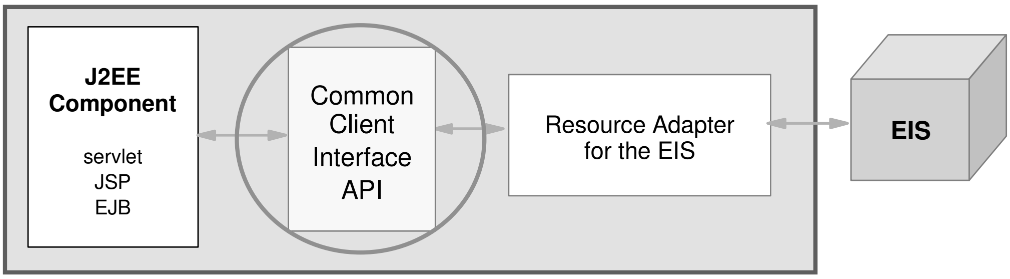
Resource Adapters

- Resource Adapters (RA) connect EIS to the J2EE connectors
  - analogue to JDBC driver
  - collaborate with application server to provide services
- Supplied by EIS or third party vendor
  - plug in point is an application server extension
  - J2EE architecture lets vendor provide a standard RA for all J2EE compliant servers



Common Client Interface (CCI)

- CCI API provides access from J2EE components to an EIS through a Resource Adapter
  - makes coding easy and potentially independent of EIS
  - J2EE components interact with CCI only - not directly with Resource Adapter
- Packaged with WebSphere in V4.x



Before the JCA and CCI

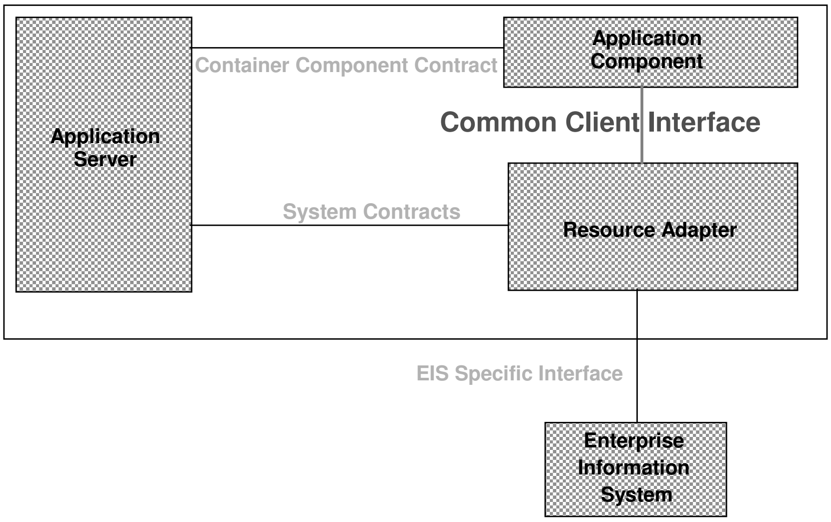
- IBM has been ahead of the industry recognizing the need for a consistent connector interface
  - first delivered Common Connector Framework (CCF) with VisualAge for Java (VAJ), version 2 in 1998
  - contributed CCF design to Sun's J2EE Connector Architecture specification
- JCA is very close in design to CCF
    - Migration is almost automatic
  - VAJ 3.5.3 and 4 included Enterprise Access Builder tooling to build CCF components
  - WSAD supports JCA as a default for new development, but also supports the ***usage*** of in VAJ developed CCF command beans

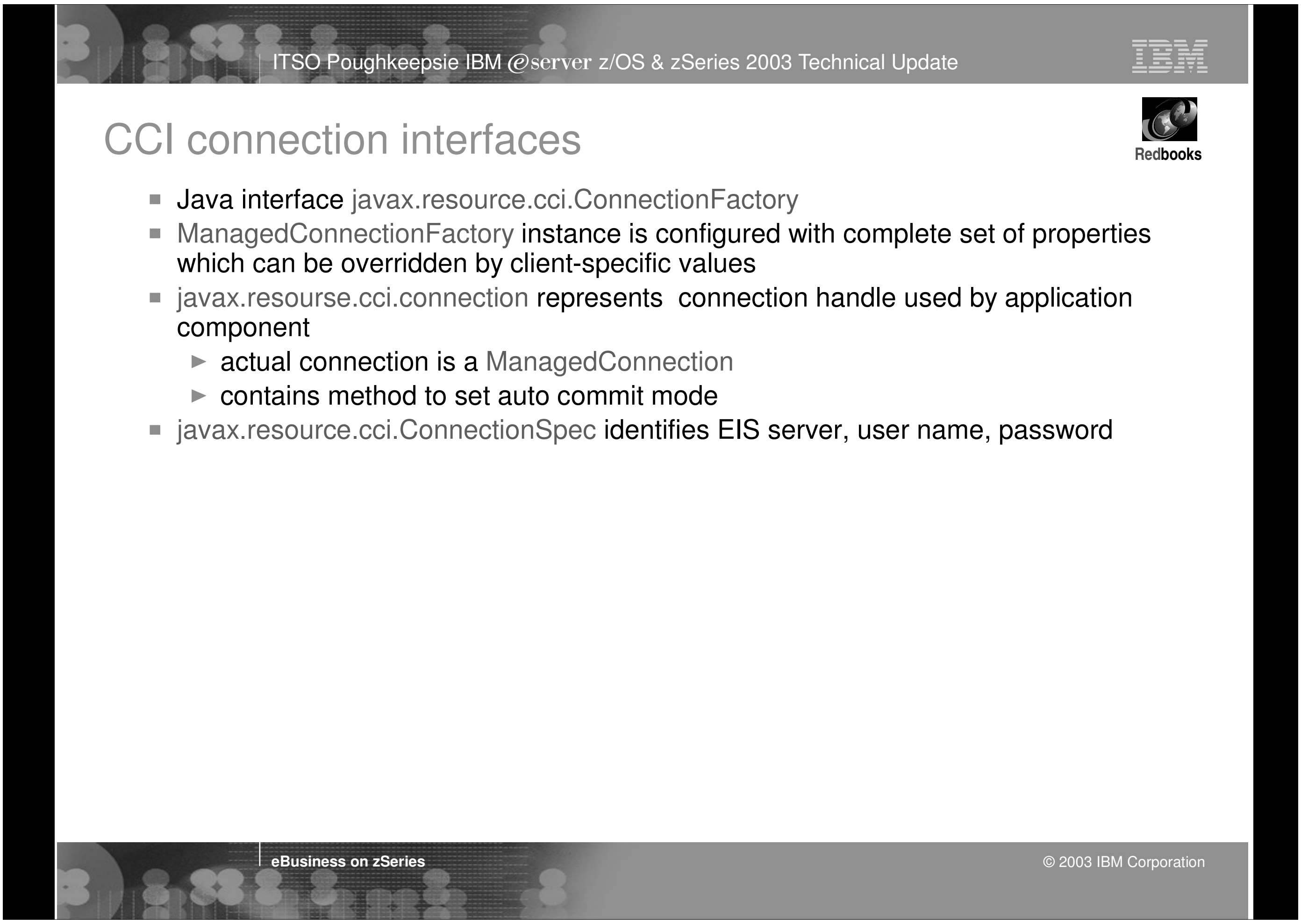
JCA programming model - main steps

1. The application assembler or component provider sets connection factory requirements in deployment descriptor
2. The application component looks up the factory reference using JNDI
3. The application component requests a connection, uses it, and when finished drops it
4. Connections are managed by the connection manager which is part of the application server

Common Client Interface (CCI)

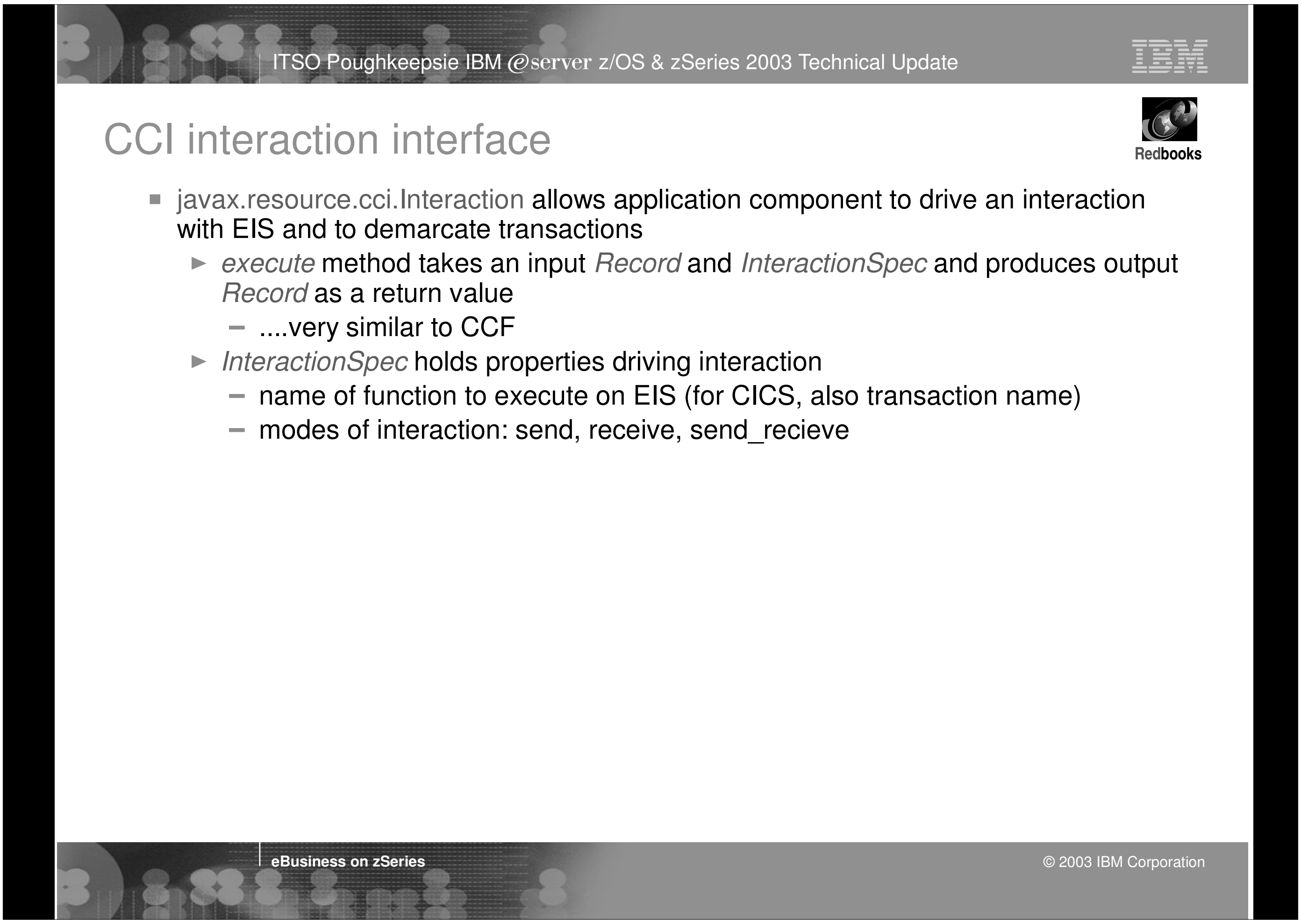
- CCI is the same in managed and non-managed mode, but in non-managed mode the client manages connection and QoS





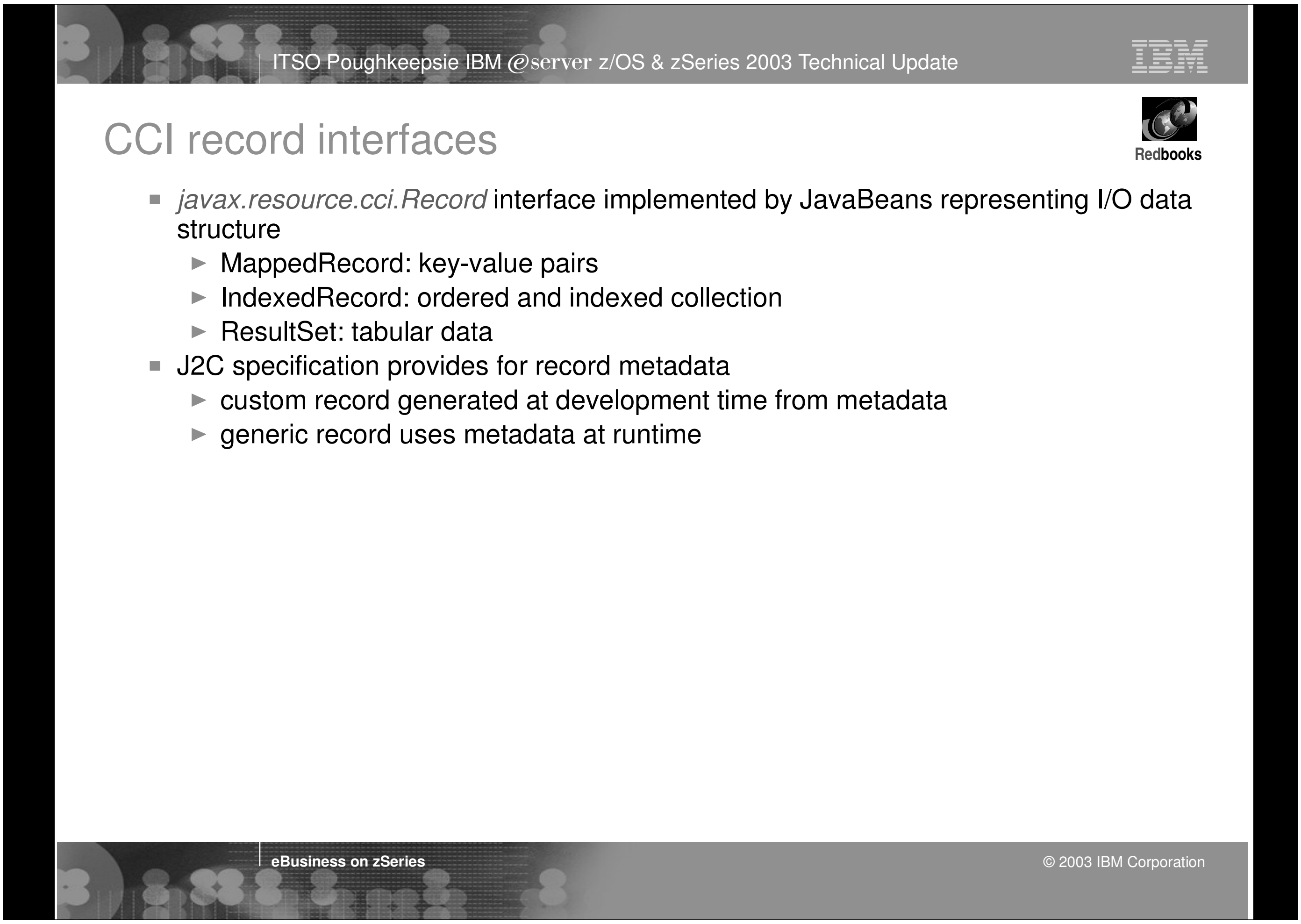
CCI connection interfaces

- Java interface javax.resource.cci.ConnectionFactory
- ManagedConnectionFactory instance is configured with complete set of properties which can be overridden by client-specific values
- javax.resourse.cci.connection represents connection handle used by application component
  - actual connection is a ManagedConnection
  - contains method to set auto commit mode
- javax.resource.cci.ConnectionSpec identifies EIS server, user name, password



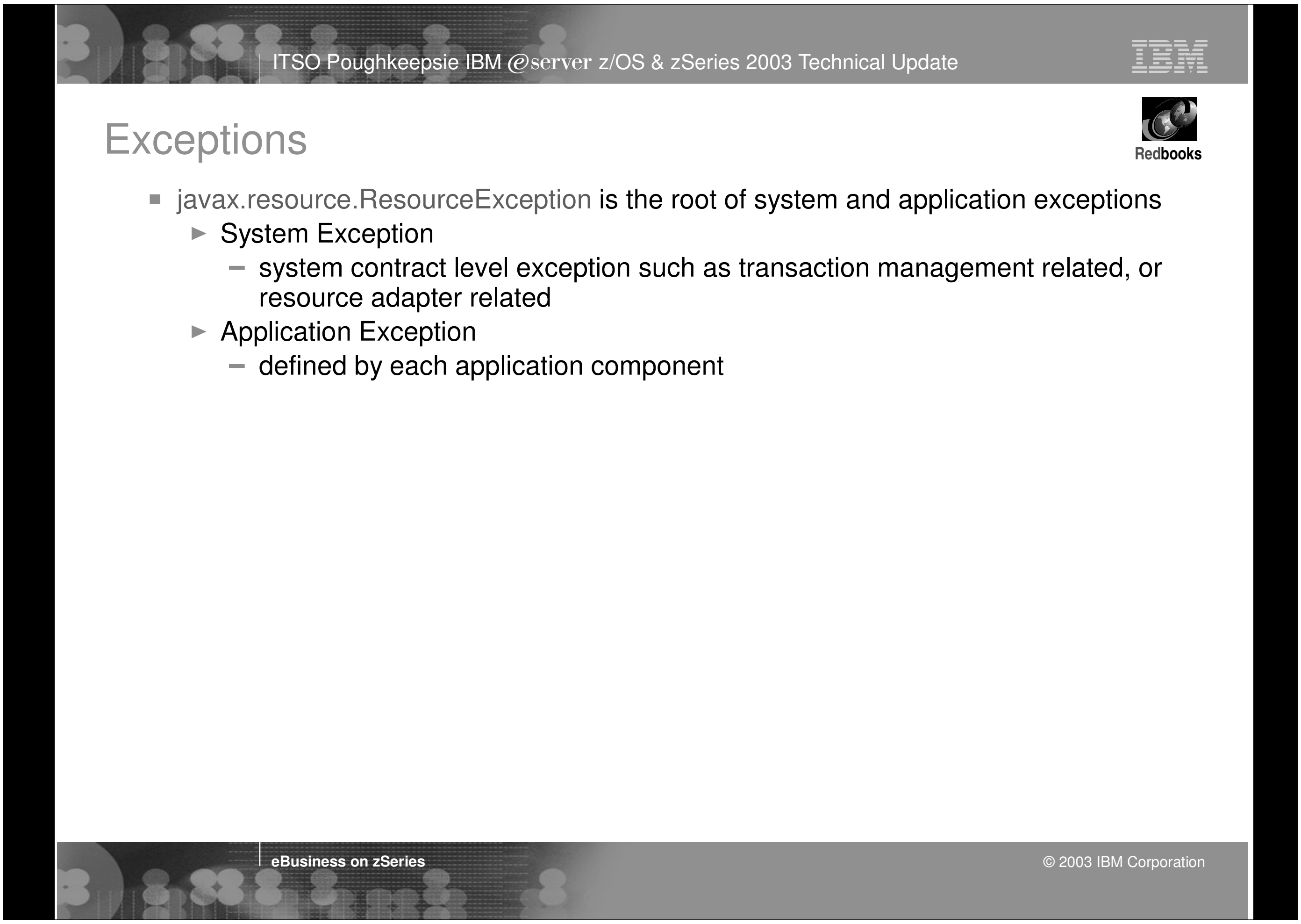
CCI interaction interface

- javax.resource.cci.Interaction allows application component to drive an interaction with EIS and to demarcate transactions
  - *execute* method takes an input *Record* and *InteractionSpec* and produces output *Record* as a return value
    - ....very similar to CCF
  - *InteractionSpec* holds properties driving interaction
    - name of function to execute on EIS (for CICS, also transaction name)
    - modes of interaction: send, receive, send_recieve



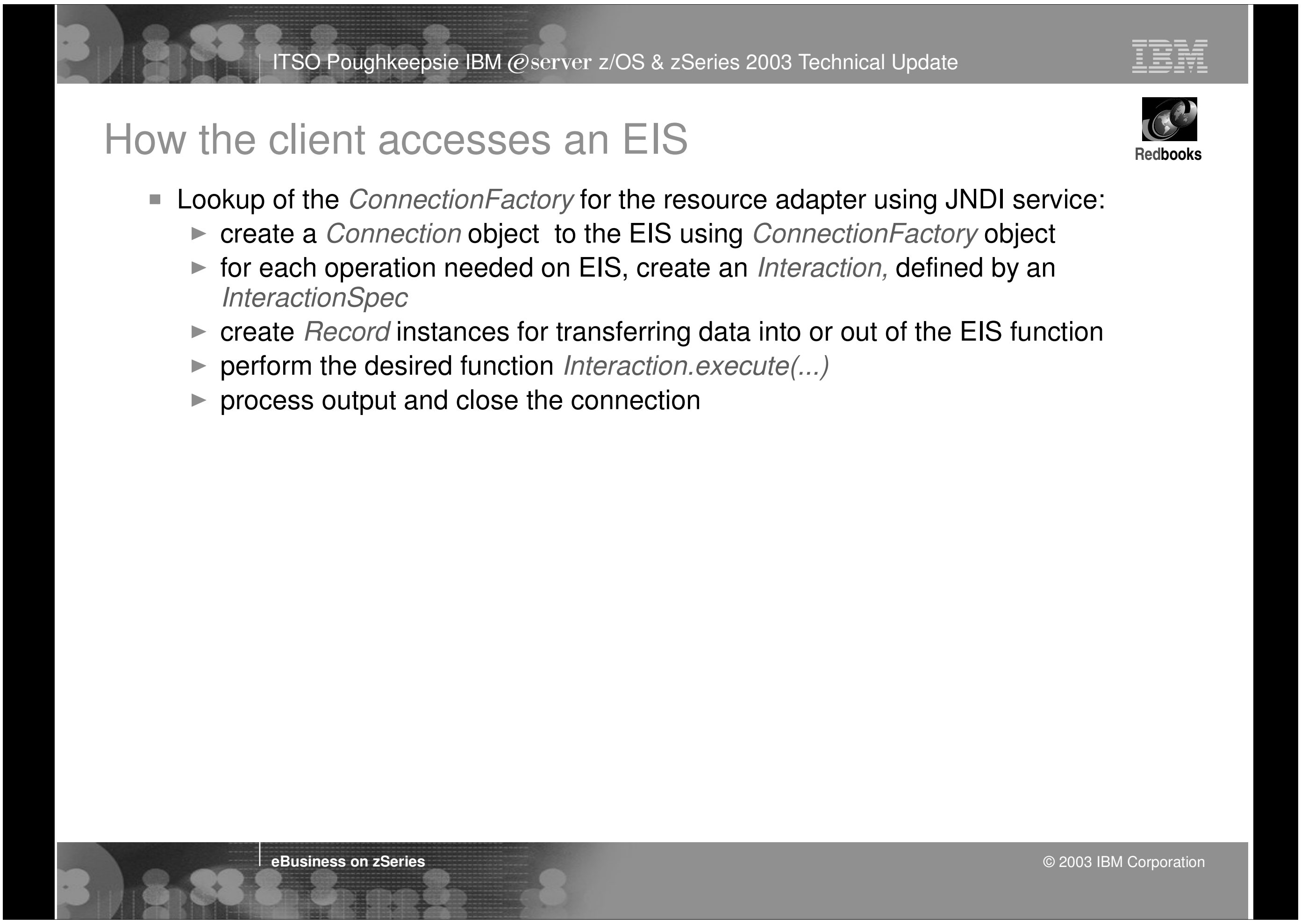
CCI record interfaces

- *javax.resource.cci.Record* interface implemented by JavaBeans representing I/O data structure
  - MappedRecord: key-value pairs
  - IndexedRecord: ordered and indexed collection
  - ResultSet: tabular data
- J2C specification provides for record metadata
  - custom record generated at development time from metadata
  - generic record uses metadata at runtime



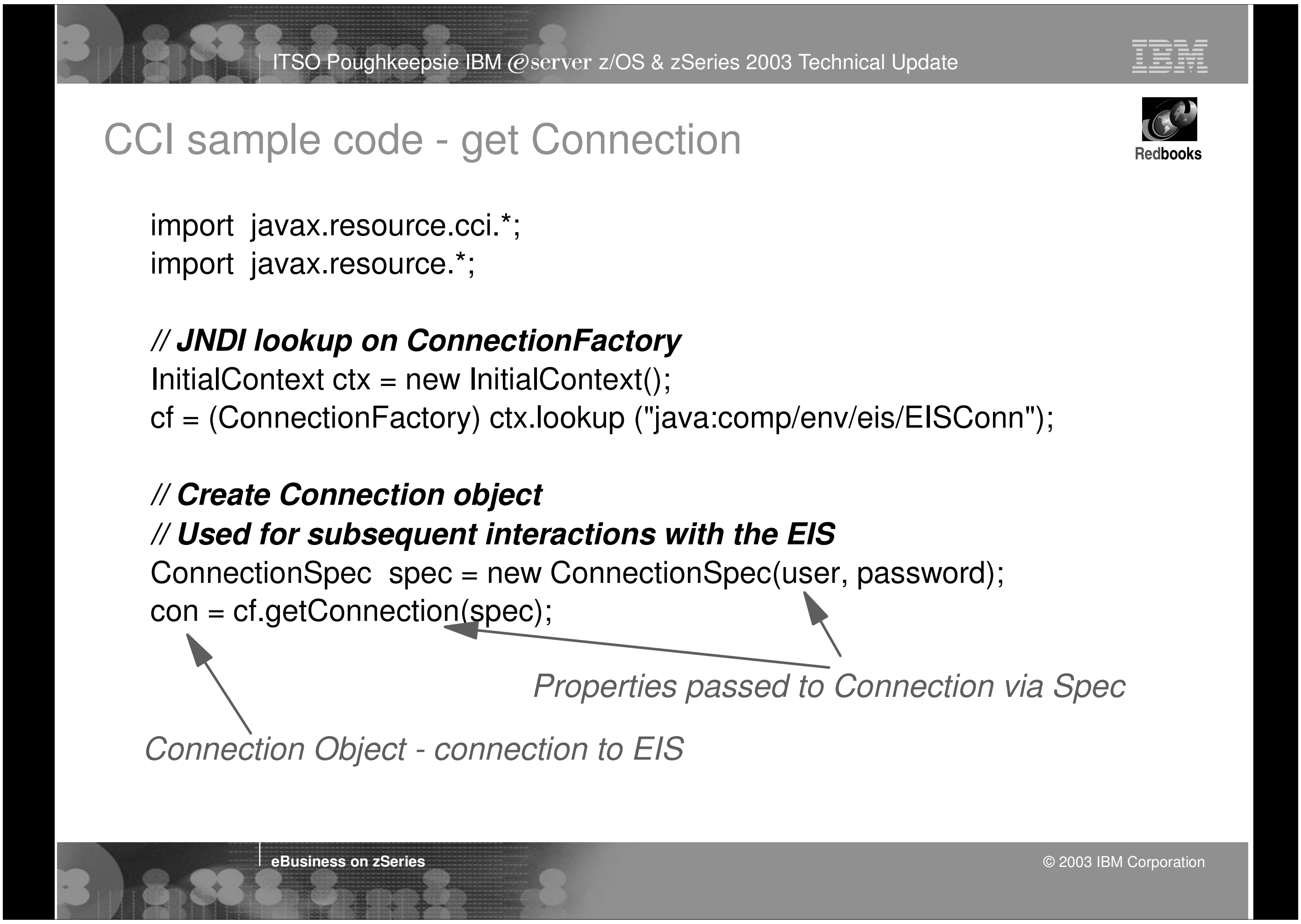
Exceptions

- javax.resource.ResourceException is the root of system and application exceptions
  - System Exception
    - system contract level exception such as transaction management related, or resource adapter related
  - Application Exception
    - defined by each application component



How the client accesses an EIS

- Lookup of the *ConnectionFactory* for the resource adapter using JNDI service:
  - create a *Connection* object to the EIS using *ConnectionFactory* object
  - for each operation needed on EIS, create an *Interaction,* defined by an *InteractionSpec*
  - create *Record* instances for transferring data into or out of the EIS function
  - perform the desired function *Interaction.execute(...)*
  - process output and close the connection

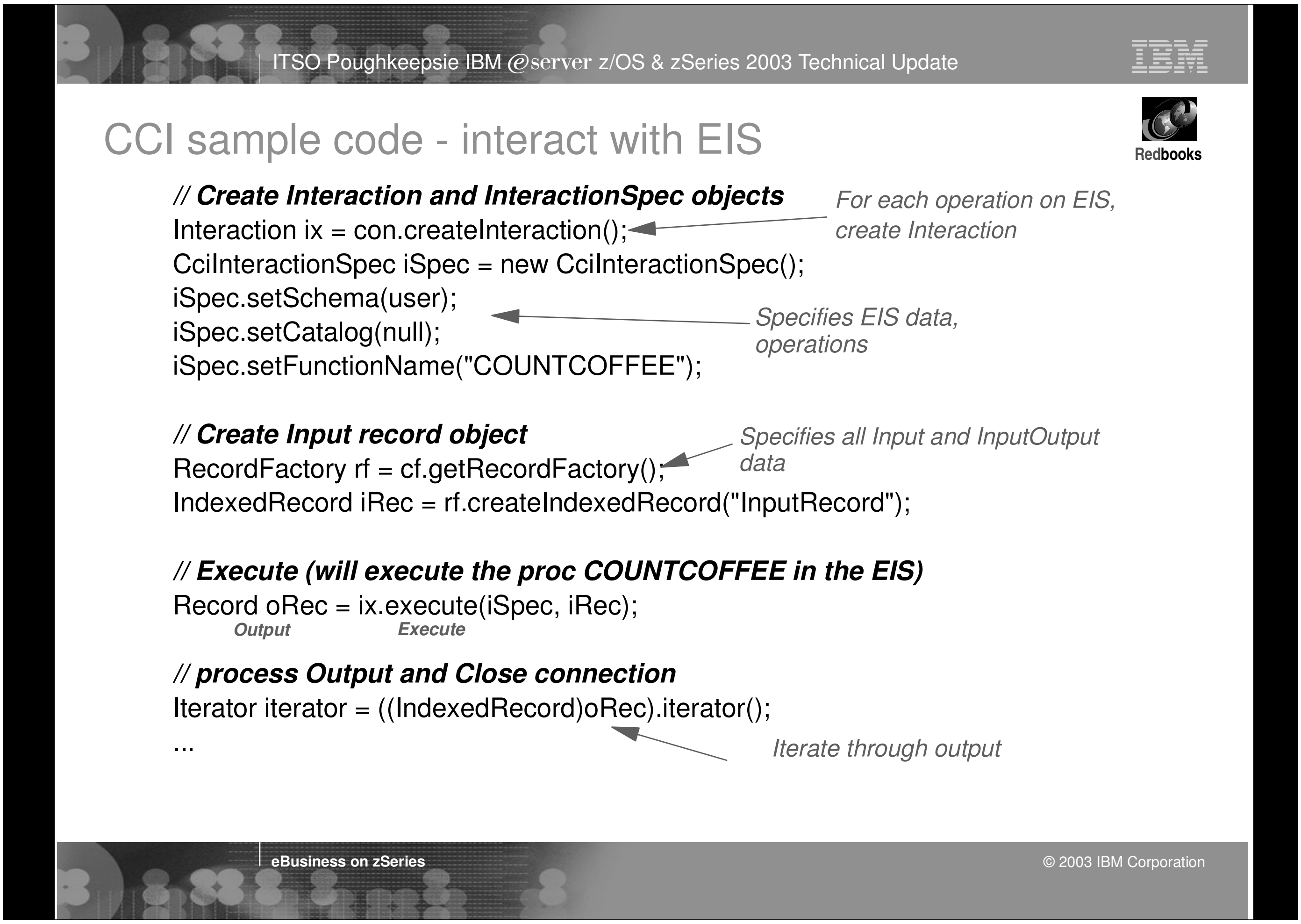


CCI sample code - get Connection

```
import  javax.resource.cci.*;
import  javax.resource.*;

// JNDI lookup on ConnectionFactory
InitialContext ctx = new InitialContext();
cf = (ConnectionFactory) ctx.lookup ("java:comp/env/eis/EISConn");

// Create Connection object
// Used for subsequent interactions with the EIS
ConnectionSpec  spec = new ConnectionSpec(user, password);
con = cf.getConnection(spec);
```

Properties passed to Connection via Spec

Connection Object - connection to EIS



CCI sample code - interact with EIS

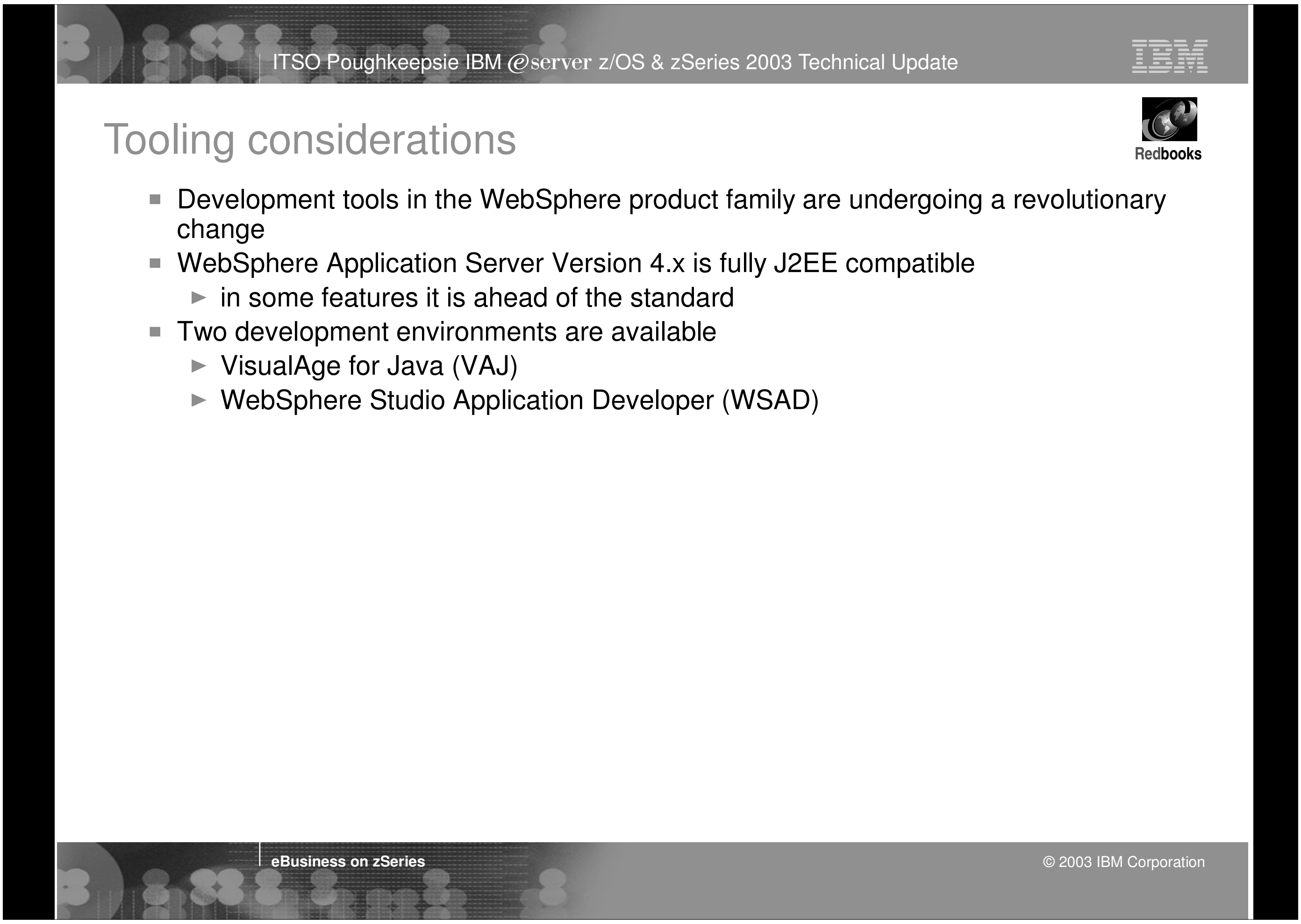
```
// Create Interaction and InteractionSpec objects
Interaction ix = con.createInteraction();
CciInteractionSpec iSpec = new CciInteractionSpec();
iSpec.setSchema(user);
iSpec.setCatalog(null);
iSpec.setFunctionName("COUNTCOFFEE");

// Create Input record object
RecordFactory rf = cf.getRecordFactory();
IndexedRecord iRec = rf.createIndexedRecord("InputRecord");

// Execute (will execute the proc COUNTCOFFEE in the EIS)
Record oRec = ix.execute(iSpec, iRec);
       Output         Execute

// process Output and Close connection
Iterator iterator = ((IndexedRecord)oRec).iterator();
...
```

For each operation on EIS, create Interaction

Specifies EIS data, operations

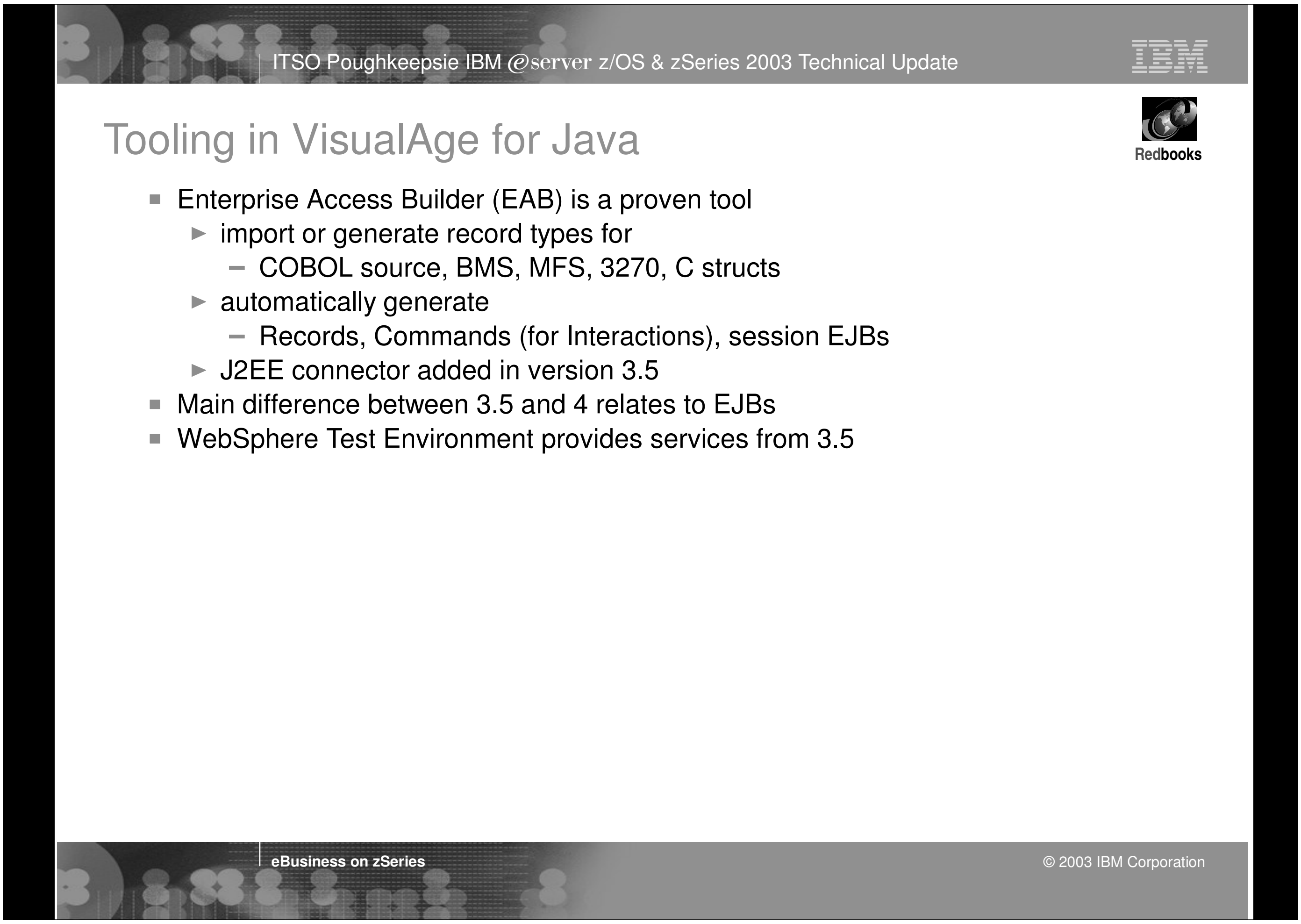
Specifies all Input and InputOutput data

Iterate through output



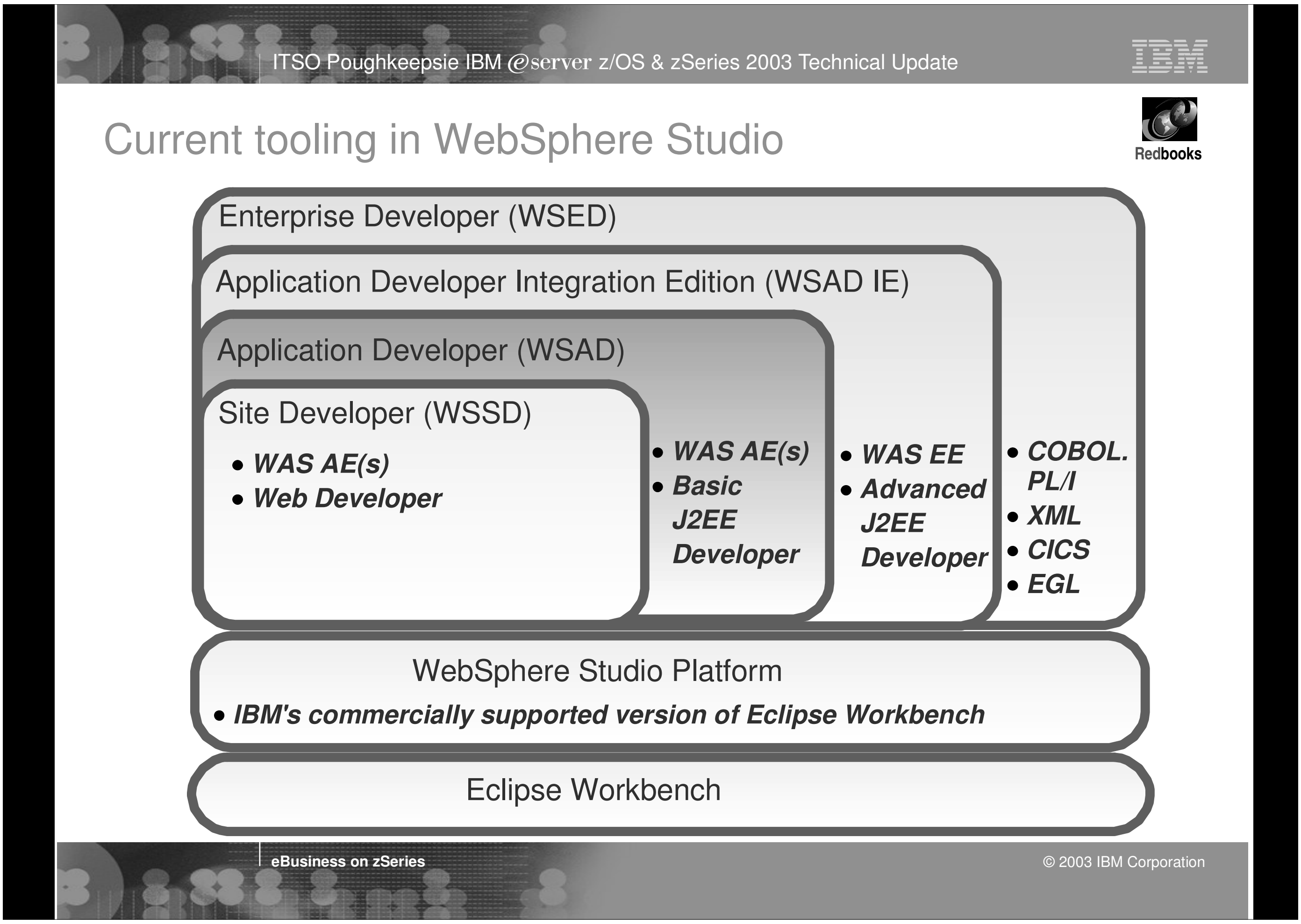
Tooling considerations

- Development tools in the WebSphere product family are undergoing a revolutionary change
- WebSphere Application Server Version 4.x is fully J2EE compatible
  - in some features it is ahead of the standard
- Two development environments are available
  - VisualAge for Java (VAJ)
  - WebSphere Studio Application Developer (WSAD)



Tooling in VisualAge for Java

- Enterprise Access Builder (EAB) is a proven tool
  - import or generate record types for
    - COBOL source, BMS, MFS, 3270, C structs
  - automatically generate
    - Records, Commands (for Interactions), session EJBs
  - J2EE connector added in version 3.5
- Main difference between 3.5 and 4 relates to EJBs
- WebSphere Test Environment provides services from 3.5



Current tooling in WebSphere Studio

Enterprise Developer (WSED)

Application Developer Integration Edition (WSAD IE)

Application Developer (WSAD)

Site Developer (WSSD)

- ***WAS AE(s)***
- ***Web Developer***

- ***WAS AE(s)***
- ***Basic J2EE Developer***

- ***WAS EE***
- ***Advanced J2EE Developer***

- ***COBOL. PL/I***
- ***XML***
- ***CICS***
- ***EGL***

WebSphere Studio Platform

- ***IBM's commercially supported version of Eclipse Workbench***

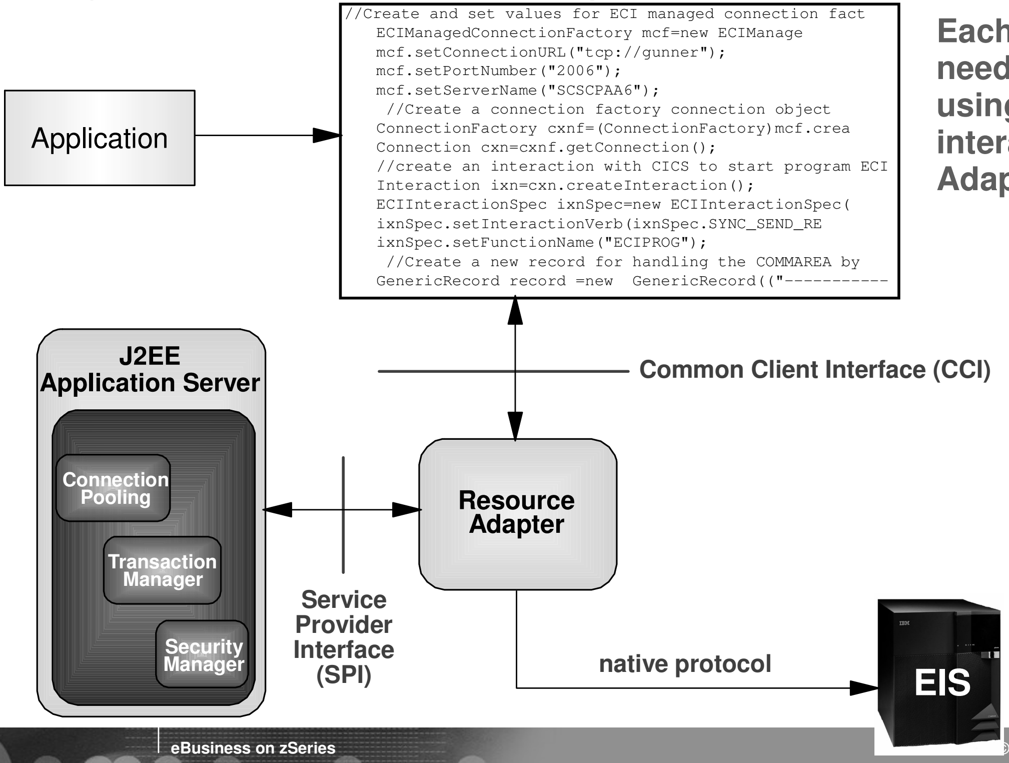
Eclipse Workbench

Migration from VAJ to WSAD

- VisualAge for Java supports tooling for migration assistance from CCF to J2EE Connector Architecture
- VisualAge for Java's Enterprise Access Builder provides tooling to migrate existing EAB commands, Navigators and Java Record Beans from CCF to J2EE Connector Architecture
- You can test applications that use JCA in WSAD
  - Export EAB components generated in VisualAge for Java
  - Add jars to project build path in WSAD
  - For JCA connectors run only in non-managed mode
- Why do this?
  - Test in a WAS V4/V5 test environment
  - Use WSAD packaging for deployment
  - Integrate connectors into a larger application

Redpiece: REDP3784 Migrating from CCF to JCA
http://w3.itso.ibm.com/redpieces/abstracts/redp3784.html

Using J2EE/CA without WebSphere Studio

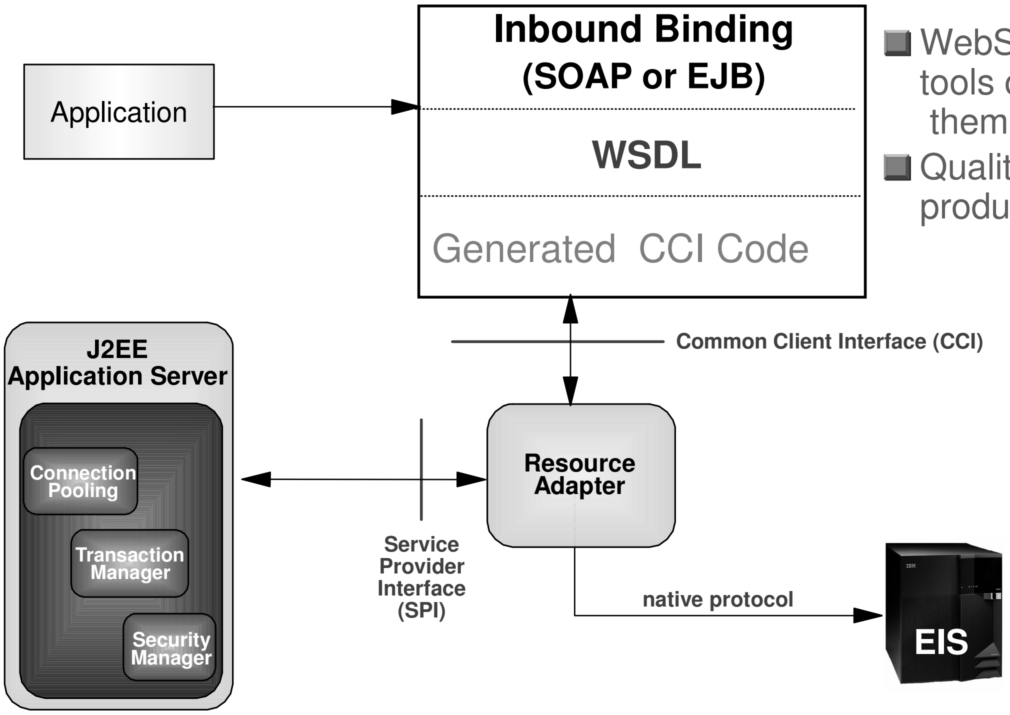


```
//Create and set values for ECI managed connection fact
   ECIManagedConnectionFactory mcf=new ECIManage
   mcf.setConnectionURL("tcp://gunner");
   mcf.setPortNumber("2006");
   mcf.setServerName("SCSCPAA6");
    //Create a connection factory connection object
   ConnectionFactory cxnf=(ConnectionFactory)mcf.crea
   Connection cxn=cxnf.getConnection();
   //create an interaction with CICS to start program ECI
   Interaction ixn=cxn.createInteraction();
   ECIInteractionSpec ixnSpec=new ECIInteractionSpec(
   ixnSpec.setInteractionVerb(ixnSpec.SYNC_SEND_RE
   ixnSpec.setFunctionName("ECIPROG");
    //Create a new record for handling the COMMAREA by
   GenericRecord record =new  GenericRecord(("-----------
```

Each application needs to code, using CCI API, to interact with Adapters!

Redbooks

Using J2EE/CA with WebSphere Studio

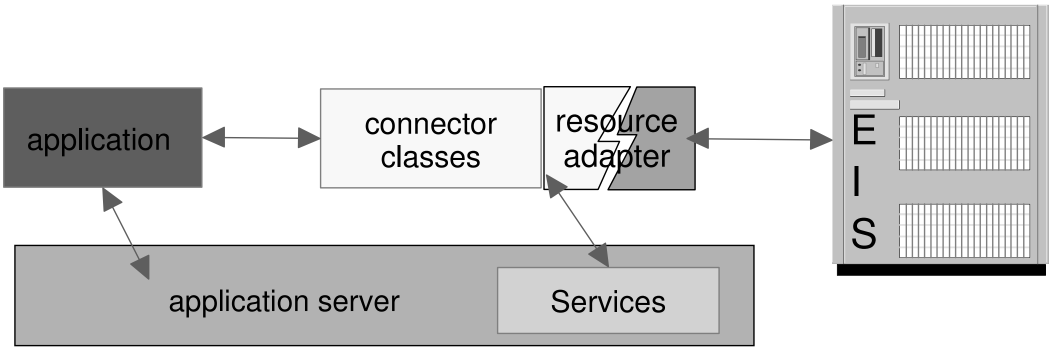


- WebSphere tools create them
- Quality and productivity!

Redbooks

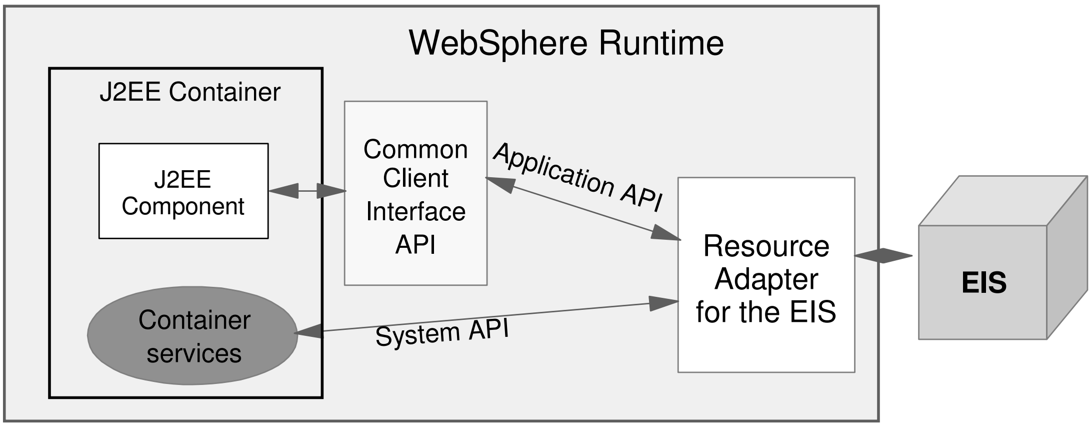
JCA interfaces

- Well defined interfaces specify interaction between main components
  - CCI is a standard Java API for applications that are clients to EIS
  - SPI is formalized as contracts
    - interface between application server and application code
    - interface between connector / resource adapter and application server services
    - Interface between resource adapter and EIS

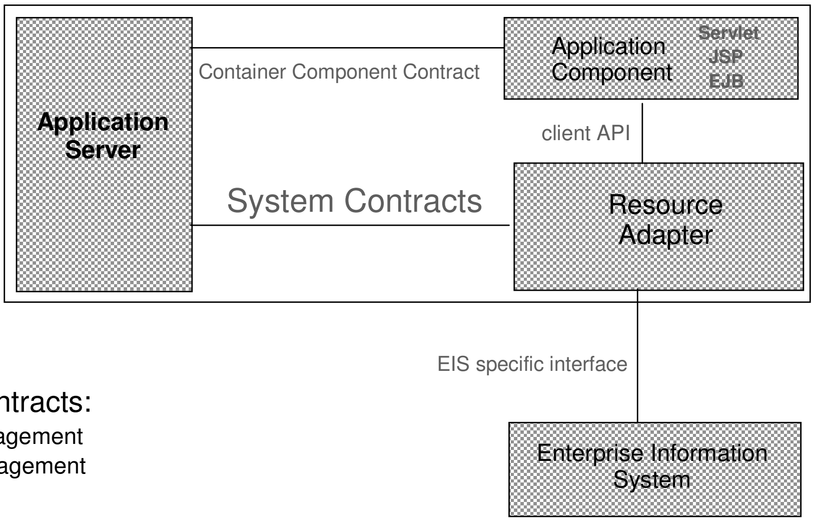


Resource Adapter API

- J2EE component accesses EIS through application API
- System API links the RA to J2EE server
- EIS-side of system contract implemented in RA to run in server address space



JCA contracts



- Key System Contracts:
  - Connection Management
  - Transaction Management
  - Security

Connection contract

- *ConnectionFactory* and *Connection* interfaces must be implemented by the resource adapter
- *ConnectionManager* is a hook for resource adapter to pass connections to the application server
  - implemented by application server
- Connection pool is implemented by the application server
  - competition point: scalability, efficiency, robustness

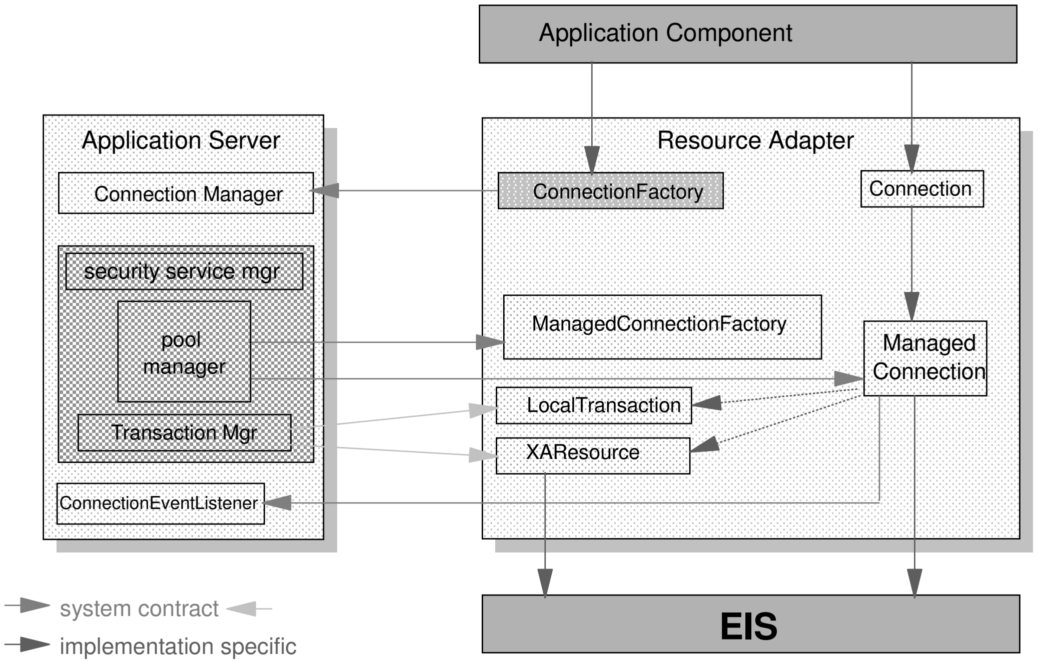
Connection Management

- Connection Management
  - provides a consistent programming model for connection acquisition in both managed and non-managed environments
- *ConnectionEventListener* enables the Application Server to get event notifications to:
    - manage connection pooling
    - manage transactions
    - cleanup connections
    - handle any error conditions
- Pool Manager lets the application server pool connections to an EIS:
  - for efficient utilization of system resources and EIS

Connection environments

- In a **managed** environment:
  - EIS adapter runs in a J2EE application server which is Web enabled
  - Business application component using the adapter runs in the Application Server
- In a **non-managed** environment:
  - application client directly uses the resource adapter to access EIS
- Resource adapter acts as a factory for connections
  - in a **managed** environment the resource adapter is not allowed to use its own internal connection pooling mechanism
  - the resource adapter provider can add connection pooling to its product for **non-managed** environments

Managed environment diagram



Application Component

Application Server

Connection Manager

security service mgr

pool manager

Transaction Mgr

ConnectionEventListener

Resource Adapter

ConnectionFactory

Connection

ManagedConnectionFactory

Managed Connection

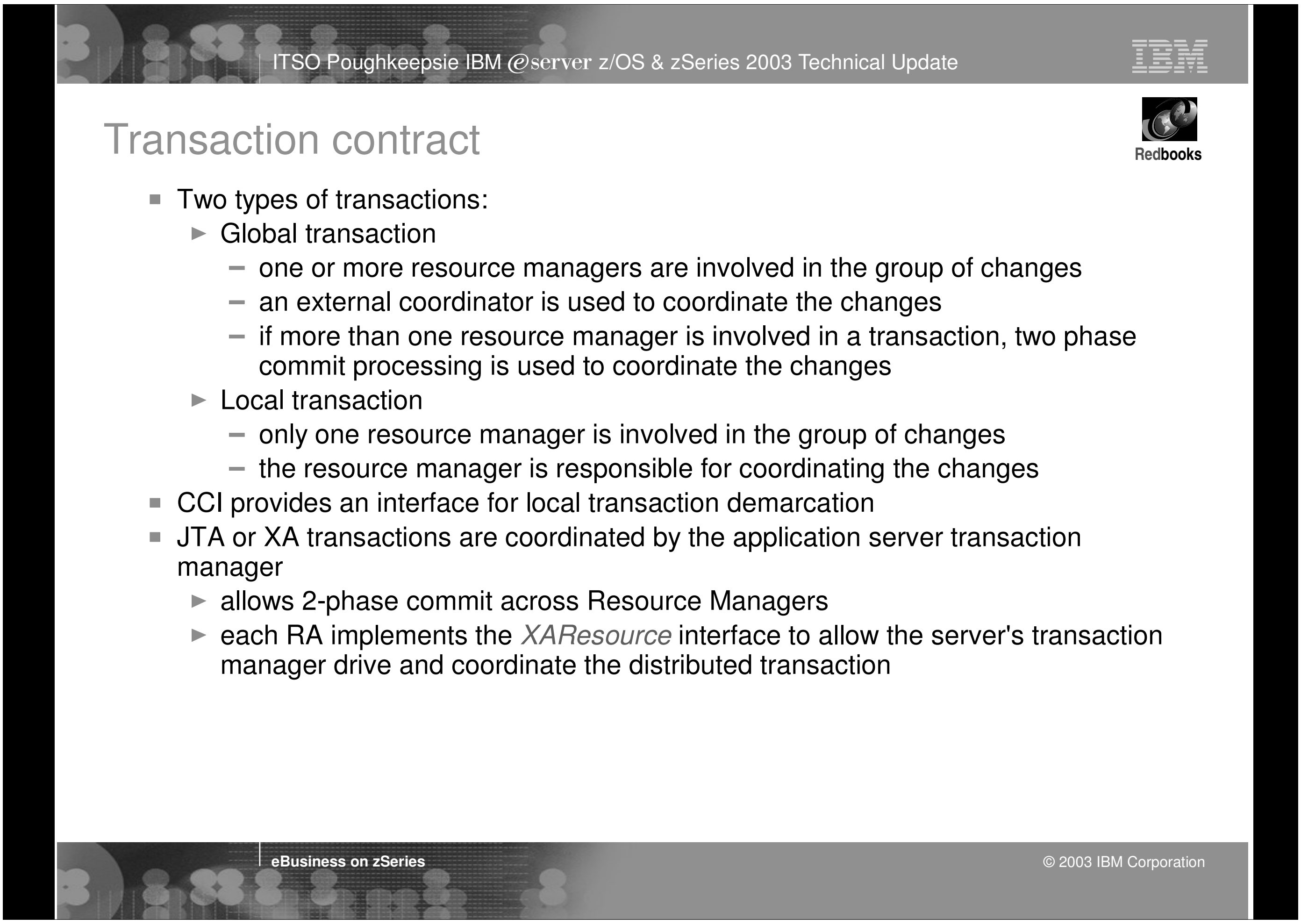
LocalTransaction

XAResource

EIS

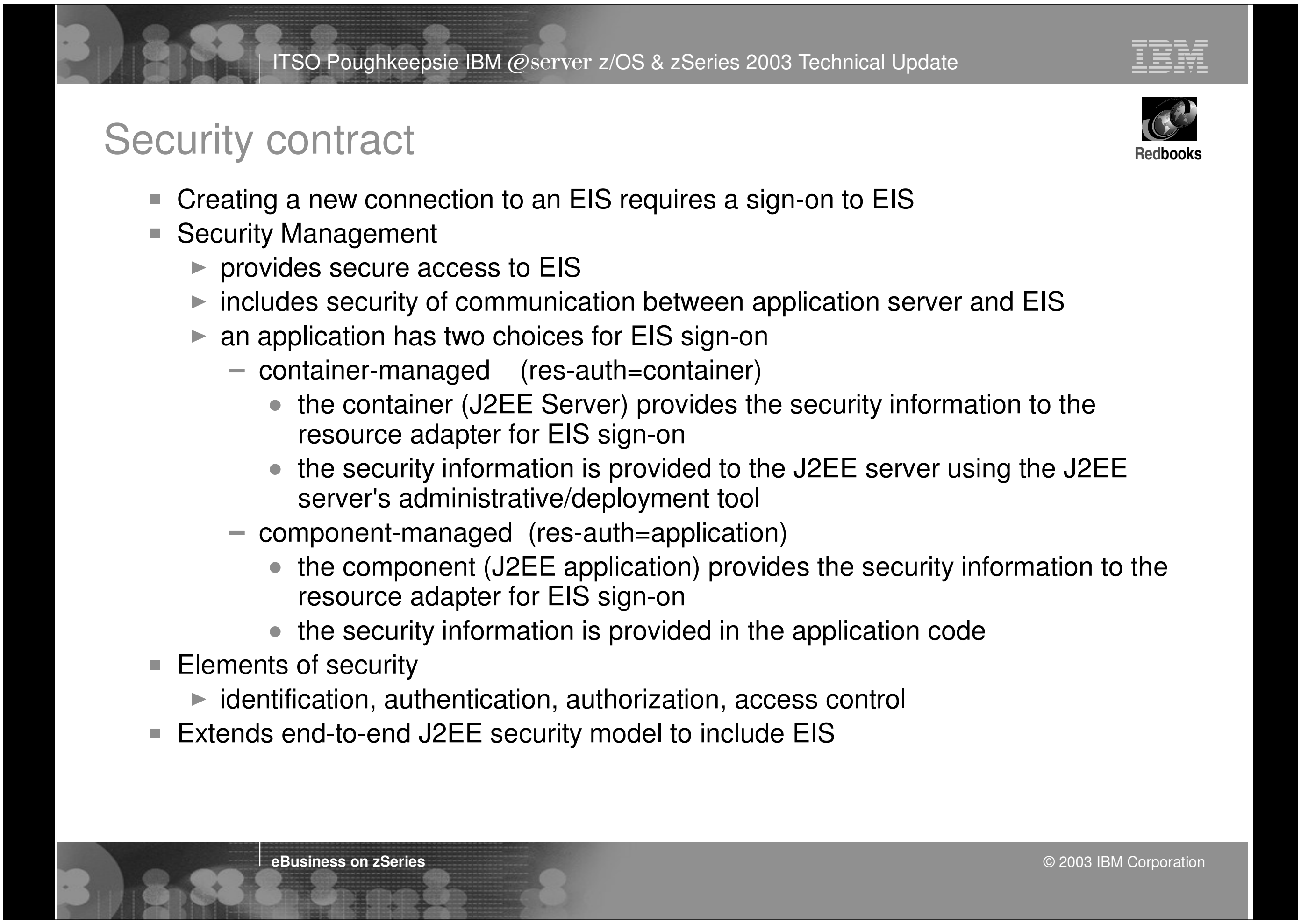
system contract

implementation specific



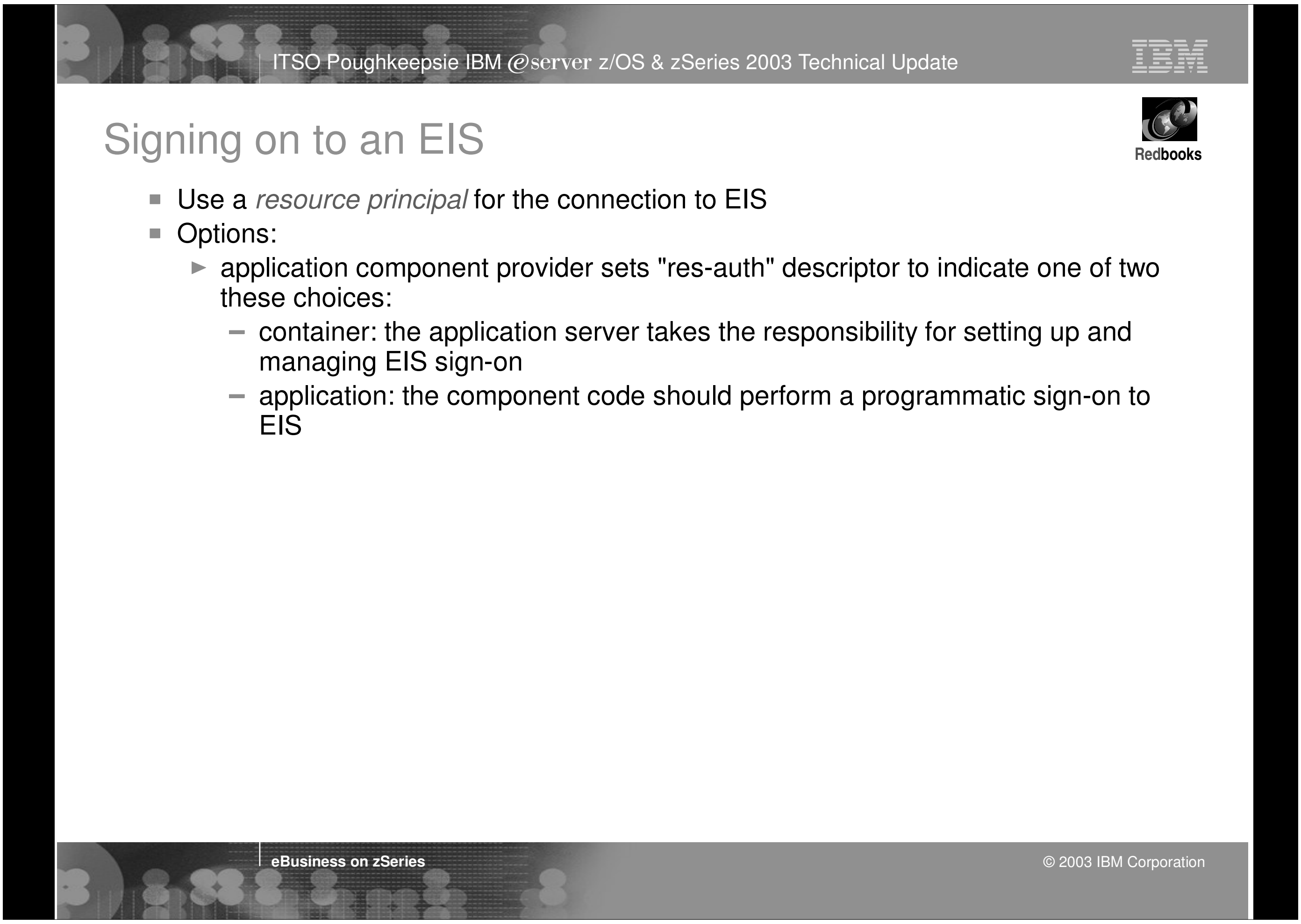
Transaction contract

- Two types of transactions:
  - Global transaction
    - one or more resource managers are involved in the group of changes
    - an external coordinator is used to coordinate the changes
    - if more than one resource manager is involved in a transaction, two phase commit processing is used to coordinate the changes
  - Local transaction
    - only one resource manager is involved in the group of changes
    - the resource manager is responsible for coordinating the changes
- CCI provides an interface for local transaction demarcation
- JTA or XA transactions are coordinated by the application server transaction manager
  - allows 2-phase commit across Resource Managers
  - each RA implements the *XAResource* interface to allow the server's transaction manager drive and coordinate the distributed transaction



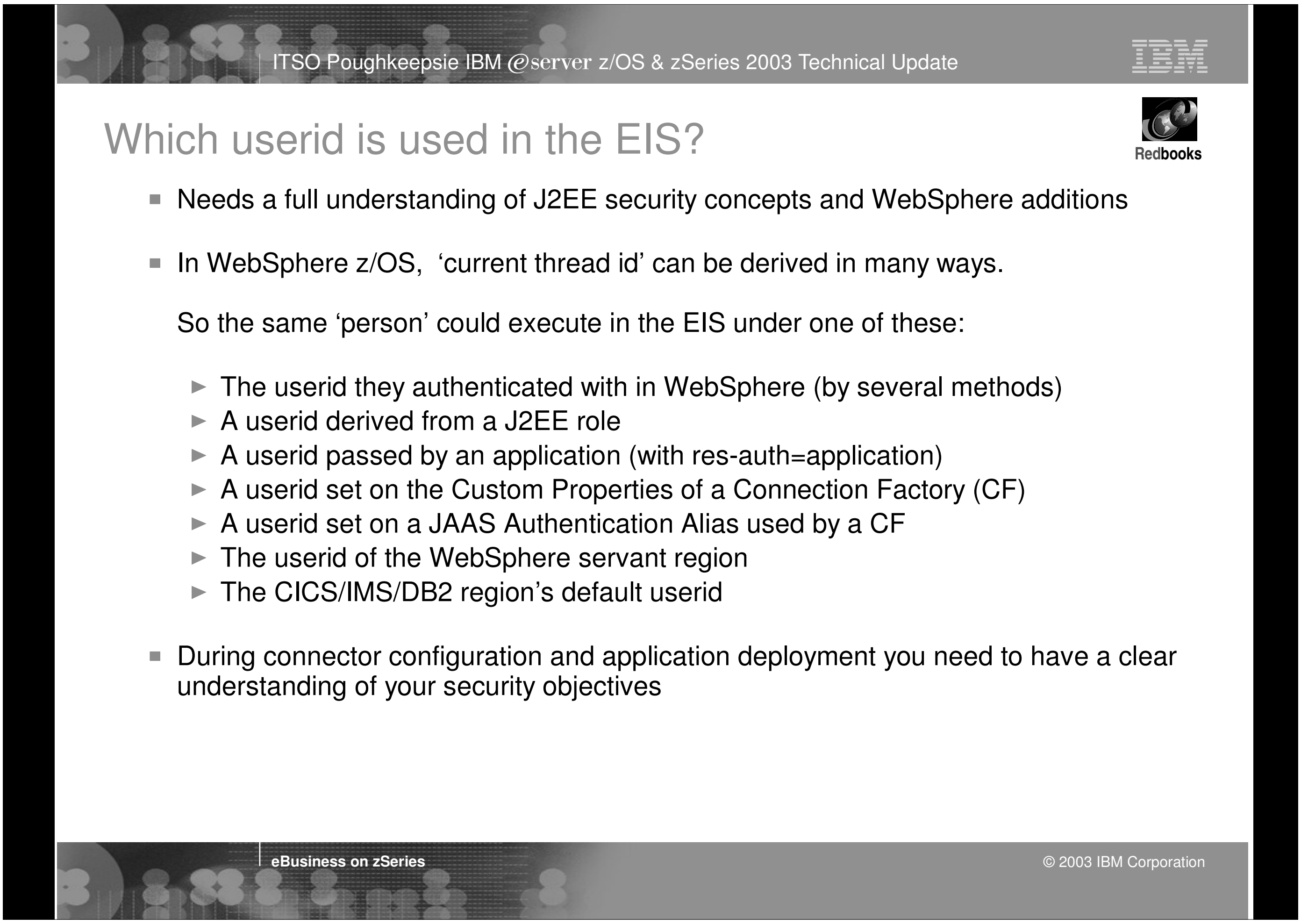
Security contract

- Creating a new connection to an EIS requires a sign-on to EIS
- Security Management
  - provides secure access to EIS
  - includes security of communication between application server and EIS
  - an application has two choices for EIS sign-on
    - container-managed (res-auth=container)
      - the container (J2EE Server) provides the security information to the resource adapter for EIS sign-on
      - the security information is provided to the J2EE server using the J2EE server's administrative/deployment tool
    - component-managed (res-auth=application)
      - the component (J2EE application) provides the security information to the resource adapter for EIS sign-on
      - the security information is provided in the application code
- Elements of security
  - identification, authentication, authorization, access control
- Extends end-to-end J2EE security model to include EIS



Signing on to an EIS

- Use a *resource principal* for the connection to EIS
- Options:
  - application component provider sets "res-auth" descriptor to indicate one of two these choices:
    - container: the application server takes the responsibility for setting up and managing EIS sign-on
    - application: the component code should perform a programmatic sign-on to EIS

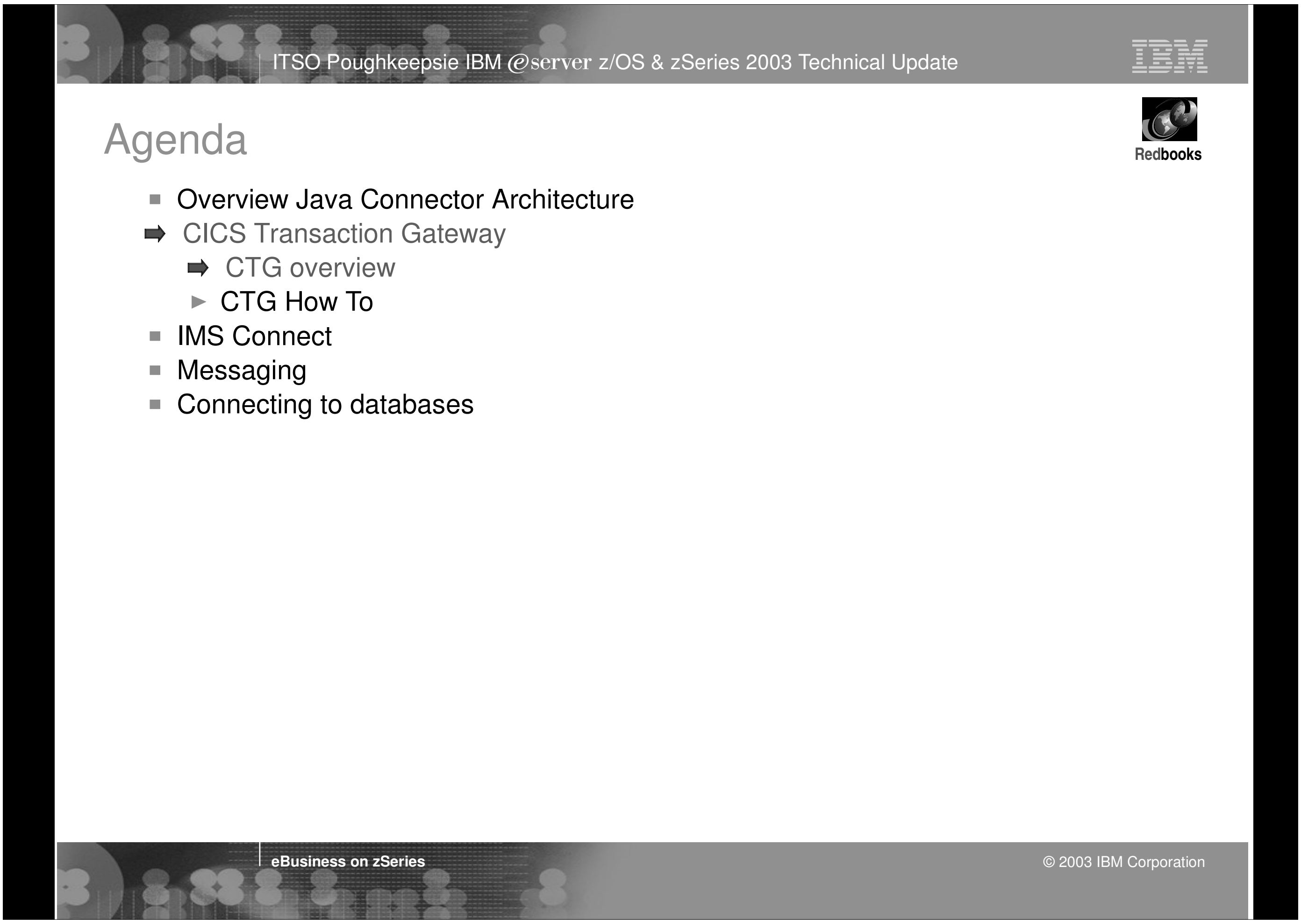


Which userid is used in the EIS?

- Needs a full understanding of J2EE security concepts and WebSphere additions
- In WebSphere z/OS, ‘current thread id’ can be derived in many ways.

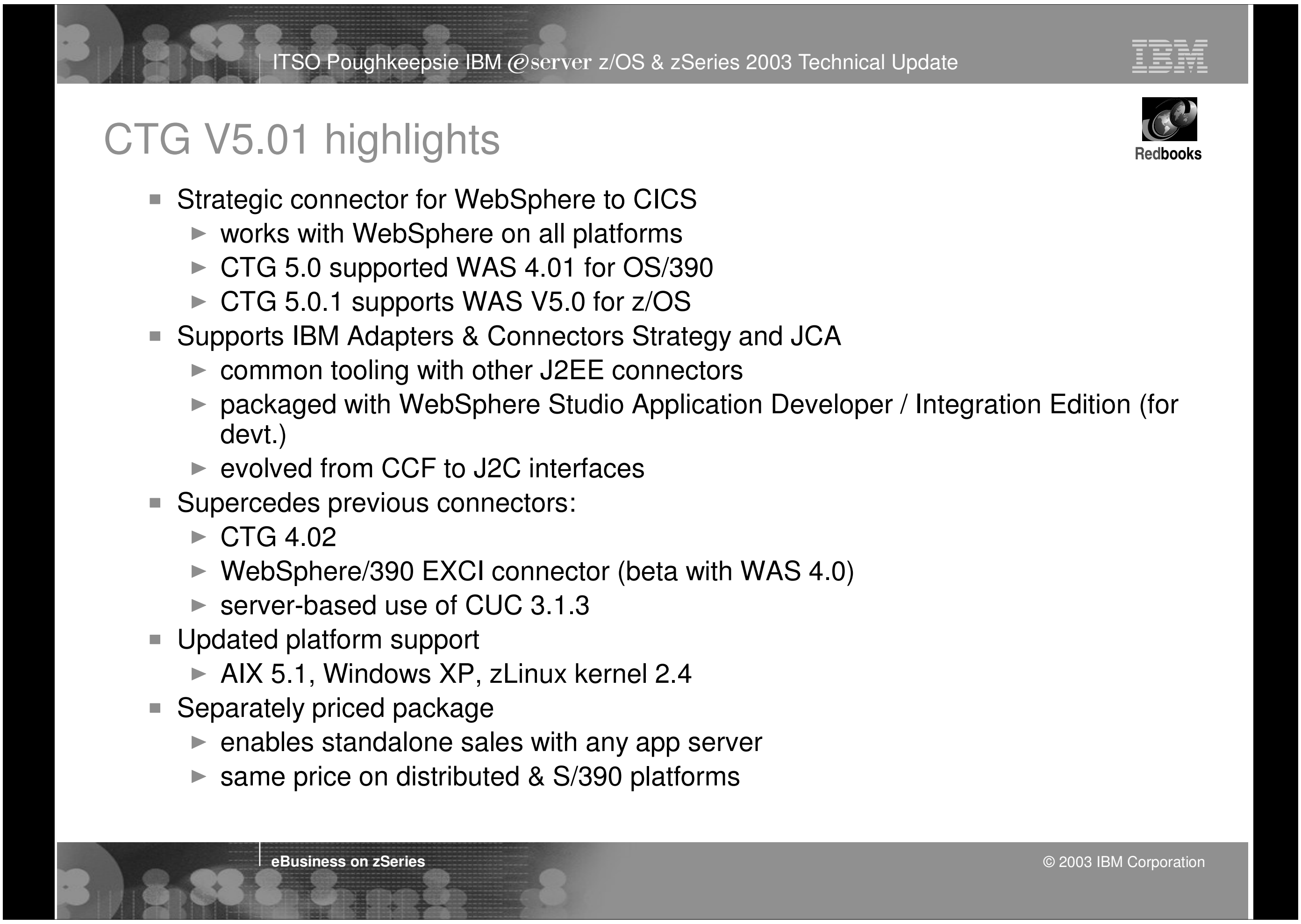
  So the same ‘person’ could execute in the EIS under one of these:

  - The userid they authenticated with in WebSphere (by several methods)
  - A userid derived from a J2EE role
  - A userid passed by an application (with res-auth=application)
  - A userid set on the Custom Properties of a Connection Factory (CF)
  - A userid set on a JAAS Authentication Alias used by a CF
  - The userid of the WebSphere servant region
  - The CICS/IMS/DB2 region’s default userid
- During connector configuration and application deployment you need to have a clear understanding of your security objectives



Agenda

- Overview Java Connector Architecture
- CICS Transaction Gateway
  - CTG overview
  - CTG How To
- IMS Connect
- Messaging
- Connecting to databases



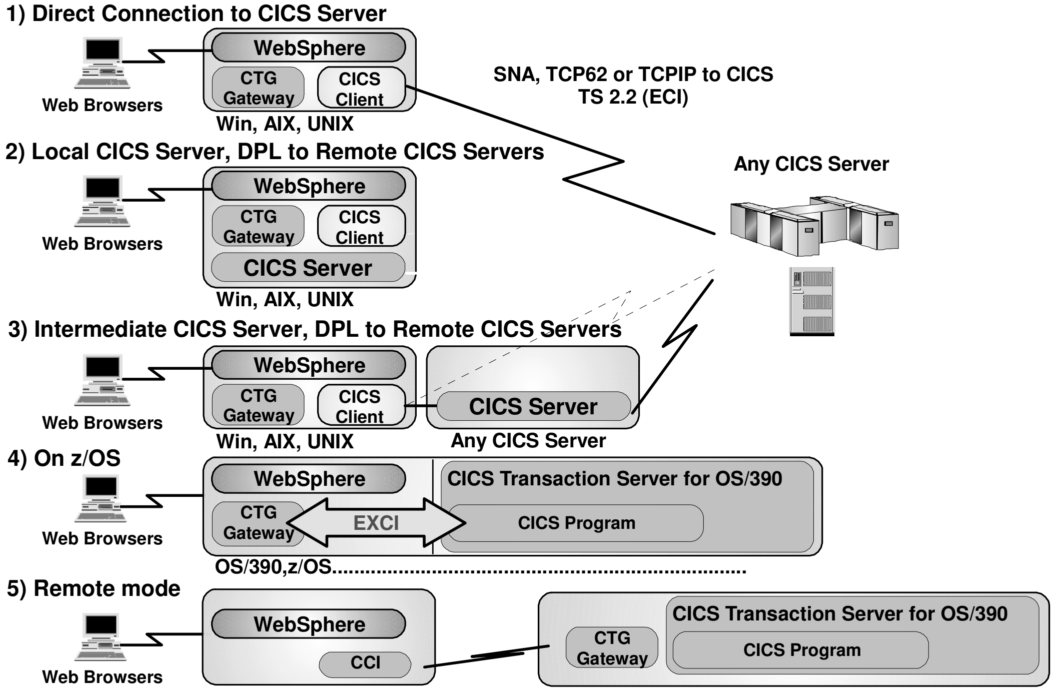
CTG V5.01 highlights

- Strategic connector for WebSphere to CICS
  - works with WebSphere on all platforms
  - CTG 5.0 supported WAS 4.01 for OS/390
  - CTG 5.0.1 supports WAS V5.0 for z/OS
- Supports IBM Adapters & Connectors Strategy and JCA
  - common tooling with other J2EE connectors
  - packaged with WebSphere Studio Application Developer / Integration Edition (for devt.)
  - evolved from CCF to J2C interfaces
- Supercedes previous connectors:
  - CTG 4.02
  - WebSphere/390 EXCI connector (beta with WAS 4.0)
  - server-based use of CUC 3.1.3
- Updated platform support
  - AIX 5.1, Windows XP, zLinux kernel 2.4
- Separately priced package
  - enables standalone sales with any app server
  - same price on distributed & S/390 platforms

CTG V5 functions

- Supports JDK 1.3
  - including JSSE (Java Secure Sockets Extension) for 128-bit encryption
- Supports J2EE Connector Architecture (JCA)
  - ECI & EPI (AIX, Solaris, HP-UX, Windows NT/2000, zLinux)
  - ECI only, 2PC transactions & enhanced security (z/OS)
  - async ECI calls also supported (all platforms)
- Enhanced support for TCP62 (all platforms ex. z/OS)
  - removes SNA dependency for connection to CICS
- Improved performance for ECI data transfers
  - datastreams truncated to application data length
- Improved availability, serviceability, manageability
  - support for ARM (Automatic Restart Manager) on z/OS
  - enhanced logging
    - logging of EXCI return codes on z/OS
    - all messages start with the date
  - dynamic control of tracing level; management infrastructure for JMX (Java Management eXtensions)

CICS Transaction Gateway Configurations



1) Direct Connection to CICS Server

Web Browsers

WebSphere

CTG Gateway

CICS Client

Win, AIX, UNIX

SNA, TCP62 or TCPIP to CICS TS 2.2 (ECI)

Any CICS Server

2) Local CICS Server, DPL to Remote CICS Servers

Web Browsers

WebSphere

CTG Gateway

CICS Client

CICS Server

Win, AIX, UNIX

3) Intermediate CICS Server, DPL to Remote CICS Servers

Web Browsers

WebSphere

CTG Gateway

CICS Client

CICS Server

Win, AIX, UNIX

Any CICS Server

4) On z/OS

Web Browsers

WebSphere

CTG Gateway

EXCI

CICS Transaction Server for OS/390

CICS Program

OS/390,z/OS...

5) Remote mode

Web Browsers

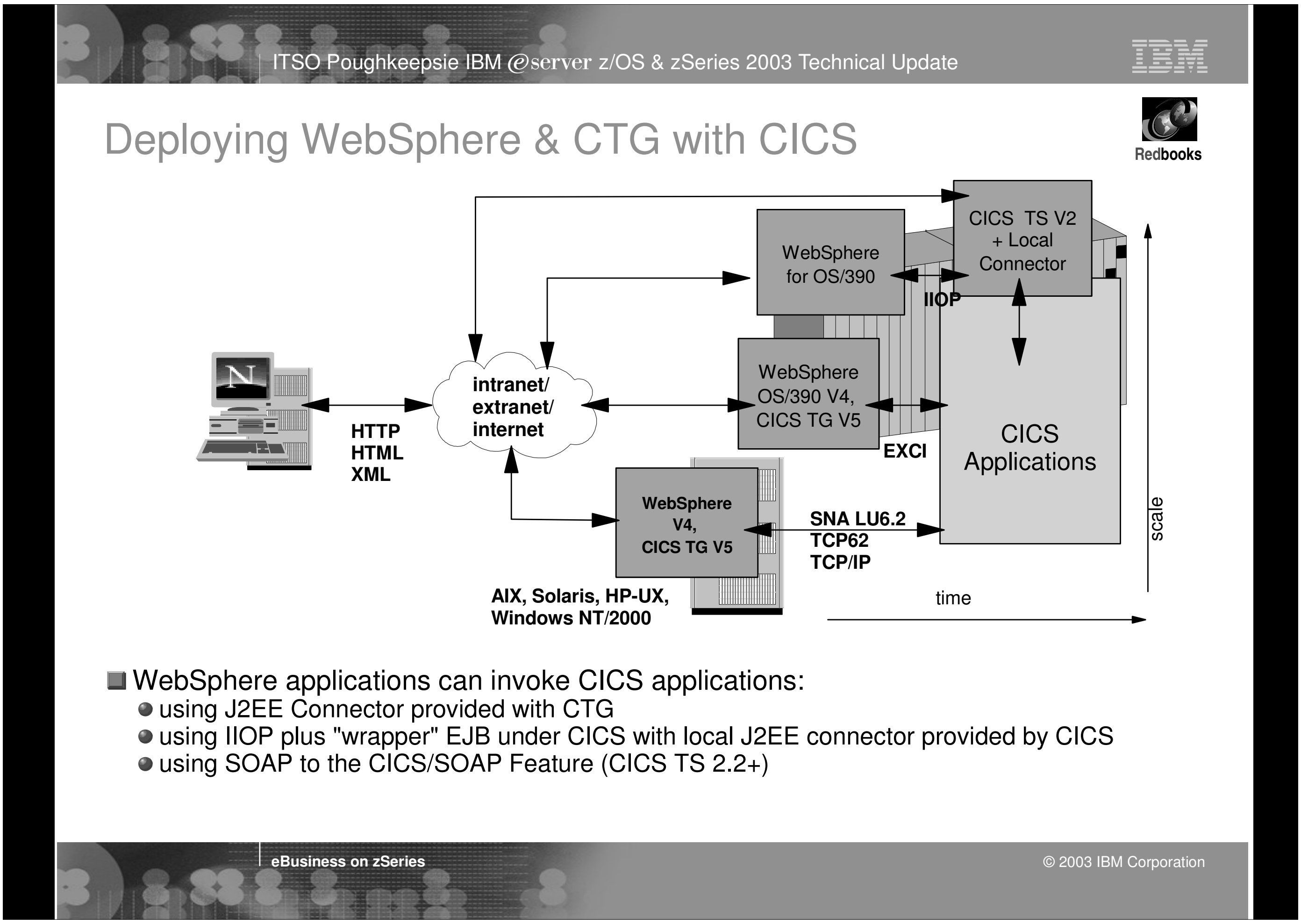
WebSphere

CCI

CTG Gateway

CICS Transaction Server for OS/390

CICS Program



Deploying WebSphere & CTG with CICS

Redbooks

CICS TS V2 + Local Connector

WebSphere for OS/390

IIOP

WebSphere OS/390 V4, CICS TG V5

intranet/ extranet/ internet

HTTP
HTML
XML

EXCI

CICS Applications

WebSphere V4, CICS TG V5

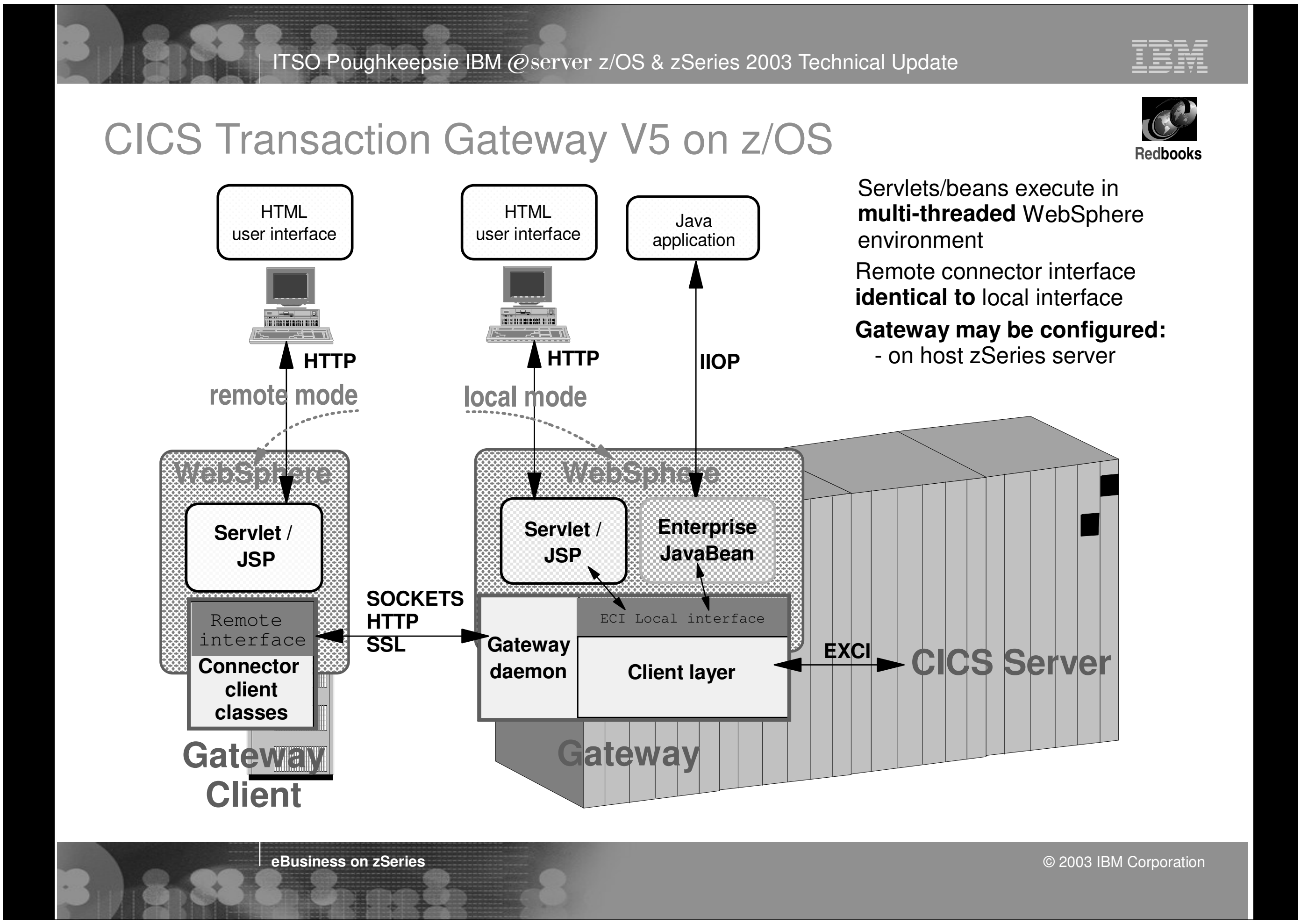
SNA LU6.2
TCP62
TCP/IP

AIX, Solaris, HP-UX, Windows NT/2000

time

scale

- WebSphere applications can invoke CICS applications:
  - using J2EE Connector provided with CTG
  - using IIOP plus "wrapper" EJB under CICS with local J2EE connector provided by CICS
  - using SOAP to the CICS/SOAP Feature (CICS TS 2.2+)



CICS Transaction Gateway V5 on z/OS

Redbooks

HTML user interface

HTML user interface

Java application

Servlets/beans execute in **multi-threaded** WebSphere environment

Remote connector interface **identical to** local interface

Gateway may be configured:
- on host zSeries server

HTTP

HTTP

IIOP

remote mode

local mode

WebSphere

WebSphere

Servlet / JSP

Servlet / JSP

Enterprise JavaBean

SOCKETS
HTTP
SSL

Remote interface

Connector client classes

ECI Local interface

Gateway daemon

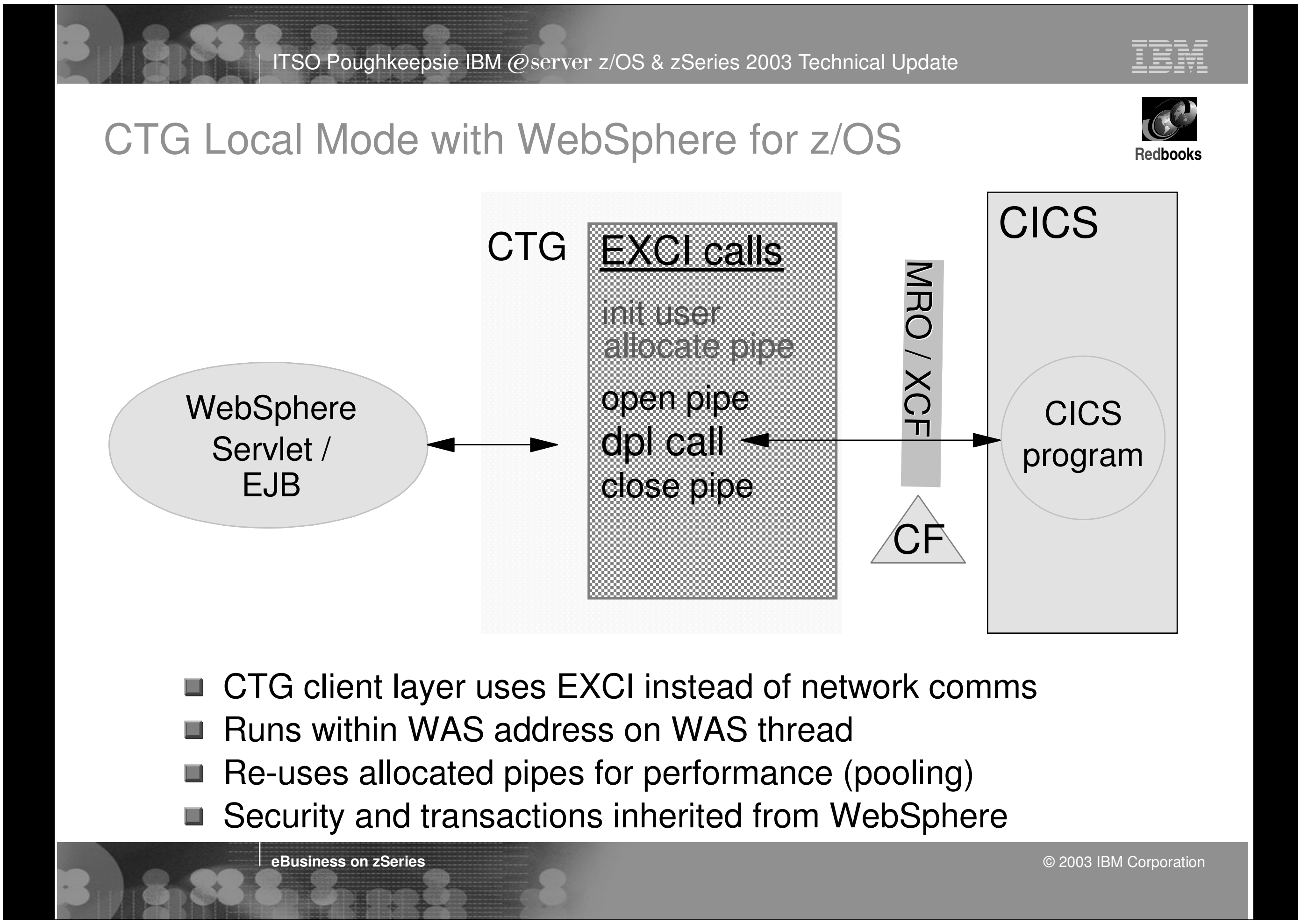
Client layer

EXCI

CICS Server

Gateway Client

Gateway



CTG Local Mode with WebSphere for z/OS

WebSphere Servlet / EJB

CTG

EXCI calls

init user
allocate pipe
open pipe
dpl call
close pipe

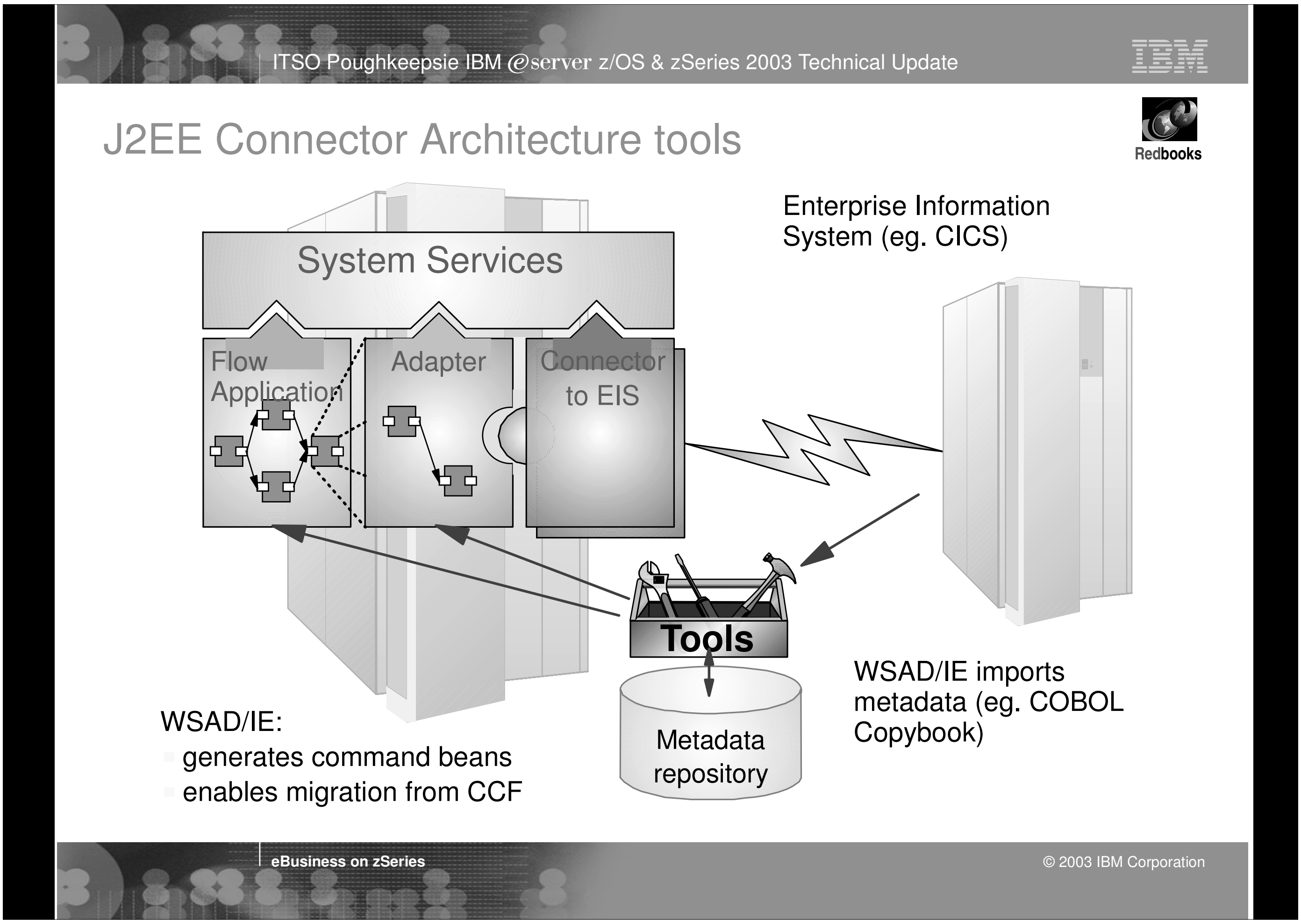
MRO / XCF

CF

CICS

CICS program

- CTG client layer uses EXCI instead of network comms
- Runs within WAS address on WAS thread
- Re-uses allocated pipes for performance (pooling)
- Security and transactions inherited from WebSphere



J2EE Connector Architecture tools

System Services

Flow Application

Adapter

Connector to EIS

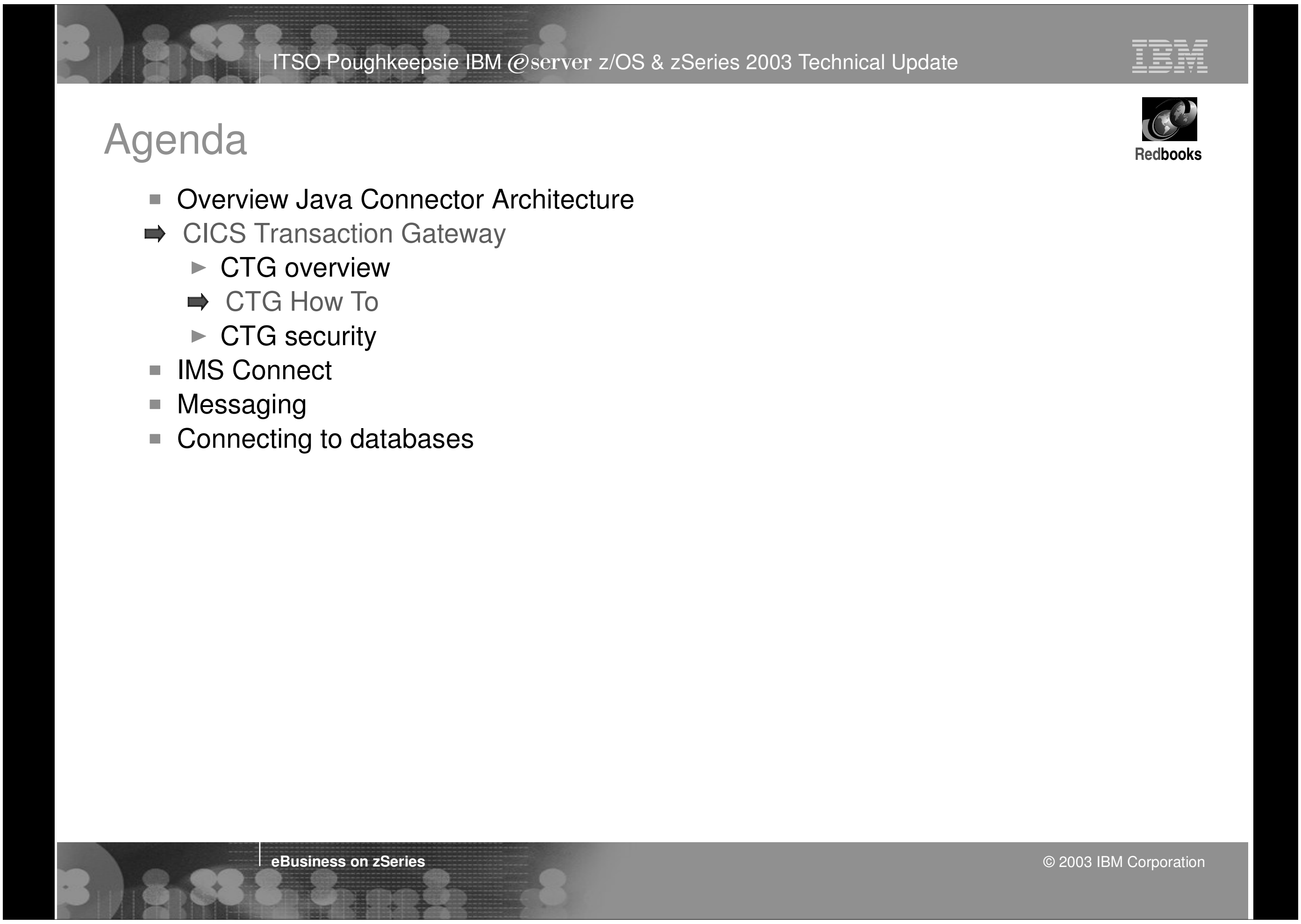
Enterprise Information System (eg. CICS)

Tools

Metadata repository

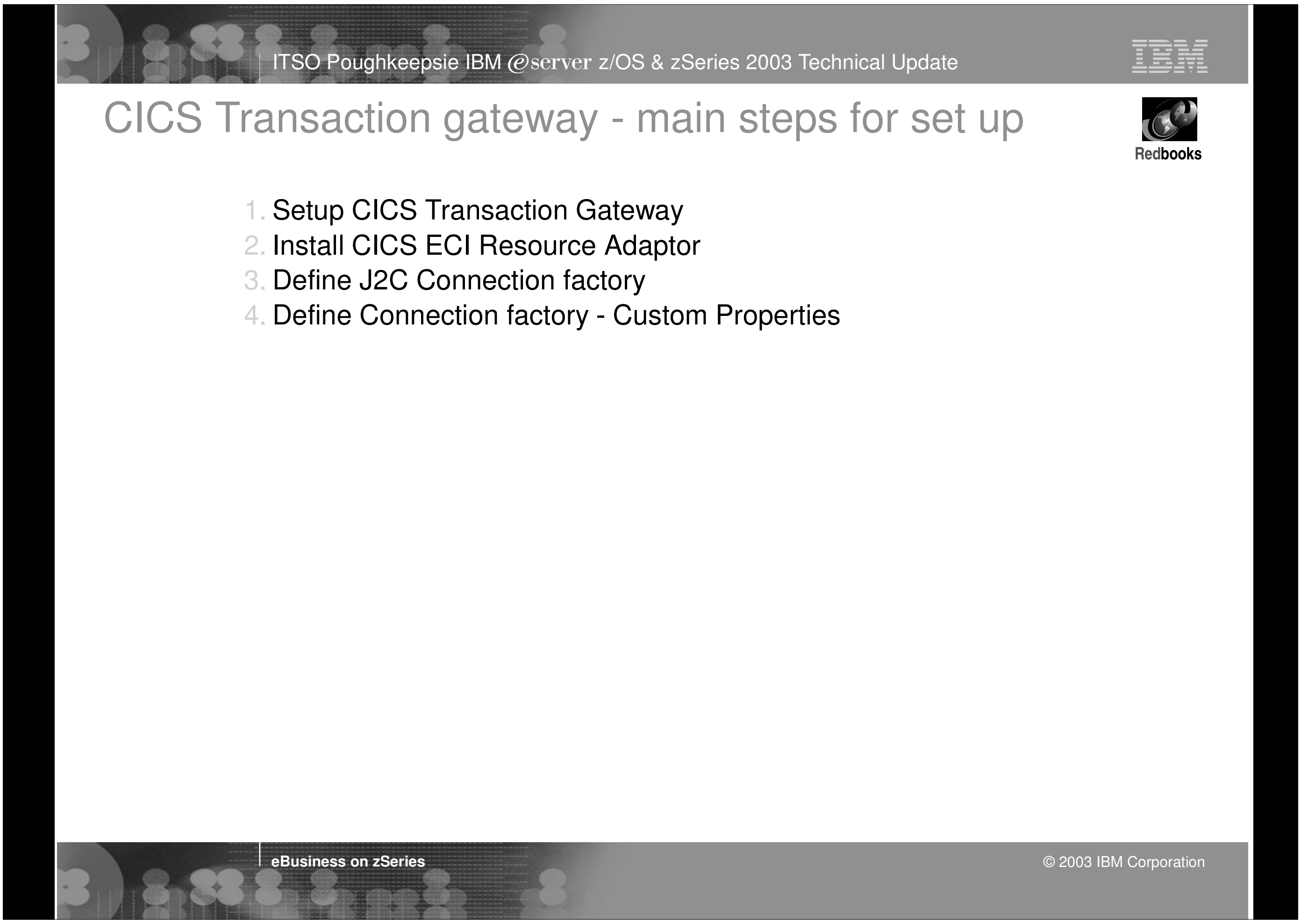
WSAD/IE imports metadata (eg. COBOL Copybook)

WSAD/IE:
- generates command beans
- enables migration from CCF



Agenda

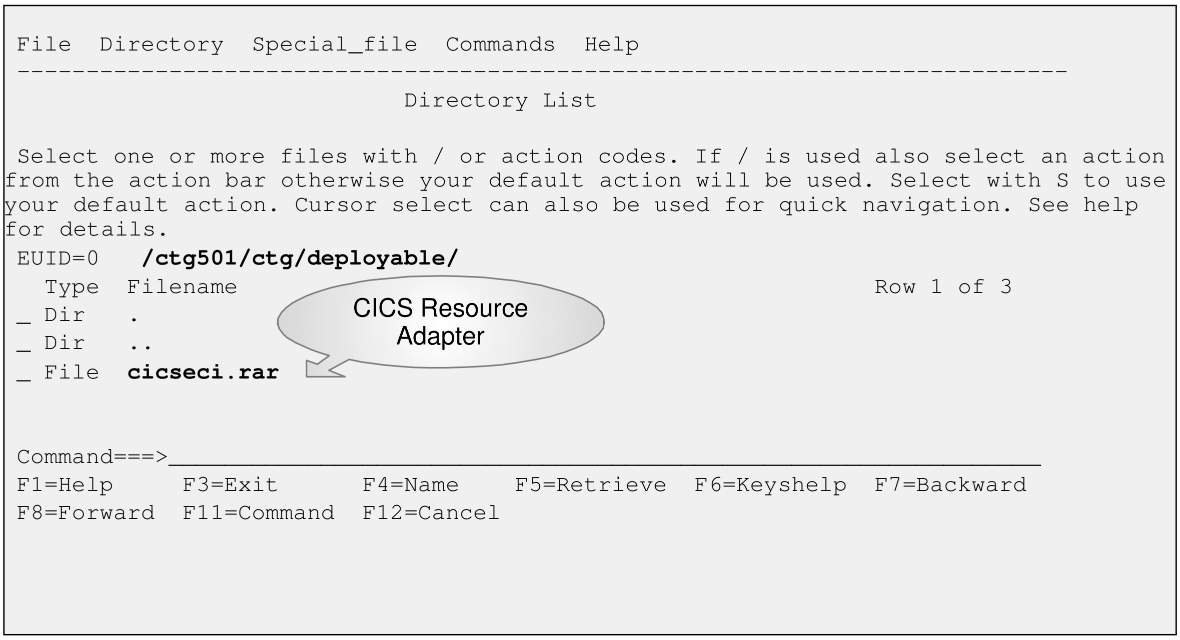
- Overview Java Connector Architecture
- ➡ CICS Transaction Gateway
  - ► CTG overview
  - ➡ CTG How To
  - ► CTG security
- IMS Connect
- Messaging
- Connecting to databases



CICS Transaction gateway - main steps for set up

1. Setup CICS Transaction Gateway
2. Install CICS ECI Resource Adaptor
3. Define J2C Connection factory
4. Define Connection factory - Custom Properties

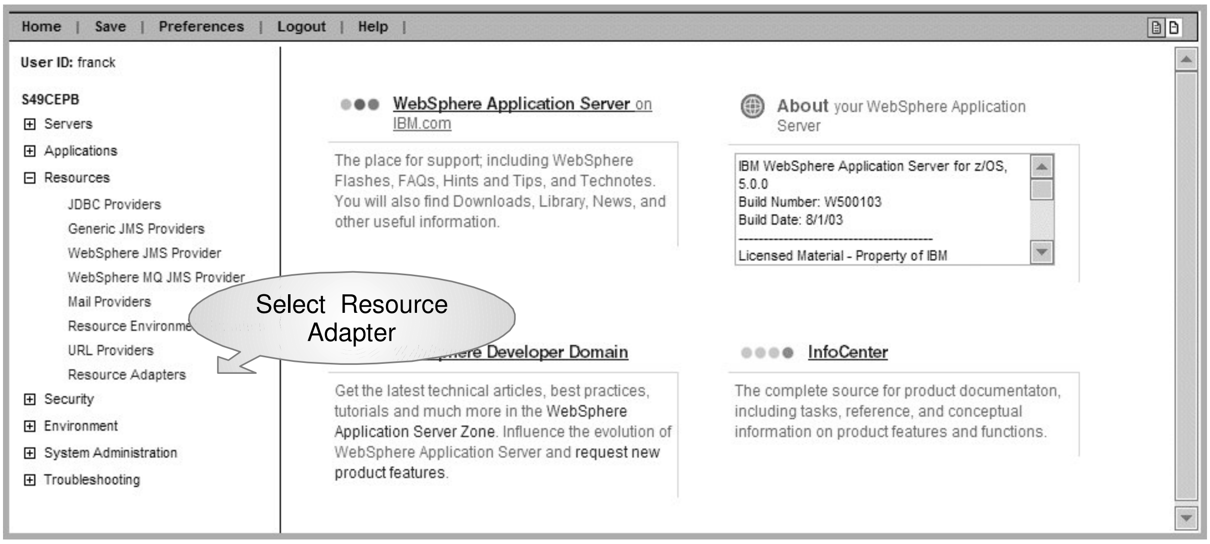
CTG - resource adapter file on z/OS



```
 File  Directory  Special_file  Commands  Help
 ----------------------------------------------------------------------------
                            Directory List

 Select one or more files with / or action codes. If / is used also select an action
from the action bar otherwise your default action will be used. Select with S to use
your default action. Cursor select can also be used for quick navigation. See help
for details.
 EUID=0   /ctg501/ctg/deployable/
   Type  Filename                                                   Row 1 of 3
 _ Dir   .
 _ Dir   ..
 _ File  cicseci.rar


 Command===>_______________________________________________________________
 F1=Help     F3=Exit      F4=Name    F5=Retrieve  F6=Keyshelp  F7=Backward
 F8=Forward  F11=Command  F12=Cancel
```

CICS Resource Adapter

WebSphere V5 console - Resource Adapters



The place for support; including WebSphere Flashes, FAQs, Hints and Tips, and Technotes. You will also find Downloads, Library, News, and other useful information.

IBM WebSphere Application Server for z/OS, 5.0.0
Build Number: W500103
Build Date: 8/1/03

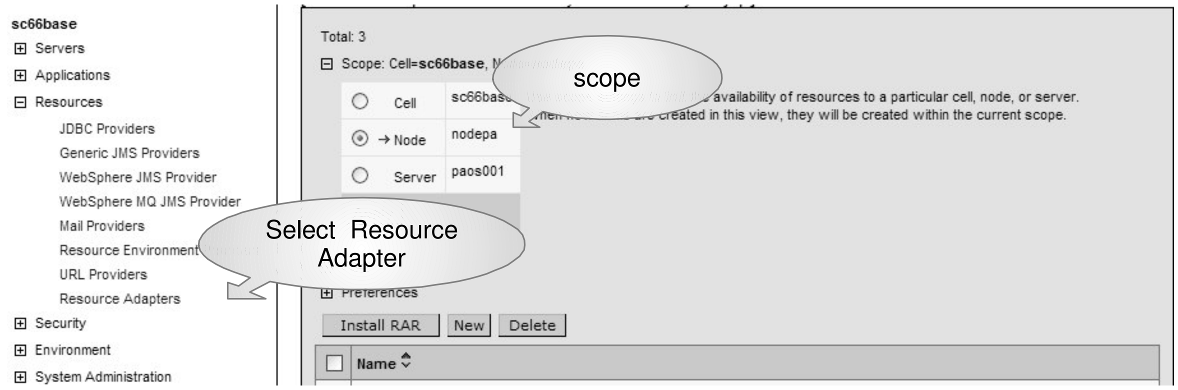
Licensed Material - Property of IBM

Get the latest technical articles, best practices, tutorials and much more in the WebSphere Application Server Zone. Influence the evolution of WebSphere Application Server and request new product features.

The complete source for product documentaton, including tasks, reference, and conceptual information on product features and functions.

Select Resource Adapter

WebSphere V5 console - selecting Resource Adapters



sc66base
⊞ Servers
⊞ Applications
⊟ Resources
JDBC Providers
Generic JMS Providers
WebSphere JMS Provider
WebSphere MQ JMS Provider
Mail Providers
Resource Environment
URL Providers
Resource Adapters
⊞ Security
⊞ Environment
⊞ System Administration

Total: 3

⊟ Scope: Cell=**sc66base**, N

○	Cell	sc66base
●	→ Node	nodepa
○	Server	paos001

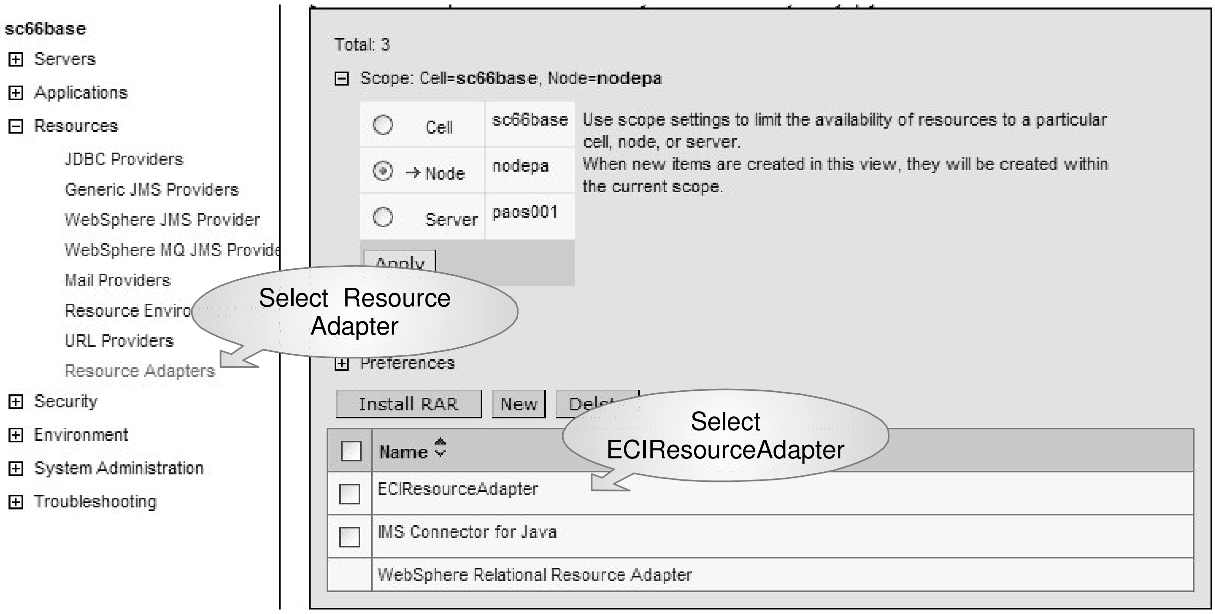
availability of resources to a particular cell, node, or server.
created in this view, they will be created within the current scope.

⊞ Preferences

Install RAR | New | Delete

☐	Name

WebSphere V5 console - selecting an existing RAR file



sc66base
⊞ Servers
⊞ Applications
⊟ Resources
JDBC Providers
Generic JMS Providers
WebSphere JMS Provider
WebSphere MQ JMS Provide
Mail Providers
Resource Enviro
URL Providers
Resource Adapters
⊞ Security
⊞ Environment
⊞ System Administration
⊞ Troubleshooting

Total: 3

⊟ Scope: Cell=**sc66base**, Node=**nodepa**

○	Cell	sc66base
●	→ Node	nodepa
○	Server	paos001

Use scope settings to limit the availability of resources to a particular cell, node, or server.
When new items are created in this view, they will be created within the current scope.

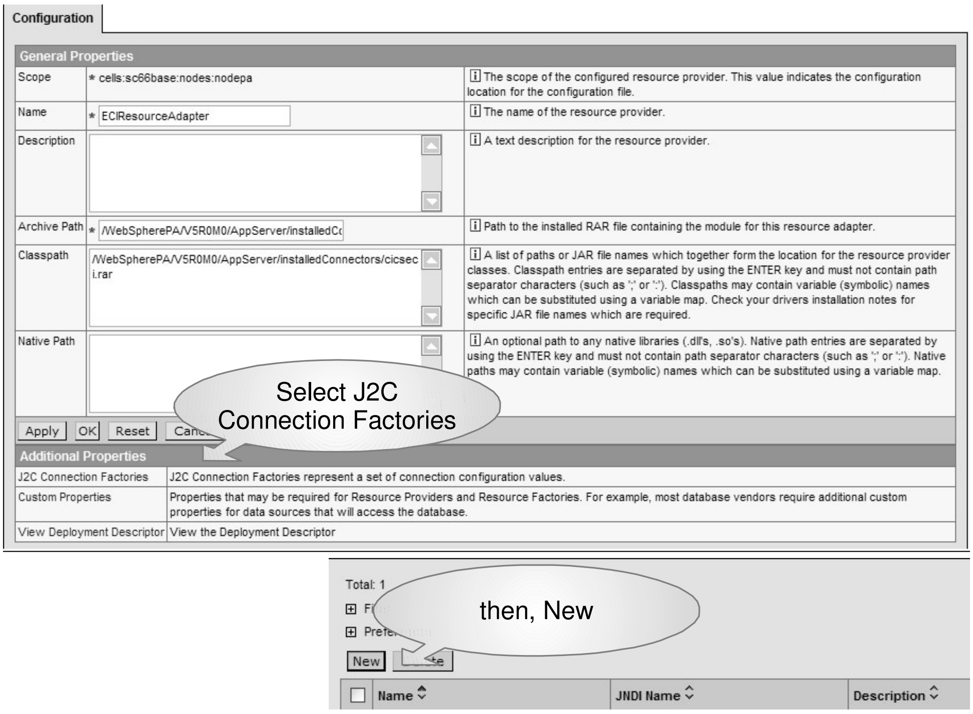
Apply

⊞ Preferences

Install RAR | New | Delet

☐	Name
☐	ECIResourceAdapter
☐	IMS Connector for Java
	WebSphere Relational Resource Adapter

WebSphere V5 console - selecting J2C conn. fact.



Configuration

General Properties		
Scope	* cells:sc66base:nodes:nodepa	The scope of the configured resource provider. This value indicates the configuration location for the configuration file.
Name	* ECIResourceAdapter	The name of the resource provider.
Description		A text description for the resource provider.
Archive Path	* /WebSpherePA/V5R0M0/AppServer/installedC	Path to the installed RAR file containing the module for this resource adapter.
Classpath	/WebSpherePA/V5R0M0/AppServer/installedConnectors/cicseci.rar	A list of paths or JAR file names which together form the location for the resource provider classes. Classpath entries are separated by using the ENTER key and must not contain path separator characters (such as ';' or ':'). Classpaths may contain variable (symbolic) names which can be substituted using a variable map. Check your drivers installation notes for specific JAR file names which are required.
Native Path		An optional path to any native libraries (.dll's, .so's). Native path entries are separated by using the ENTER key and must not contain path separator characters (such as ';' or ':'). Native paths may contain variable (symbolic) names which can be substituted using a variable map.

Additional Properties	
J2C Connection Factories	J2C Connection Factories represent a set of connection configuration values.
Custom Properties	Properties that may be required for Resource Providers and Resource Factories. For example, most database vendors require additional custom properties for data sources that will access the database.
View Deployment Descriptor	View the Deployment Descriptor

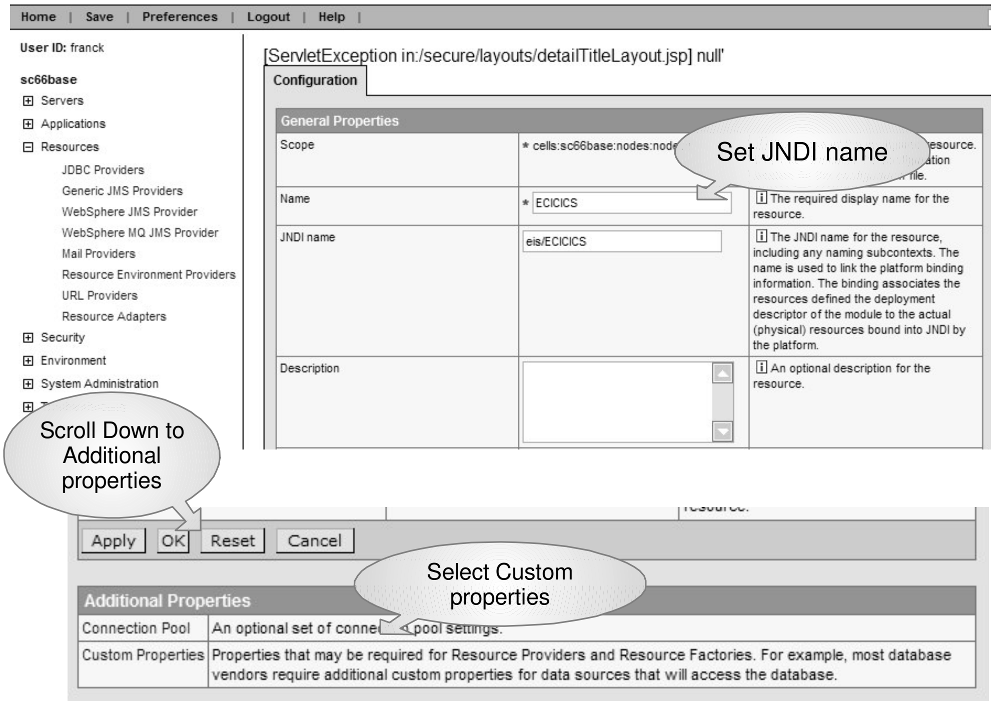
Select J2C Connection Factories

Total: 1

□	Name	JNDI Name	Description

then, New

WebSphere V5 console - JNDI name and add. prop.



User ID: franck

[ServletException in:/secure/layouts/detailTitleLayout.jsp] null'

Configuration

General Properties		
Scope	* cells:sc66base:nodes:node	esource. ation file.
Name	* ECICICS	The required display name for the resource.
JNDI name	eis/ECICICS	The JNDI name for the resource, including any naming subcontexts. The name is used to link the platform binding information. The binding associates the resources defined the deployment descriptor of the module to the actual (physical) resources bound into JNDI by the platform.
Description		An optional description for the resource.

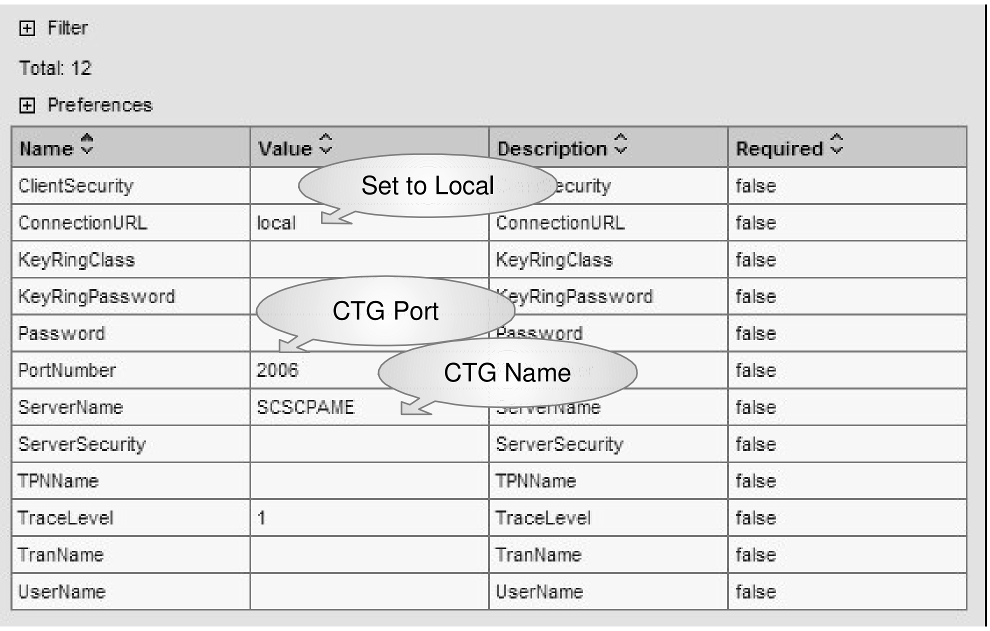
Set JNDI name

Scroll Down to Additional properties

Additional Properties	
Connection Pool	An optional set of conne pool settings.
Custom Properties	Properties that may be required for Resource Providers and Resource Factories. For example, most database vendors require additional custom properties for data sources that will access the database.

Select Custom properties

WebSphere V5 console - properties for CTG



⊞ Filter

Total: 12

⊞ Preferences

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

Apply, Save.

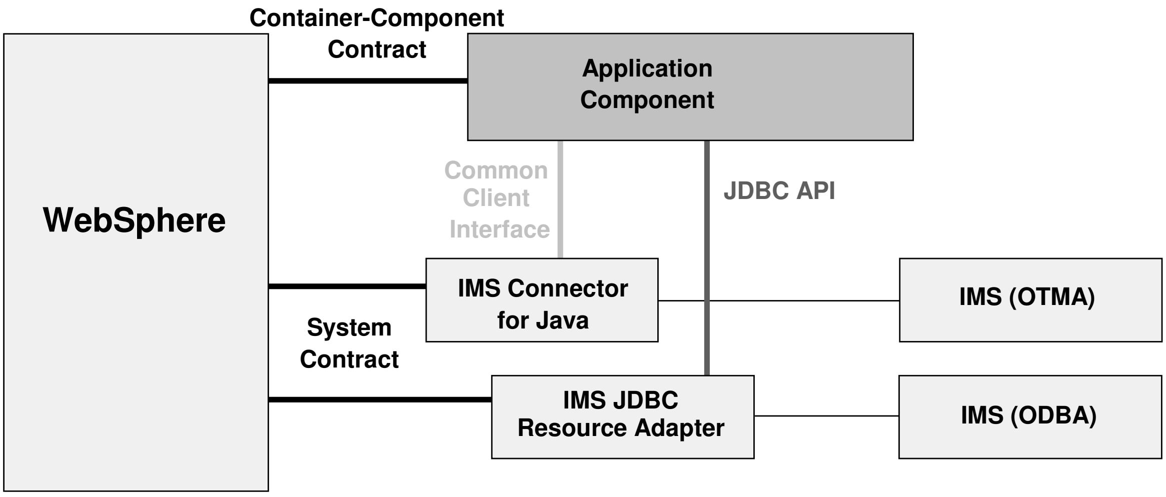
Agenda

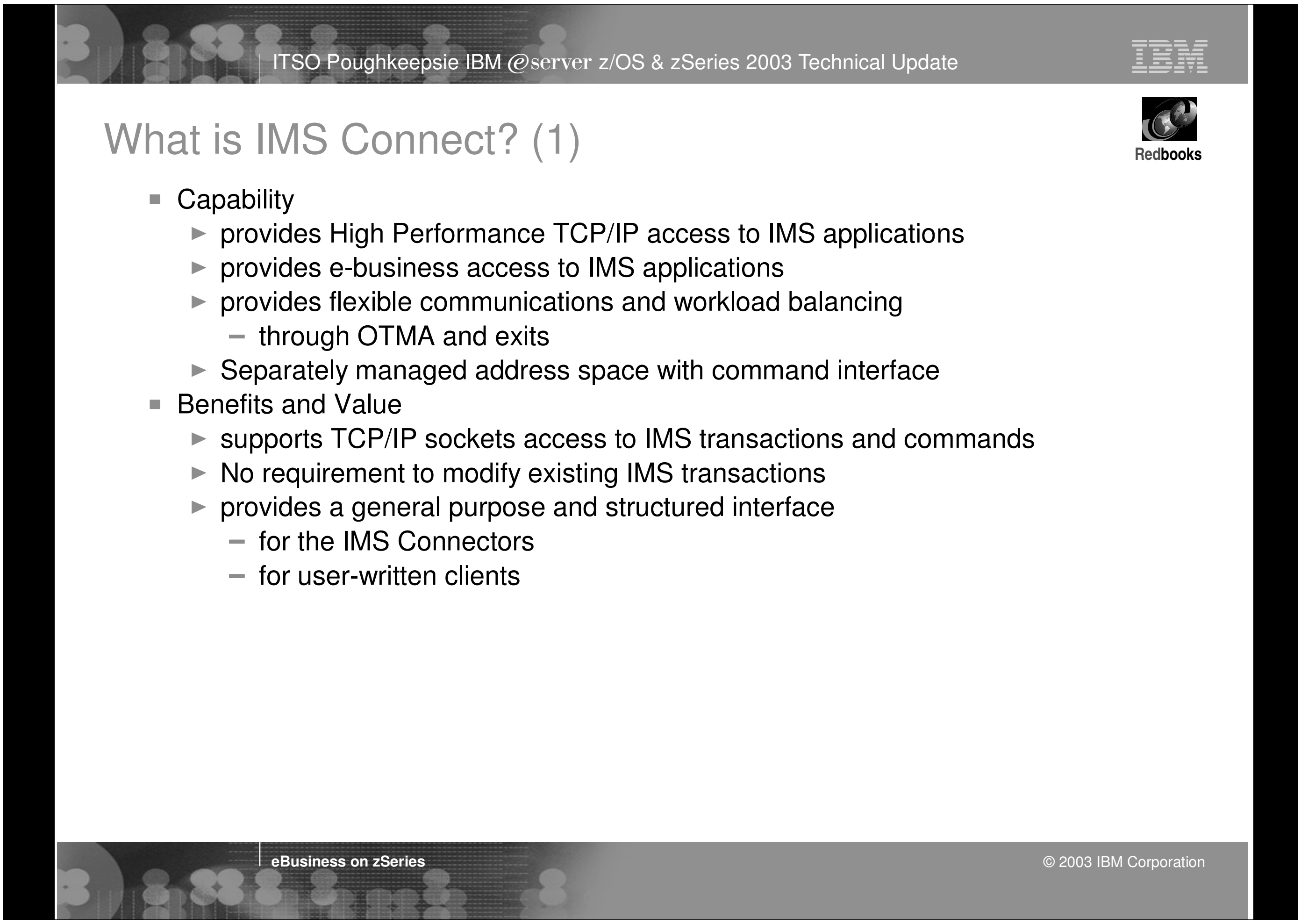
- Overview Java Connector Architecture
- CICS Transaction Gateway
- ➡ IMS Connect
  - ➡ IMS Connect overview
  - IMS Connect How To
- Messaging
- Connecting to databases

WebSphere/IMS integration - scenarios

- Continue with App development in IMS dependent regions and wrap with EJB session beans
  - Pro - fast Web deployment of existing apps, easiest management, distributed transactions not really needed, Java development for new applications
  - Con - slower development of traditional IMS apps due to aging skills, less tooling
- Deploy new apps as EJBs and integrate with existing (perhaps multiple) IMS applications
  - Pro - rapid App development due to language and tool support, can take advantage of new technologies (Workflow, SOAP), portable (write once, run anywhere)
  - Con - more complex because business logic in multiple places, may require distributed transactions, EJBs are a fairly recent, and still evolving, technology
- Deploy new apps as EJBs to access IMS data directly
  - Pro - business logic in single source, Java development can address skills shortage
  - Con - rewriting applications costly, SQL tooling limited by IMS DB capabilities, IMS and WebSphere must be on same image

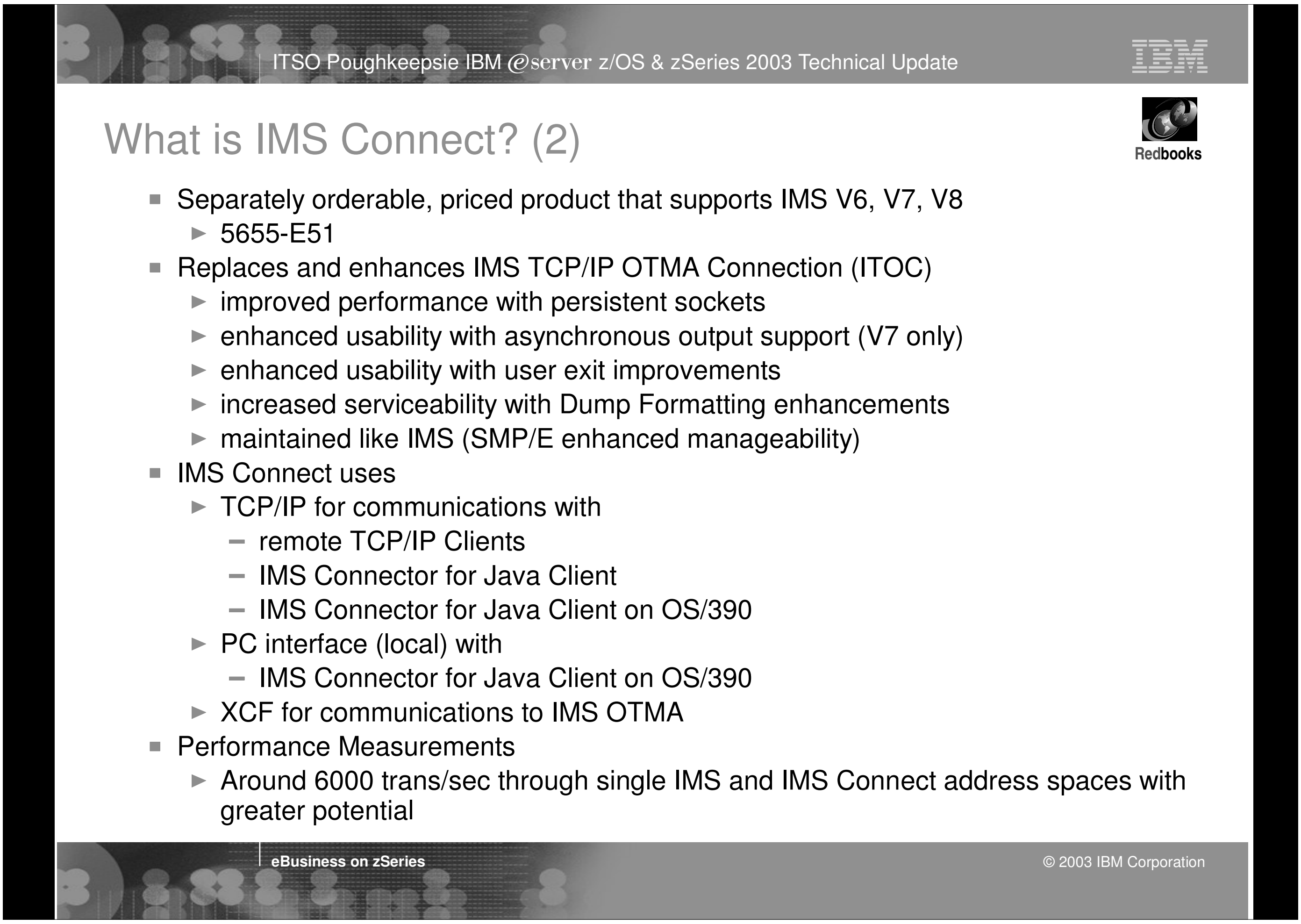
IMS J2EE Connectors





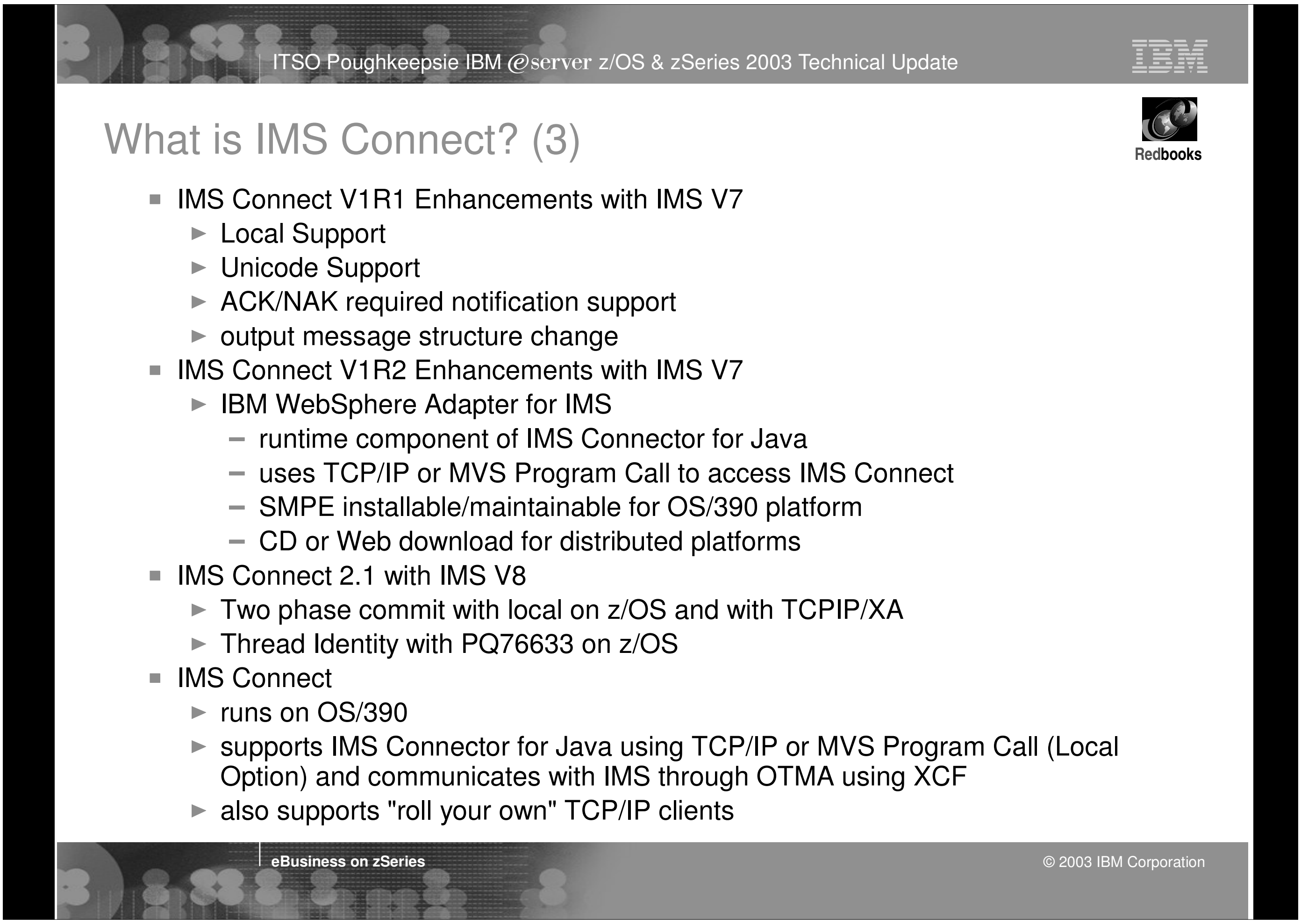
What is IMS Connect? (1)

- Capability
  - provides High Performance TCP/IP access to IMS applications
  - provides e-business access to IMS applications
  - provides flexible communications and workload balancing
    - through OTMA and exits
  - Separately managed address space with command interface
- Benefits and Value
  - supports TCP/IP sockets access to IMS transactions and commands
  - No requirement to modify existing IMS transactions
  - provides a general purpose and structured interface
    - for the IMS Connectors
    - for user-written clients



What is IMS Connect? (2)

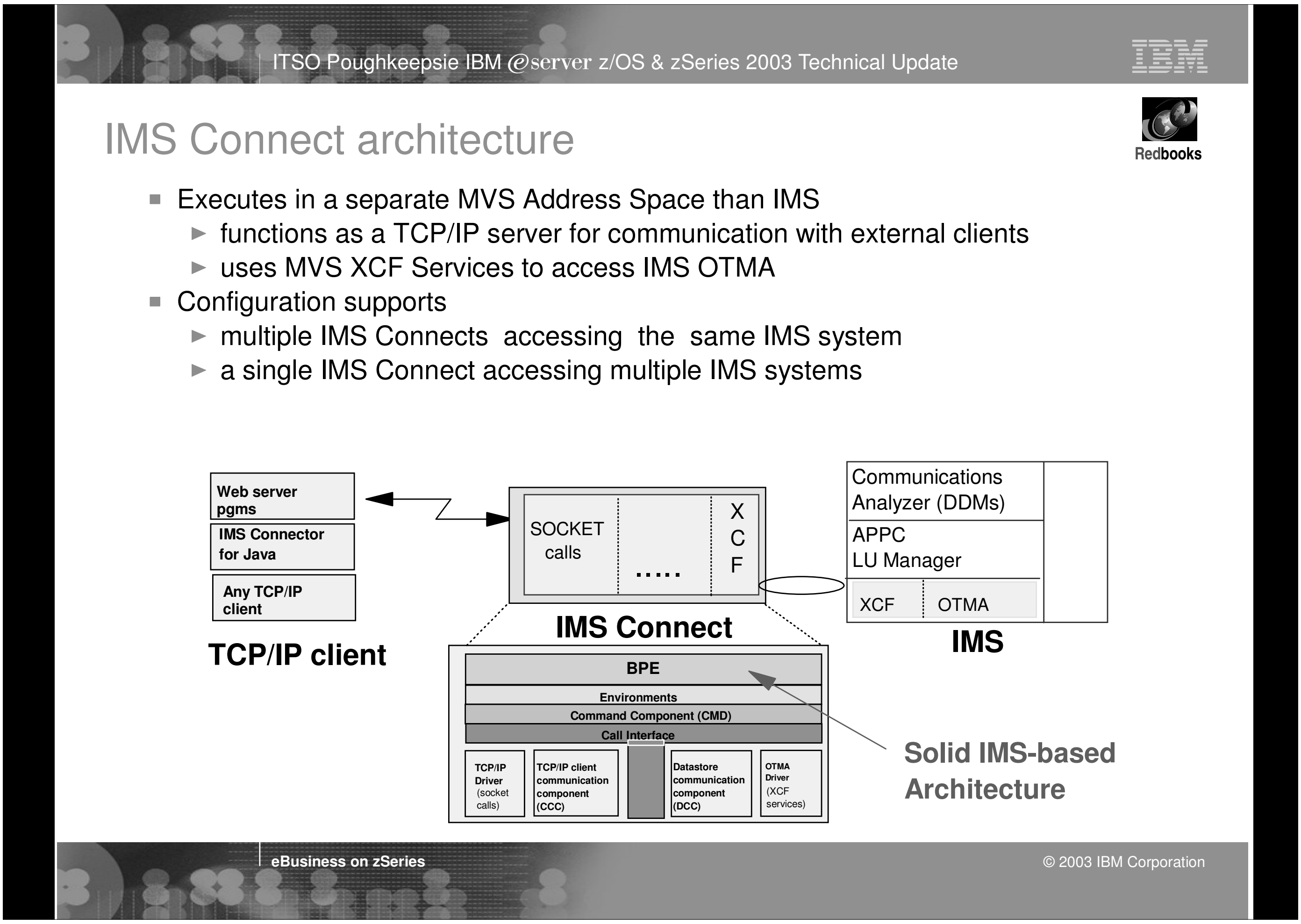
- Separately orderable, priced product that supports IMS V6, V7, V8
  - 5655-E51
- Replaces and enhances IMS TCP/IP OTMA Connection (ITOC)
  - improved performance with persistent sockets
  - enhanced usability with asynchronous output support (V7 only)
  - enhanced usability with user exit improvements
  - increased serviceability with Dump Formatting enhancements
  - maintained like IMS (SMP/E enhanced manageability)
- IMS Connect uses
  - TCP/IP for communications with
    - remote TCP/IP Clients
    - IMS Connector for Java Client
    - IMS Connector for Java Client on OS/390
  - PC interface (local) with
    - IMS Connector for Java Client on OS/390
  - XCF for communications to IMS OTMA
- Performance Measurements
  - Around 6000 trans/sec through single IMS and IMS Connect address spaces with greater potential



What is IMS Connect? (3)

Redbooks

- IMS Connect V1R1 Enhancements with IMS V7
  - Local Support
  - Unicode Support
  - ACK/NAK required notification support
  - output message structure change
- IMS Connect V1R2 Enhancements with IMS V7
  - IBM WebSphere Adapter for IMS
    - runtime component of IMS Connector for Java
    - uses TCP/IP or MVS Program Call to access IMS Connect
    - SMPE installable/maintainable for OS/390 platform
    - CD or Web download for distributed platforms
- IMS Connect 2.1 with IMS V8
  - Two phase commit with local on z/OS and with TCPIP/XA
  - Thread Identity with PQ76633 on z/OS
- IMS Connect
  - runs on OS/390
  - supports IMS Connector for Java using TCP/IP or MVS Program Call (Local Option) and communicates with IMS through OTMA using XCF
  - also supports "roll your own" TCP/IP clients



IMS Connect architecture

Redbooks

- Executes in a separate MVS Address Space than IMS
  - functions as a TCP/IP server for communication with external clients
  - uses MVS XCF Services to access IMS OTMA
- Configuration supports
  - multiple IMS Connects accessing the same IMS system
  - a single IMS Connect accessing multiple IMS systems

Web server pgms

IMS Connector for Java

Any TCP/IP client

TCP/IP client

SOCKET calls

.....

XCF

IMS Connect

BPE

Environments

Command Component (CMD)

Call Interface

TCP/IP Driver (socket calls)

TCP/IP client communication component (CCC)

Datastore communication component (DCC)

OTMA Driver (XCF services)

Communications Analyzer (DDMs)

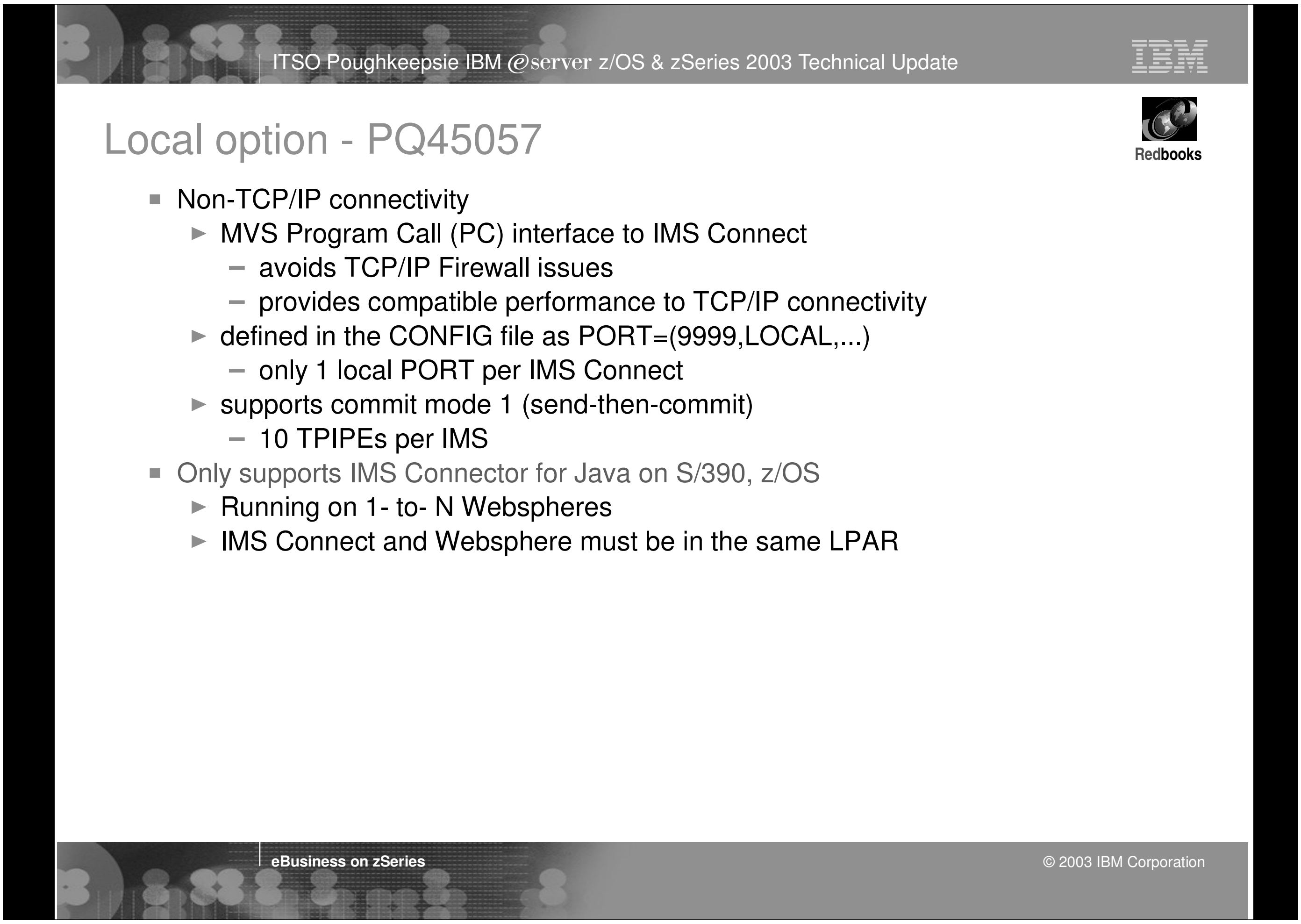
APPC LU Manager

XCF

OTMA

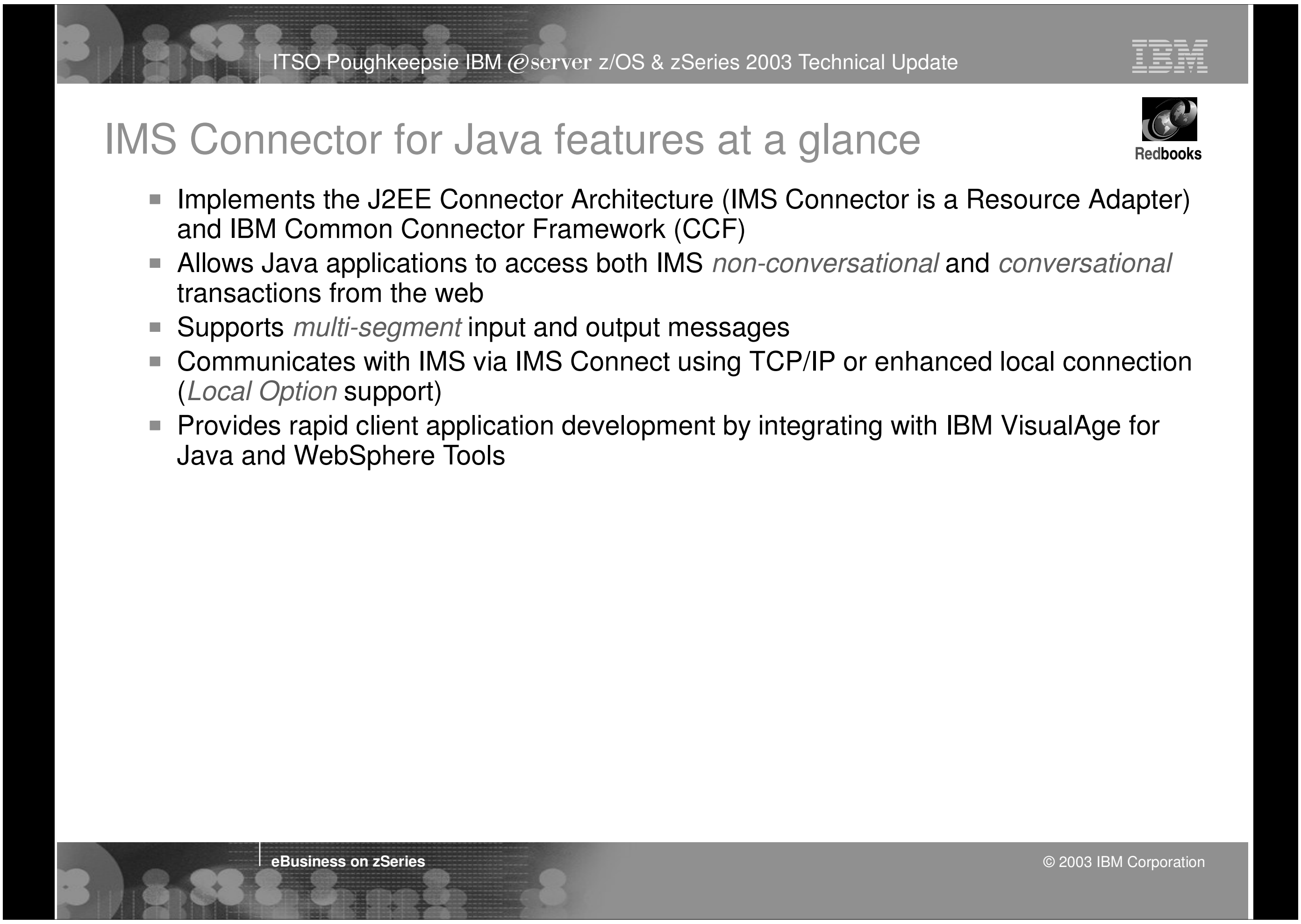
IMS

Solid IMS-based Architecture



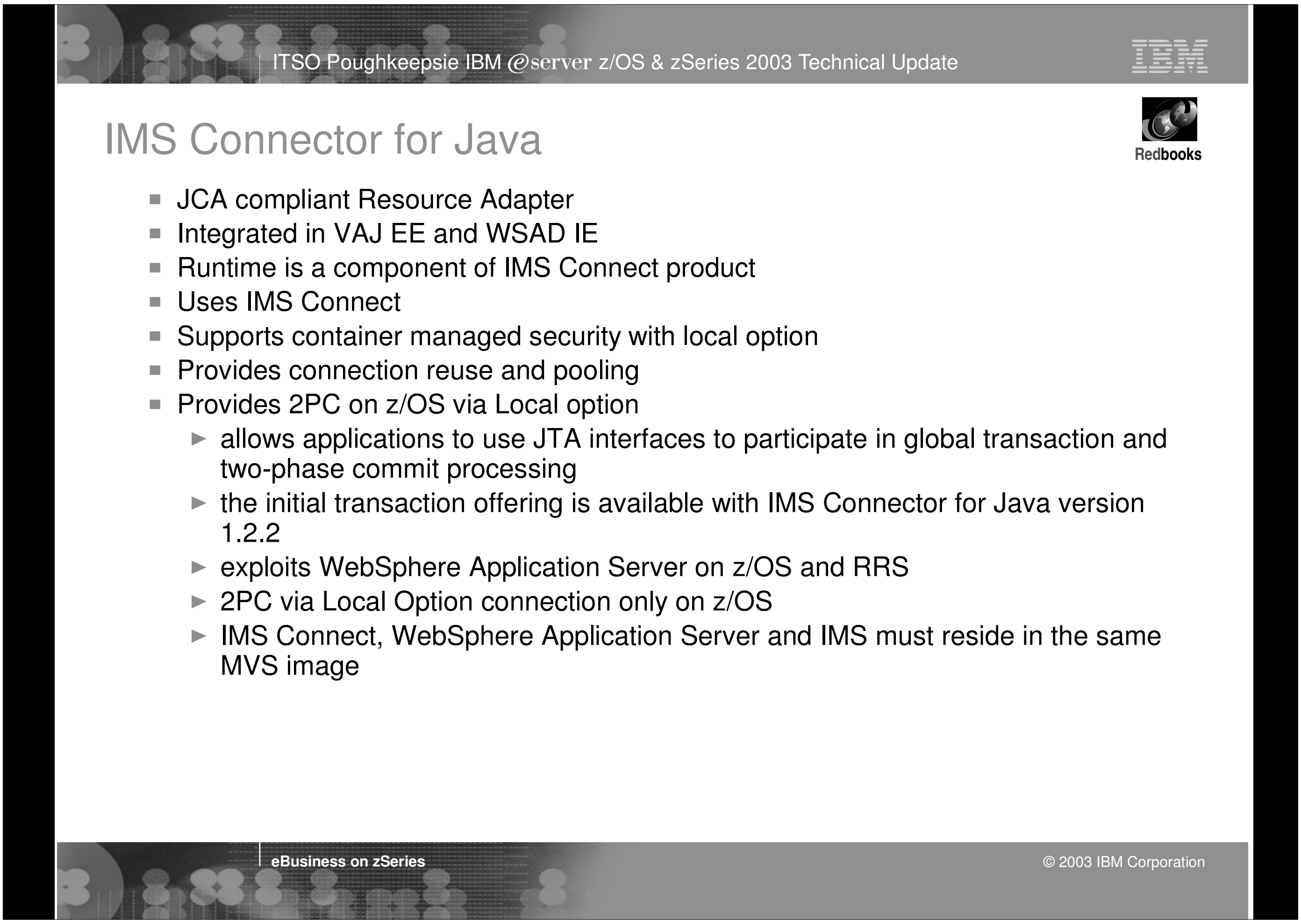
Local option - PQ45057

- Non-TCP/IP connectivity
  - MVS Program Call (PC) interface to IMS Connect
    - avoids TCP/IP Firewall issues
    - provides compatible performance to TCP/IP connectivity
  - defined in the CONFIG file as PORT=(9999,LOCAL,...)
    - only 1 local PORT per IMS Connect
  - supports commit mode 1 (send-then-commit)
    - 10 TPIPEs per IMS
- Only supports IMS Connector for Java on S/390, z/OS
  - Running on 1- to- N Webspheres
  - IMS Connect and Websphere must be in the same LPAR



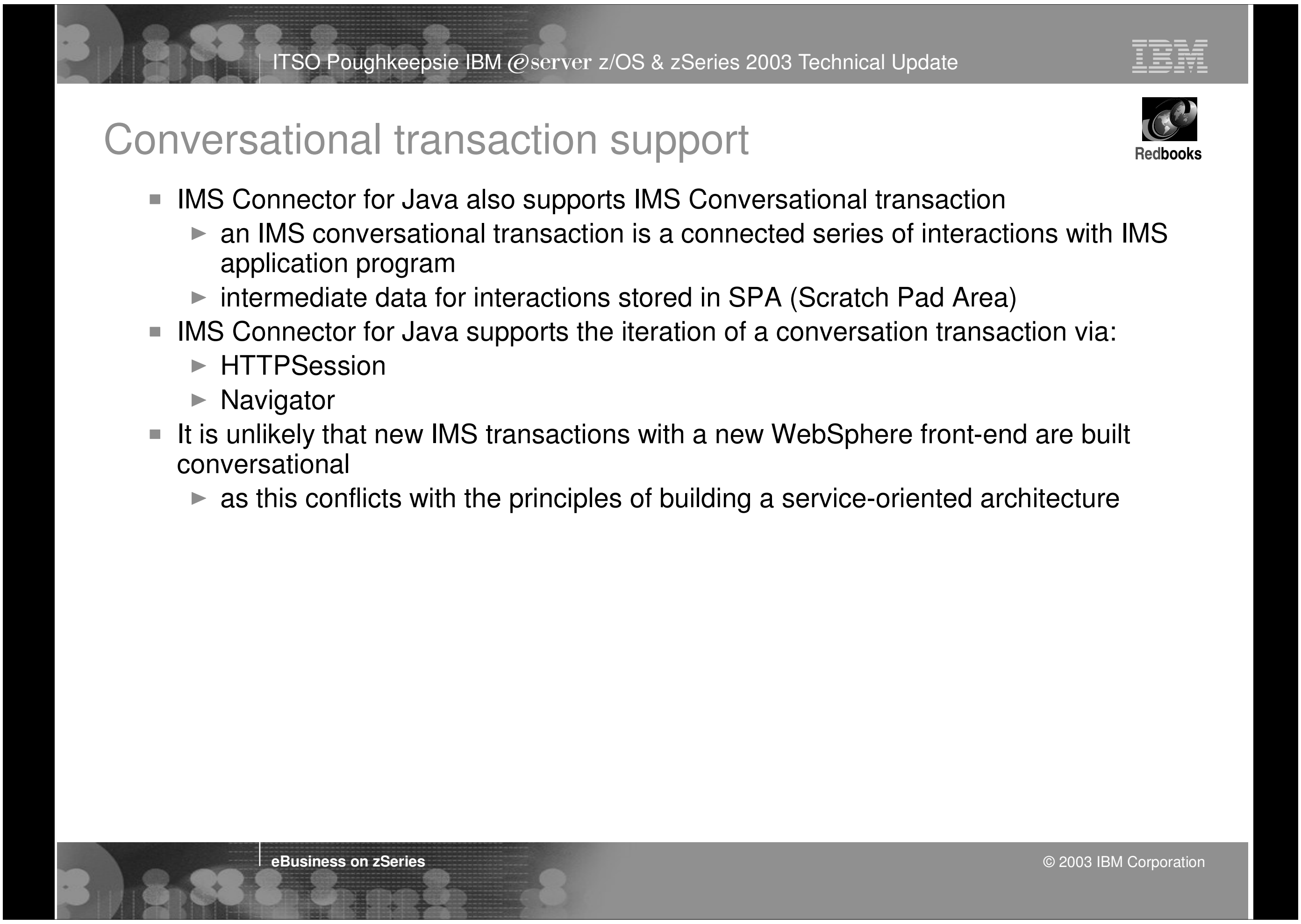
IMS Connector for Java features at a glance

- Implements the J2EE Connector Architecture (IMS Connector is a Resource Adapter) and IBM Common Connector Framework (CCF)
- Allows Java applications to access both IMS *non-conversational* and *conversational* transactions from the web
- Supports *multi-segment* input and output messages
- Communicates with IMS via IMS Connect using TCP/IP or enhanced local connection (*Local Option* support)
- Provides rapid client application development by integrating with IBM VisualAge for Java and WebSphere Tools



IMS Connector for Java

- JCA compliant Resource Adapter
- Integrated in VAJ EE and WSAD IE
- Runtime is a component of IMS Connect product
- Uses IMS Connect
- Supports container managed security with local option
- Provides connection reuse and pooling
- Provides 2PC on z/OS via Local option
  - allows applications to use JTA interfaces to participate in global transaction and two-phase commit processing
  - the initial transaction offering is available with IMS Connector for Java version 1.2.2
  - exploits WebSphere Application Server on z/OS and RRS
  - 2PC via Local Option connection only on z/OS
  - IMS Connect, WebSphere Application Server and IMS must reside in the same MVS image

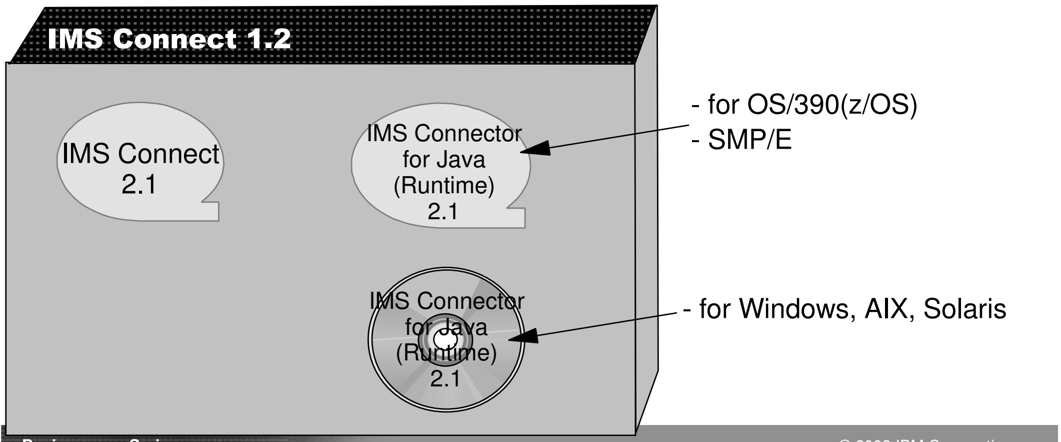


Conversational transaction support

- IMS Connector for Java also supports IMS Conversational transaction
  - an IMS conversational transaction is a connected series of interactions with IMS application program
  - intermediate data for interactions stored in SPA (Scratch Pad Area)
- IMS Connector for Java supports the iteration of a conversation transaction via:
  - HTTPSession
  - Navigator
- It is unlikely that new IMS transactions with a new WebSphere front-end are built conversational
  - as this conflicts with the principles of building a service-oriented architecture

IMS Connector for Java - Packaging

- IMS Connector for Java shipped in two pieces:
  - Runtime - Available as a component of IMS Connect
  - Development - Available with WSAD 5.0.1 (update via WebSphere Studio Developer's Domain (WSDD))
- IMS Connector for Java shares the same version number with IMS Connect (V1.2.x onwards)


IMS Connect 1.2
IMS Connect 2.1
IMS Connector for Java (Runtime) 2.1
IMS Connector for Java (Runtime) 2.1
- for OS/390(z/OS)
- SMP/E
- for Windows, AIX, Solaris

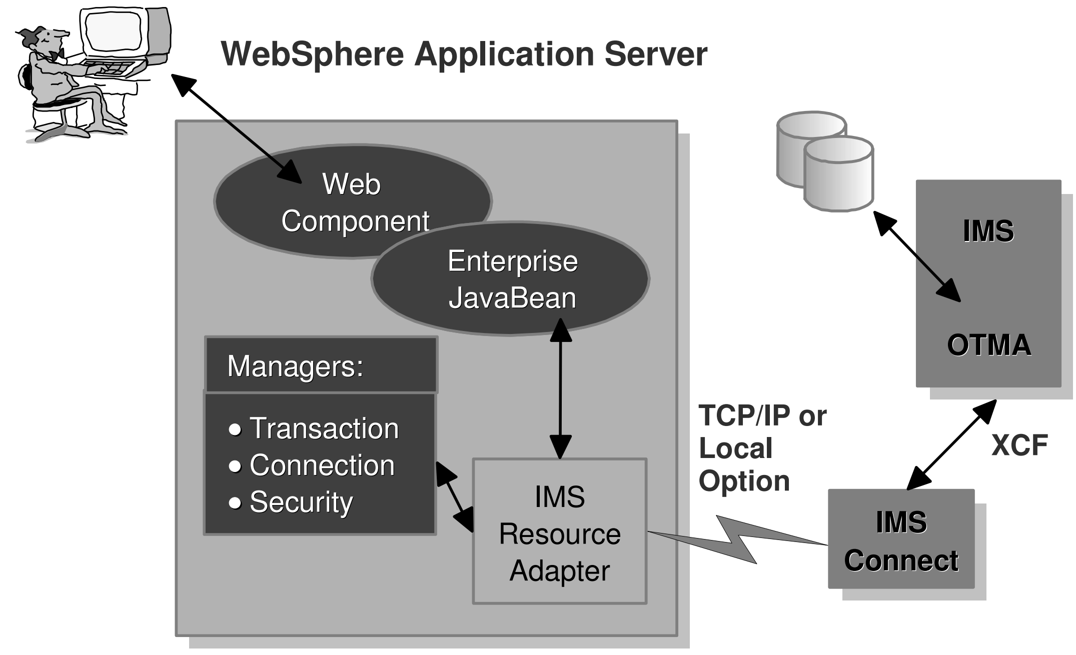
IMS Connector for Java - WAS V4 Versions

- IMS Connector for Java 4.0.1
  - CCF support only
  - required IMS V7 if Local Option is used (i.e. cCan use IMS V6 with TCP/IP)
- IMS Connector for Java 1.2.0
  - providing J2EE Connector Architecture support
  - Prereqs:
    - IMS Connect 1.2
    - IMS V7 if JCA support or Local Option used
  - CCF Connector still supported
    - provided by fixpacks for VisualAge for Java
- IMS Connector for Java 1.2.1
  - NLS-enabled version of IMS Connector for Java Version 1.2.0
- IMS Connector for Java 1.2.2
  - providing J2EE Connector Architecture support and two phase commit support for WAS on z/OS
  - Prereqs
    - IMS Connect 1.2
    - IMS V7 if JCA, Local Option or TPC/zOS used
  - CCF Connector still supported
    - provided by fixpacks for VAJ
- IMS Connector for Java 1.2.3
  - NLS-enabled version of 1.2.2
- IMS Connector for Java 1.2.4
  - also named IMS Resource Adapter
  - integrated with Win WSAD-IE 4.1
    - WebSphere Studio Application Developer Integration Edition
- IMS Connector for Java 1.2.5
  - NLS-enabled version of 1.2.4

IMS Connector for Java WAS V5 version

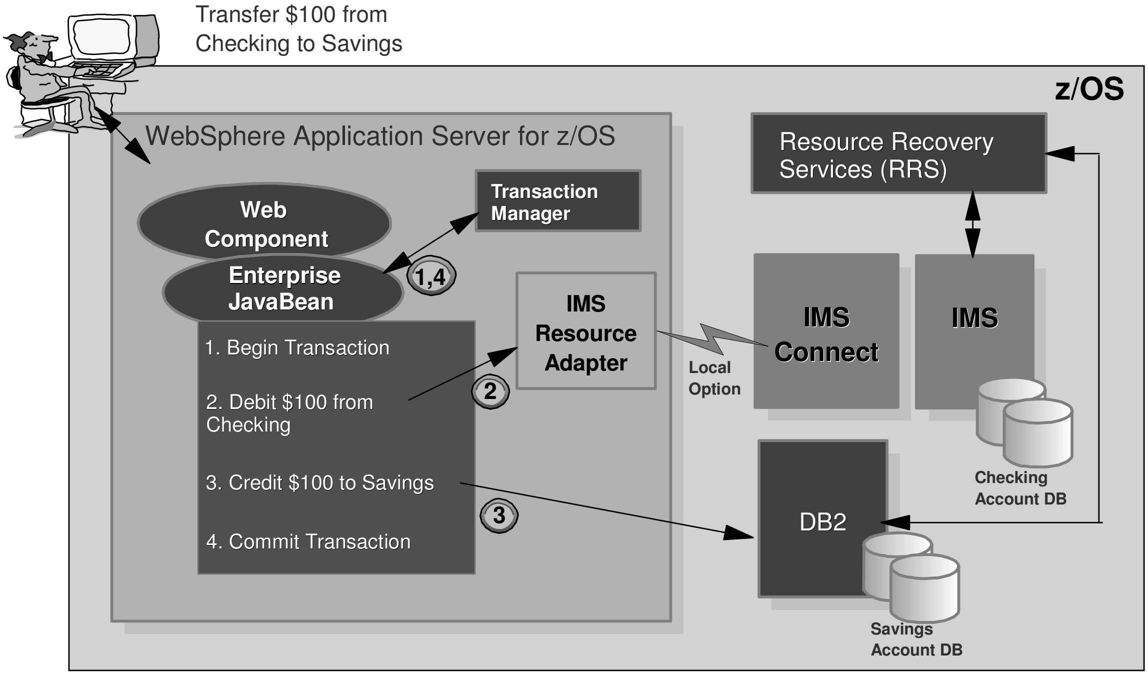
- IMS Connector for Java 2.1
- Requires WSAD IE 5.0.1
- IMS V8

IMS Resource Adapter = Java Connector



Redbooks

Running a transaction



Transfer $100 from Checking to Savings

z/OS

WebSphere Application Server for z/OS

Web Component

Enterprise JavaBean

Transaction Manager

1,4

IMS Resource Adapter

2

3

1. Begin Transaction

2. Debit $100 from Checking

3. Credit $100 to Savings

4. Commit Transaction

Resource Recovery Services (RRS)

IMS Connect

IMS

Local Option

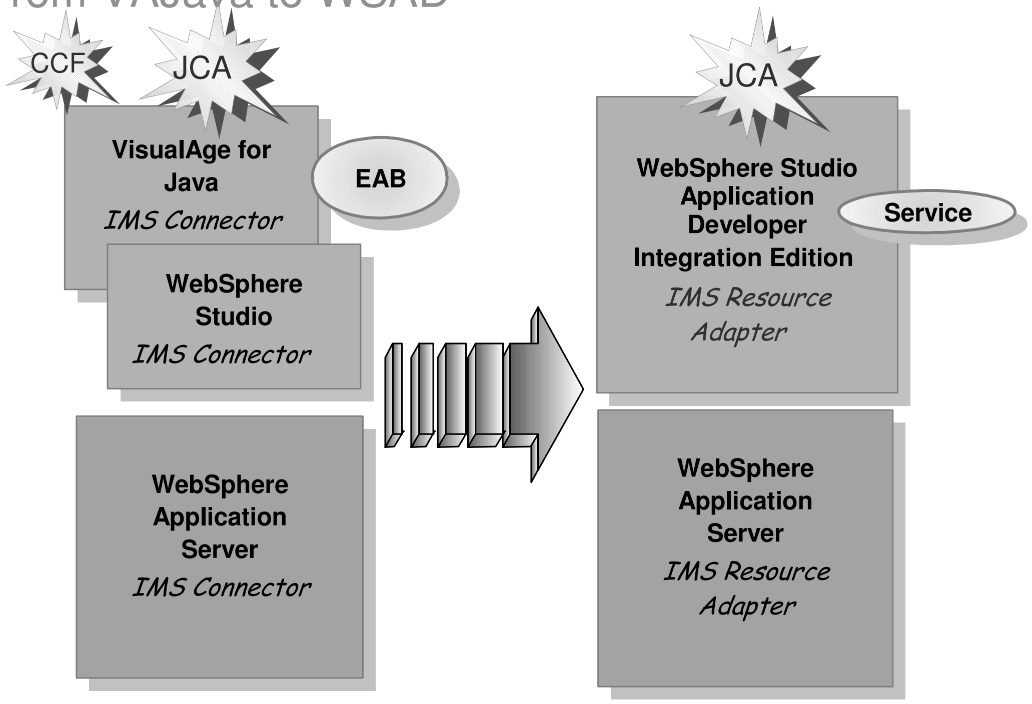
Checking Account DB

DB2

Savings Account DB

Redbooks

From VAJava to WSAD



CCF

JCA

VisualAge for Java

IMS Connector

EAB

WebSphere Studio

IMS Connector

WebSphere Application Server

IMS Connector

JCA

WebSphere Studio Application Developer Integration Edition

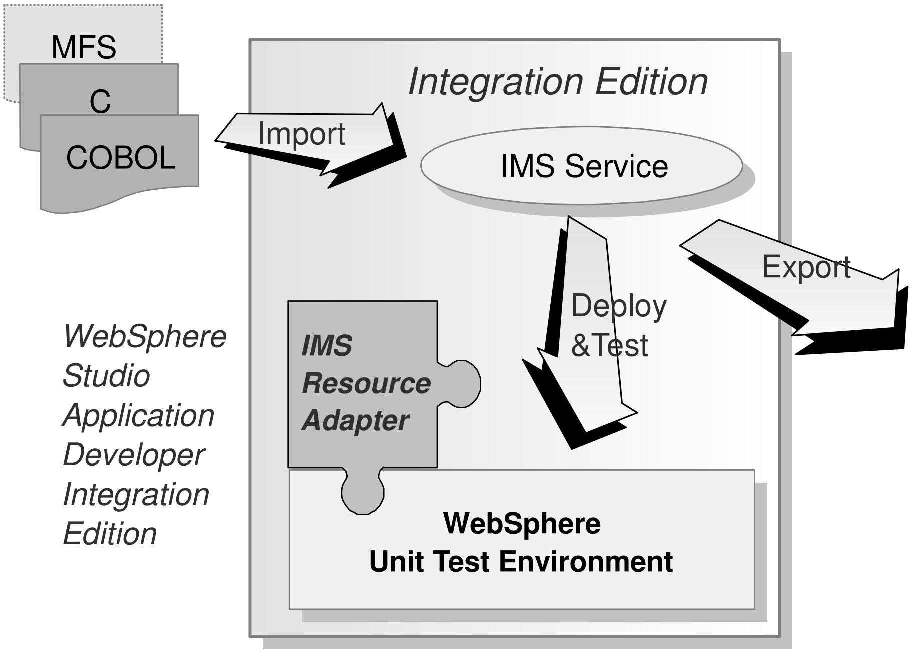
IMS Resource Adapter

Service

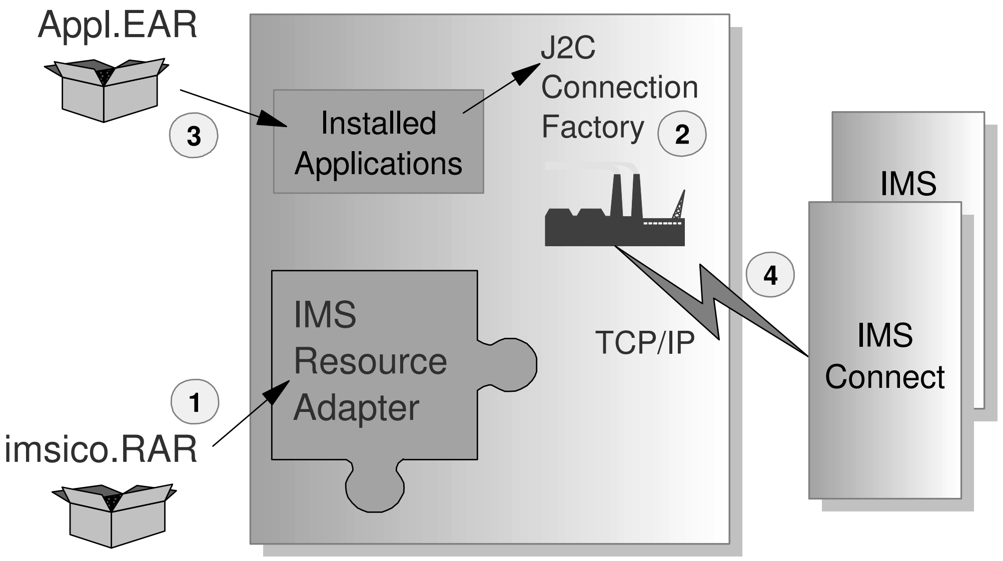
WebSphere Application Server

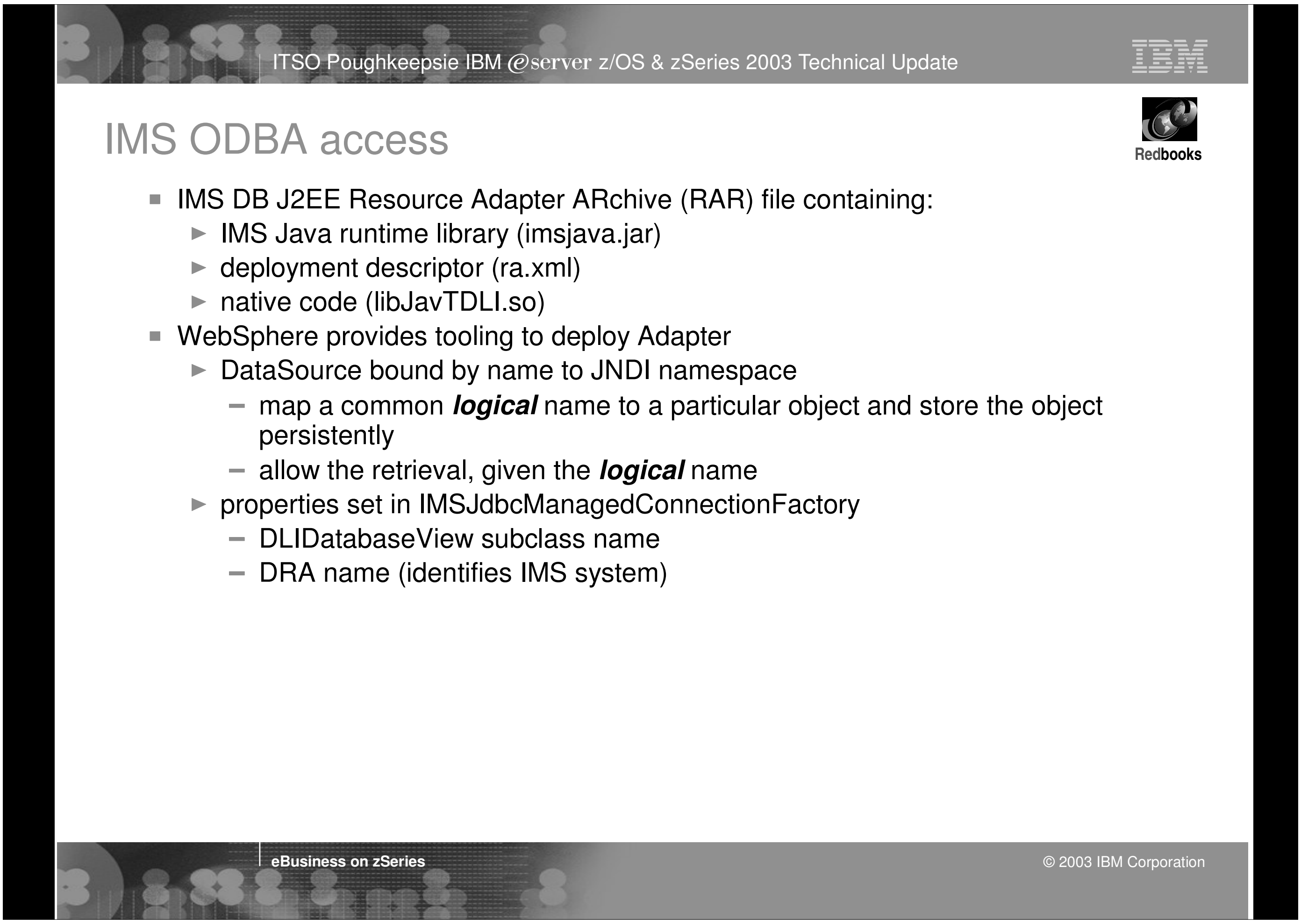
IMS Resource Adapter

Developing with WSAD IE



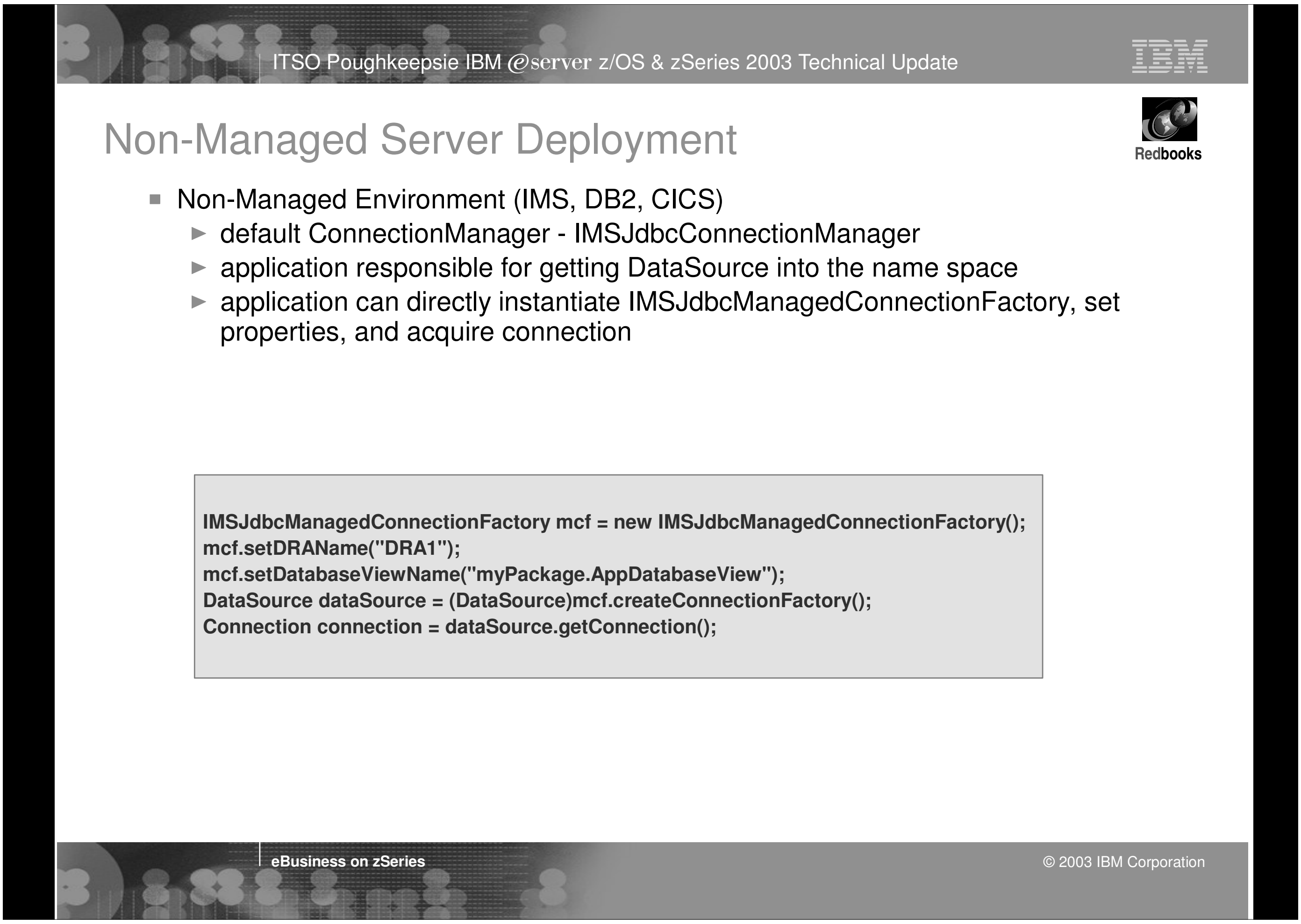
Running the application





IMS ODBA access

- IMS DB J2EE Resource Adapter ARchive (RAR) file containing:
  - IMS Java runtime library (imsjava.jar)
  - deployment descriptor (ra.xml)
  - native code (libJavTDLI.so)
- WebSphere provides tooling to deploy Adapter
  - DataSource bound by name to JNDI namespace
    - map a common ***logical*** name to a particular object and store the object persistently
    - allow the retrieval, given the ***logical*** name
  - properties set in IMSJdbcManagedConnectionFactory
    - DLIDatabaseView subclass name
    - DRA name (identifies IMS system)



Non-Managed Server Deployment

- Non-Managed Environment (IMS, DB2, CICS)
  - default ConnectionManager - IMSJdbcConnectionManager
  - application responsible for getting DataSource into the name space
  - application can directly instantiate IMSJdbcManagedConnectionFactory, set properties, and acquire connection

```
IMSJdbcManagedConnectionFactory mcf = new IMSJdbcManagedConnectionFactory();
mcf.setDRAName("DRA1");
mcf.setDatabaseViewName("myPackage.AppDatabaseView");
DataSource dataSource = (DataSource)mcf.createConnectionFactory();
Connection connection = dataSource.getConnection();
```

JCA Connection Pooling

- WebSphere maintains pool of already used Connections
- Put into pool when application closes Connection and associated UOW is commited
- Physical connection to IMS is not closed and the PSB is not deallocated (unless WebSphere destroys Connection)
- J2EE Connection Pooling only - no implementation of javax.sql.ConnectionPoolDataSource
- Connection pooling not optional, always on

JCA PreparedStatement caching

- Creation of PreparedStatement causes parse of SQL and generation of IMS SSAs
- IMS PreparedStatement are cached by ManagedConnectionFactory
- Keyed set based on SQL query
- Supported in unmanaged-server environments (only) when using JCA to acquire connections

Agenda

- Overview Java Connector Architecture
- CICS Transaction Gateway
- ➡ IMS Connect
  - IMS Connect overview
  - ➡ IMS Connect How To
- Messaging
- Connecting to databases

IMS Connect - main steps to set up

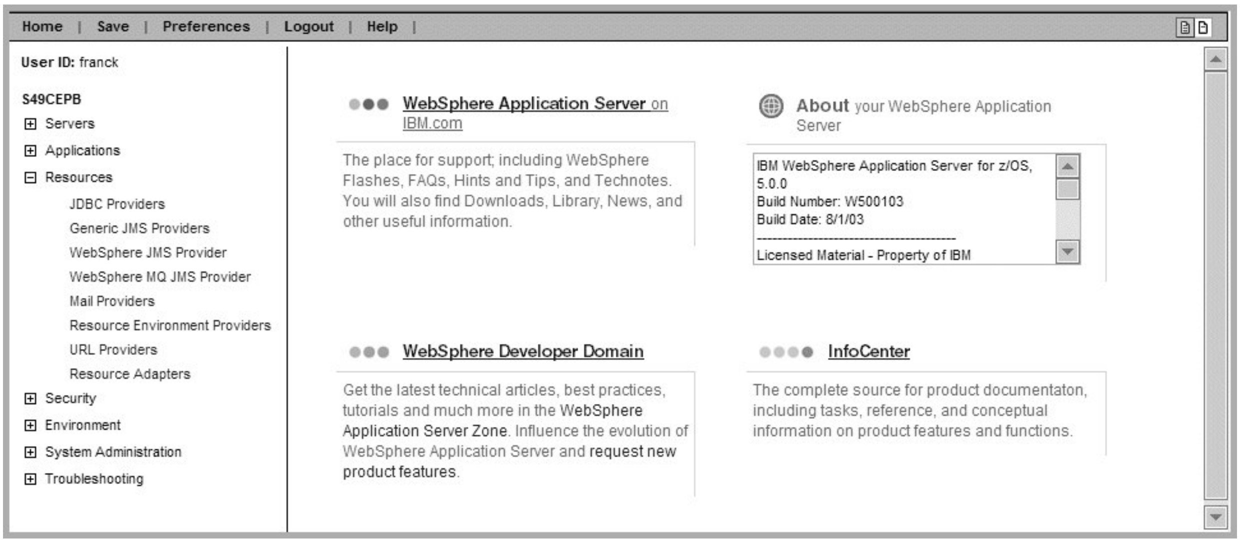
1. Setup IMS Connect
2. Install IMS Resource Adaptor - IMS Connector for Java
3. Define IMS J2C Connection factory
4. Define Connection factory - Custom Properties

IMS Connect - important info for the syspro...

```
//IMSECONN  PROC  RGN=4M,SOUT=S,SYS1=,
//             BPECFG=BPECFG00,HWSCFG=HWSCFG00,TCPDATA=TCPDATA
//*
//*******************************************************************
//*  BRING UP AN IMS CONNECT SYSTEM                                 *
//*******************************************************************
//STEP1    EXEC PGM=HWSHWS00,REGION=&RGN,TIME=1440,PARM='BPECFG=&BPECFG,HWSCFG=&HWSCFG'
//STEPLIB   DD  DSN=HWS.V2R1M0.SHWSRESL,DISP=SHR
//         DD   DSN=IMS810E.SDFSRESL,DISP=SHR
//PROCLIB  DD   DSN=IMS810E.&SYS1.PROCLIB,DISP=SHR
//SYSTCPD  DD   DSN=TCP.&SYSNAME..TCPPARMS(&TCPDATA),DISP=SHR
//SYSPRINT DD   SYSOUT=&SOUT
//SYSUDUMP DD   SYSOUT=&SOUT
//HWSRCORD DD   DSN=IMS810E.HWSRCORD,DISP=SHR
```

```
EDIT       IMS810E.PROCLIB(HWSCFG00) - 01.15          Columns 00001 00072
Command ===>                                             Scroll ===> CSR
****** ************************** Top of Data ****************************
000001 HWS (ID=IMSECONN,RRS=Y,RACF=N,XIBAREA=20)
000002 TCPIP (HOSTNAME=TCPIPOE,PORTID=(6001,LOCAL),MAXSOC=2000)
000003 DATASTORE (ID=IMSE,GROUP=IMS8EXCF,MEMBER=HWS810E,TMEMBER=SCSIMS8E)
 ****** ************************* Bottom of Data **************************
```

WebSphere console



User ID: franck

S49CEPB

WebSphere Application Server on IBM.com

The place for support; including WebSphere Flashes, FAQs, Hints and Tips, and Technotes. You will also find Downloads, Library, News, and other useful information.

About your WebSphere Application Server

IBM WebSphere Application Server for z/OS, 5.0.0
Build Number: W500103
Build Date: 8/1/03

Licensed Material - Property of IBM

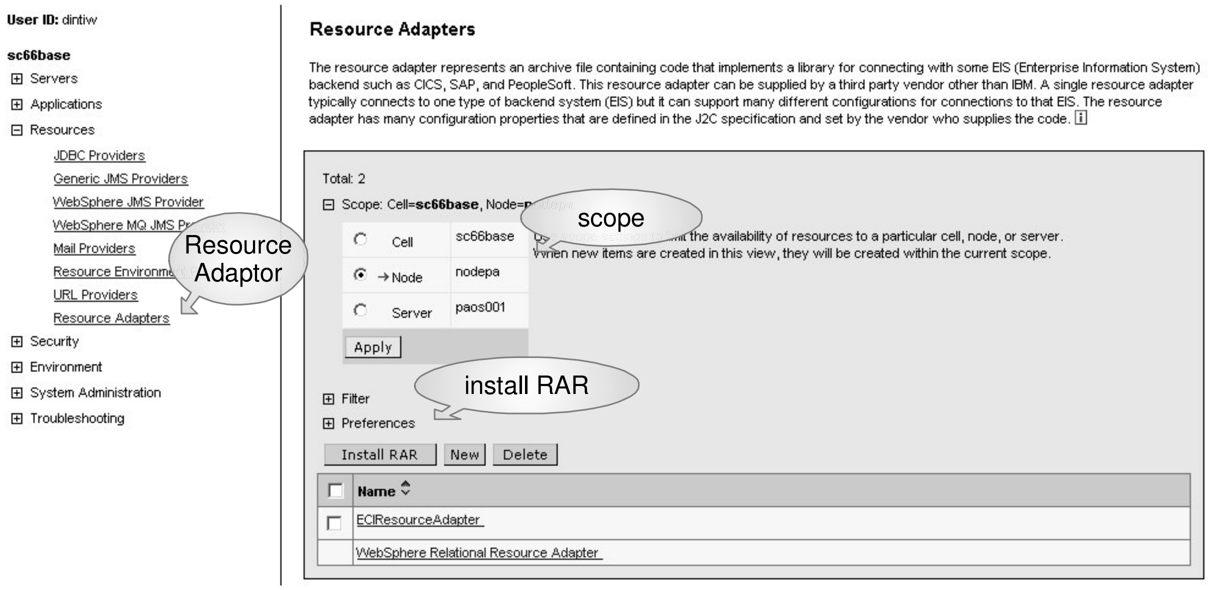
WebSphere Developer Domain

Get the latest technical articles, best practices, tutorials and much more in the WebSphere Application Server Zone. Influence the evolution of WebSphere Application Server and request new product features.

InfoCenter

The complete source for product documentaton, including tasks, reference, and conceptual information on product features and functions.

WebSphere console - installing the IMS Resource Adapter



Resource Adapters

The resource adapter represents an archive file containing code that implements a library for connecting with some EIS (Enterprise Information System) backend such as CICS, SAP, and PeopleSoft. This resource adapter can be supplied by a third party vendor other than IBM. A single resource adapter typically connects to one type of backend system (EIS) but it can support many different configurations for connections to that EIS. The resource adapter has many configuration properties that are defined in the J2C specification and set by the vendor who supplies the code.

Total: 2

⊟ Scope: Cell=**sc66base**, Node=

○ Cell	sc66base	the availability of resources to a particular cell, node, or server. When new items are created in this view, they will be created within the current scope.
● → Node	nodepa	
○ Server	paos001	

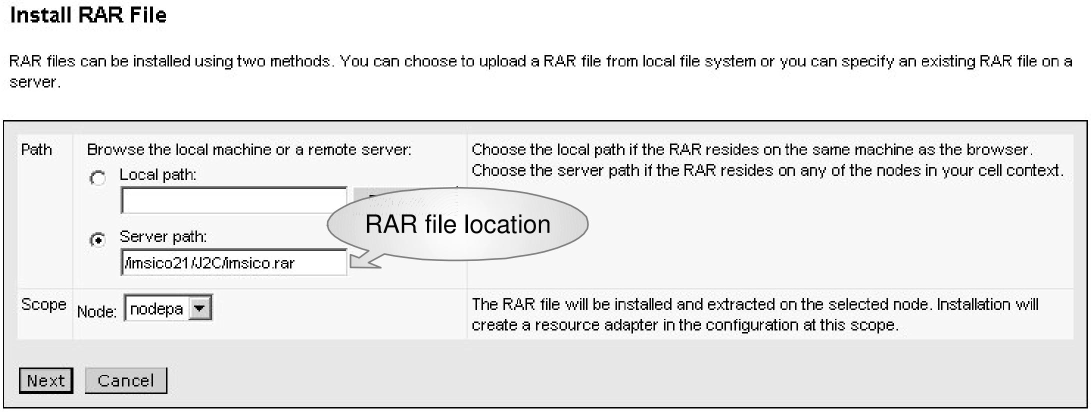
□	Name
□	ECIResourceAdapter
	WebSphere Relational Resource Adapter

Resource Adaptor

scope

install RAR

WebSphere console - installing RAR file from disk



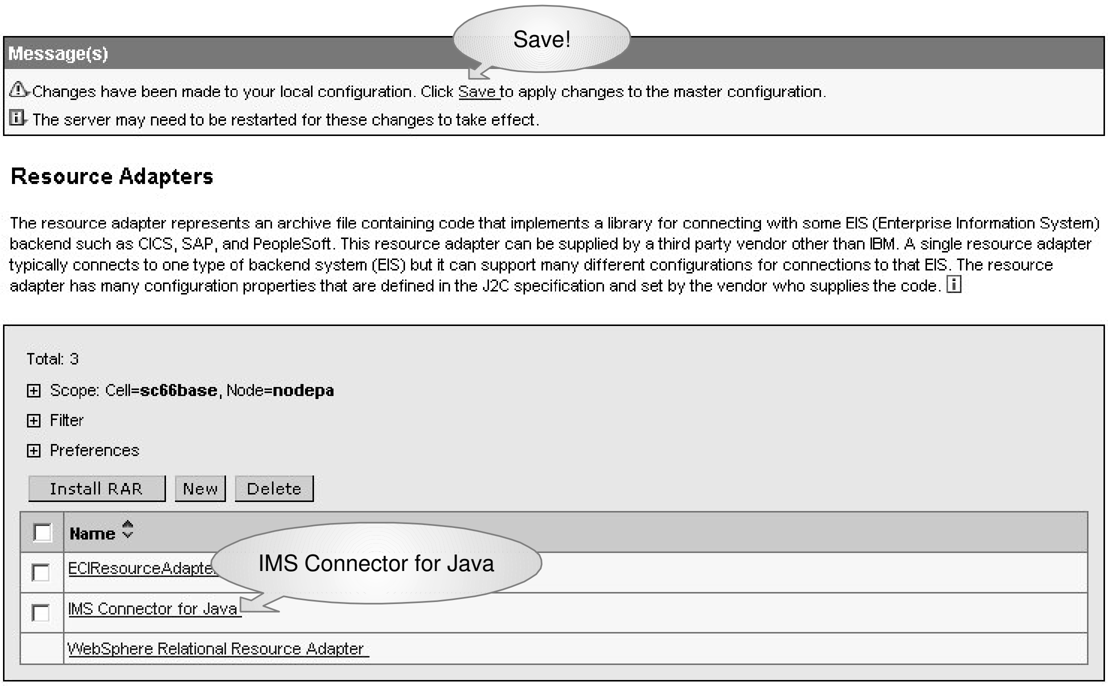
Install RAR File

RAR files can be installed using two methods. You can choose to upload a RAR file from local file system or you can specify an existing RAR file on a server.

Path	Browse the local machine or a remote server: ○ Local path: ● Server path: /imsico21/J2C/imsico.rar	Choose the local path if the RAR resides on the same machine as the browser. Choose the server path if the RAR resides on any of the nodes in your cell context.
Scope	Node: nodepa	The RAR file will be installed and extracted on the selected node. Installation will create a resource adapter in the configuration at this scope.

RAR file location

WebSphere console - selcting an existing RAR file



Save!

Message(s)

⚠ Changes have been made to your local configuration. Click Save to apply changes to the master configuration.

ⓘ The server may need to be restarted for these changes to take effect.

Resource Adapters

The resource adapter represents an archive file containing code that implements a library for connecting with some EIS (Enterprise Information System) backend such as CICS, SAP, and PeopleSoft. This resource adapter can be supplied by a third party vendor other than IBM. A single resource adapter typically connects to one type of backend system (EIS) but it can support many different configurations for connections to that EIS. The resource adapter has many configuration properties that are defined in the J2C specification and set by the vendor who supplies the code. ⓘ

Total: 3

⊞ Scope: Cell=**sc66base**, Node=**nodepa**

⊞ Filter

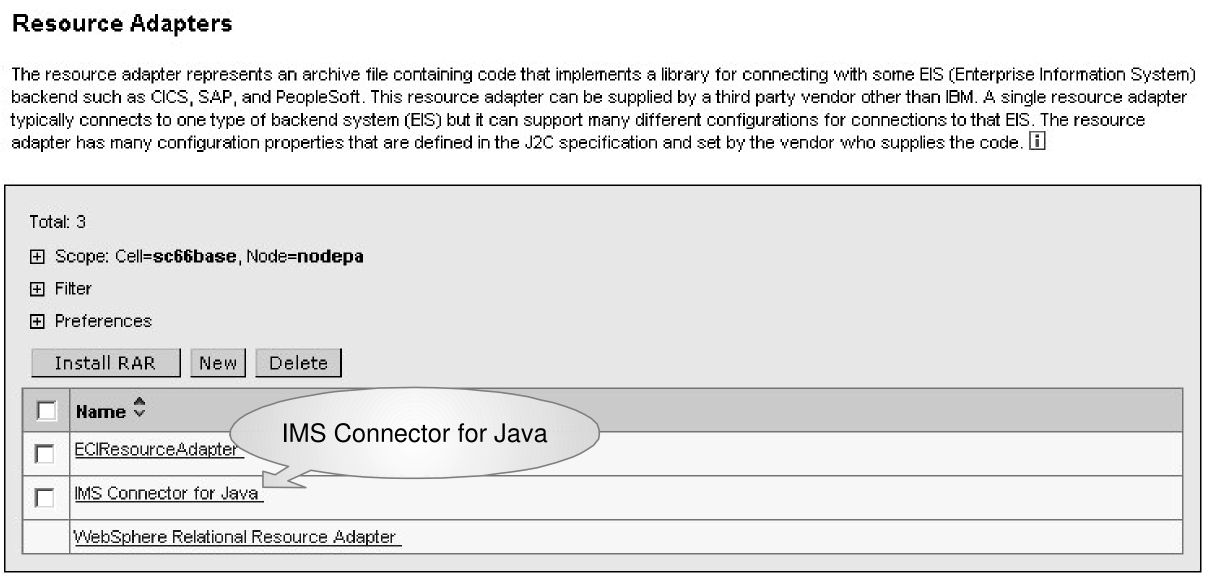
⊞ Preferences

Install RAR | New | Delete

□	**Name** ⇕
□	ECIResourceAdapter
□	IMS Connector for Java
	WebSphere Relational Resource Adapter

IMS Connector for Java

WebSphere V5 console - installing the RAR file



Resource Adapters

The resource adapter represents an archive file containing code that implements a library for connecting with some EIS (Enterprise Information System) backend such as CICS, SAP, and PeopleSoft. This resource adapter can be supplied by a third party vendor other than IBM. A single resource adapter typically connects to one type of backend system (EIS) but it can support many different configurations for connections to that EIS. The resource adapter has many configuration properties that are defined in the J2C specification and set by the vendor who supplies the code. ⓘ

Total: 3

⊞ Scope: Cell=**sc66base**, Node=**nodepa**

⊞ Filter

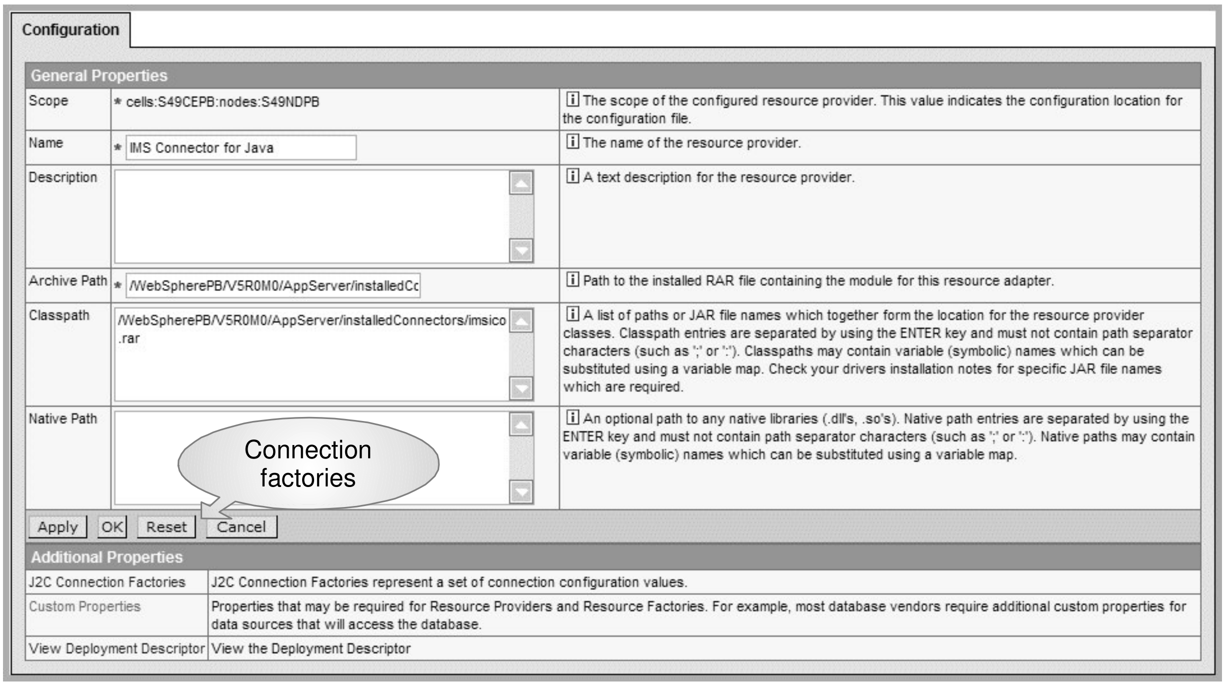
⊞ Preferences

Install RAR | New | Delete

□	**Name** ⇕
□	ECIResourceAdapter
□	IMS Connector for Java
	WebSphere Relational Resource Adapter

IMS Connector for Java

WebSphere V5 console - selecting J2C conn. fact.



Configuration

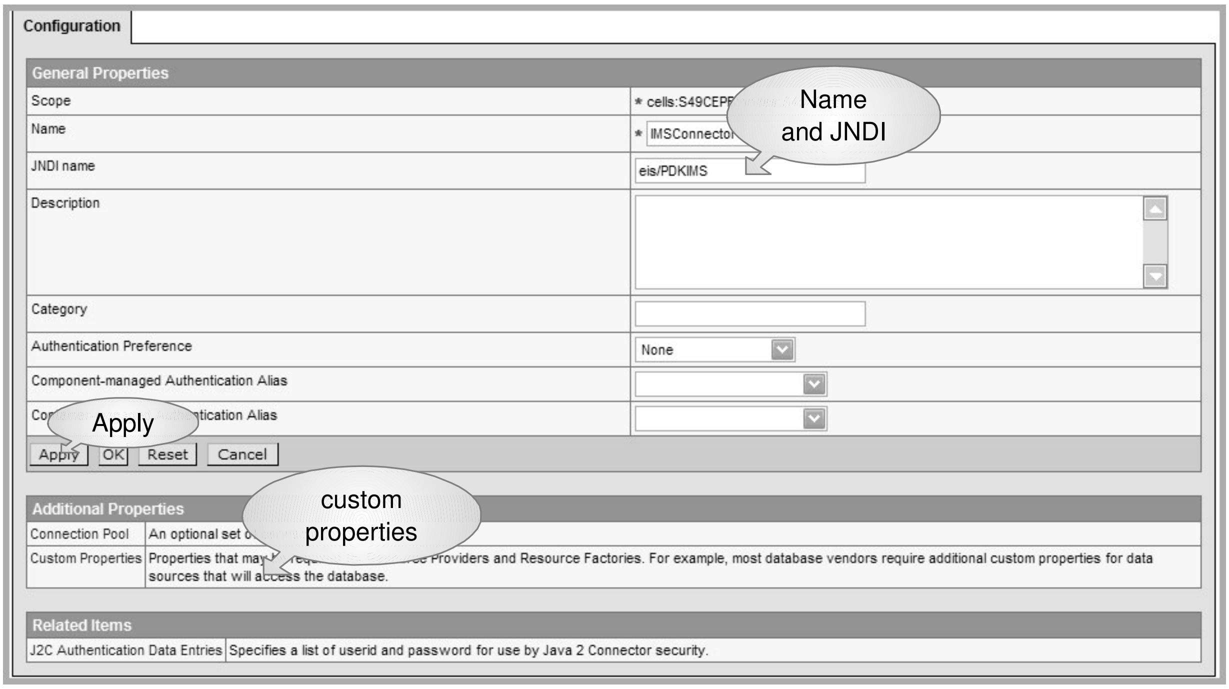
General Properties		
Scope	* cells:S49CEPB:nodes:S49NDPB	The scope of the configured resource provider. This value indicates the configuration location for the configuration file.
Name	* IMS Connector for Java	The name of the resource provider.
Description		A text description for the resource provider.
Archive Path	* /WebSpherePB/V5R0M0/AppServer/installedCc	Path to the installed RAR file containing the module for this resource adapter.
Classpath	/WebSpherePB/V5R0M0/AppServer/installedConnectors/imsico .rar	A list of paths or JAR file names which together form the location for the resource provider classes. Classpath entries are separated by using the ENTER key and must not contain path separator characters (such as ';' or ':'). Classpaths may contain variable (symbolic) names which can be substituted using a variable map. Check your drivers installation notes for specific JAR file names which are required.
Native Path		An optional path to any native libraries (.dll's, .so's). Native path entries are separated by using the ENTER key and must not contain path separator characters (such as ';' or ':'). Native paths may contain variable (symbolic) names which can be substituted using a variable map.

Apply | OK | Reset | Cancel

Additional Properties	
J2C Connection Factories	J2C Connection Factories represent a set of connection configuration values.
Custom Properties	Properties that may be required for Resource Providers and Resource Factories. For example, most database vendors require additional custom properties for data sources that will access the database.
View Deployment Descriptor	View the Deployment Descriptor

Connection factories

WebSphere V5 console - JNDI name and add. prop.



Configuration

General Properties	
Scope	* cells:S49CEPF
Name	* IMSConnector
JNDI name	eis/PDKIMS
Description	
Category	
Authentication Preference	None
Component-managed Authentication Alias	
Co tication Alias	

Apply | OK | Reset | Cancel

Additional Properties	
Connection Pool	An optional set o
Custom Properties	Properties that may e Providers and Resource Factories. For example, most database vendors require additional custom properties for data sources that will access the database.

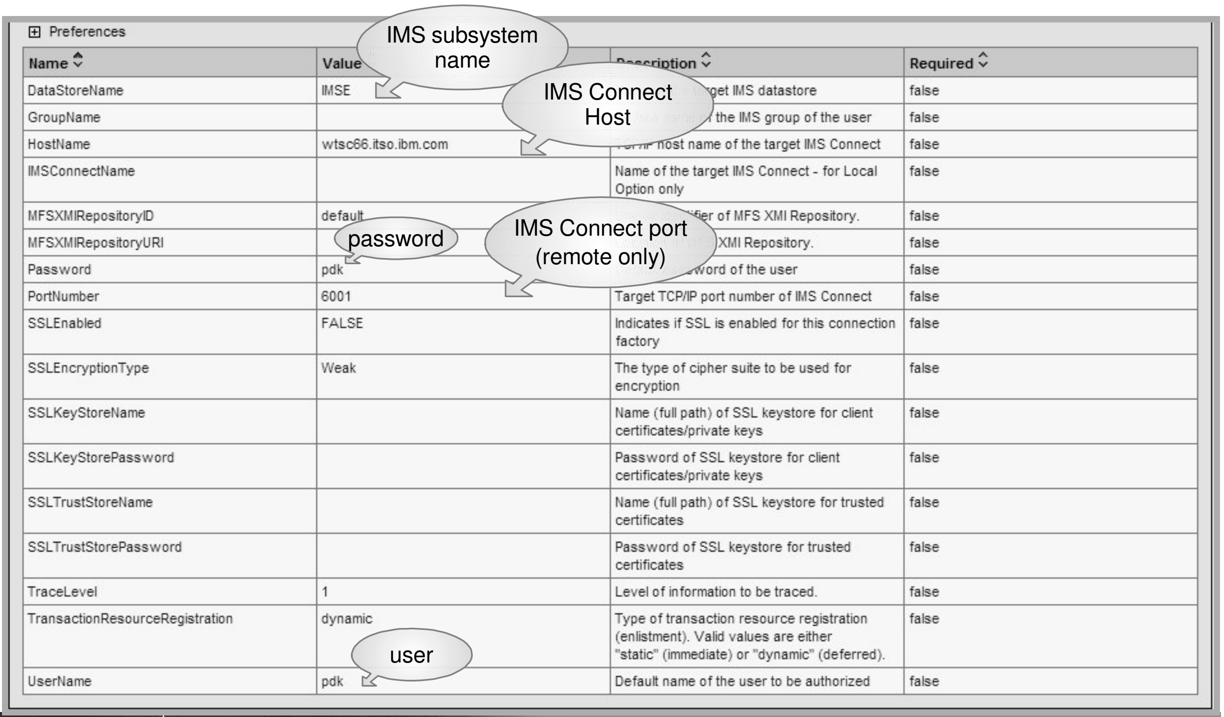
Related Items	
J2C Authentication Data Entries	Specifies a list of userid and password for use by Java 2 Connector security.

Name and JNDI

Apply

custom properties

WebSphere V5 console - properties for IMS Connect



⊞ Preferences

Name	Value	Description	Required
DataStoreName	IMSE	get IMS datastore	false
GroupName		the IMS group of the user	false
HostName	wtsc66.itso.ibm.com	host name of the target IMS Connect	false
IMSConnectName		Name of the target IMS Connect - for Local Option only	false
MFSXMIRepositoryID	default	ifier of MFS XMI Repository.	false
MFSXMIRepositoryURI		XMI Repository.	false
Password	pdk	word of the user	false
PortNumber	6001	Target TCP/IP port number of IMS Connect	false
SSLEnabled	FALSE	Indicates if SSL is enabled for this connection factory	false
SSLEncryptionType	Weak	The type of cipher suite to be used for encryption	false
SSLKeyStoreName		Name (full path) of SSL keystore for client certificates/private keys	false
SSLKeyStorePassword		Password of SSL keystore for client certificates/private keys	false
SSLTrustStoreName		Name (full path) of SSL keystore for trusted certificates	false
SSLTrustStorePassword		Password of SSL keystore for trusted certificates	false
TraceLevel	1	Level of information to be traced.	false
TransactionResourceRegistration	dynamic	Type of transaction resource registration (enlistment). Valid values are either "static" (immediate) or "dynamic" (deferred).	false
UserName	pdk	Default name of the user to be authorized	false

Agenda

- Overview Java Connector Architecture
- CICS Transaction Gateway
- IMS Connect
- ➡ Messaging
  - ➡ Overview
- Connecting to databases

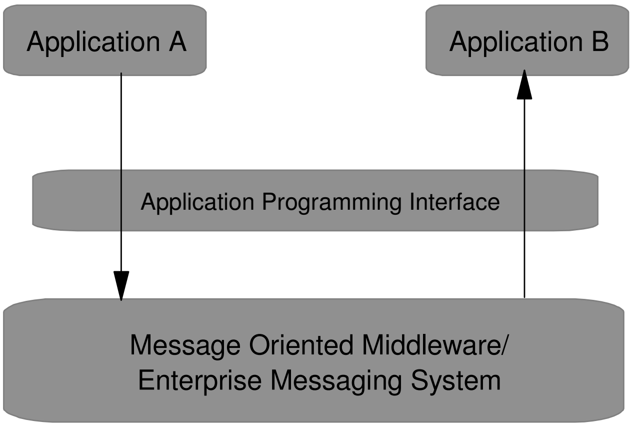
Types of integration

- "Integration at the glass"
  - at the user interface level
  - for example, filling in a field in a window/screen will result directly in changing a value of another field in another window/screen
  - WebSphere Portal Server
- Information connectivity
  - messaging
    - exchanging messages between components
      - "request/reply"
      - "send and forget"
  - triggering of components
  - WebSphere MQ
  - Information integration
    - "mapping" and transformation of data
    - WebSphere MQ Integrator
- Business Process Integration
  - integration focus shifts from applications and data to business processes
  - MQSeries Workflow and IBM CrossWorlds

Enterprise Messaging Systems

- Known as Message Oriented Middleware (MOM)
- Mechanism to integrate application in a loosely coupled, flexible manner
- Asynchronous delivery of data between applications on a *store and forward* basis
- MOM provides assured delivery of messages
- WebSphere MQ is a MOM solution

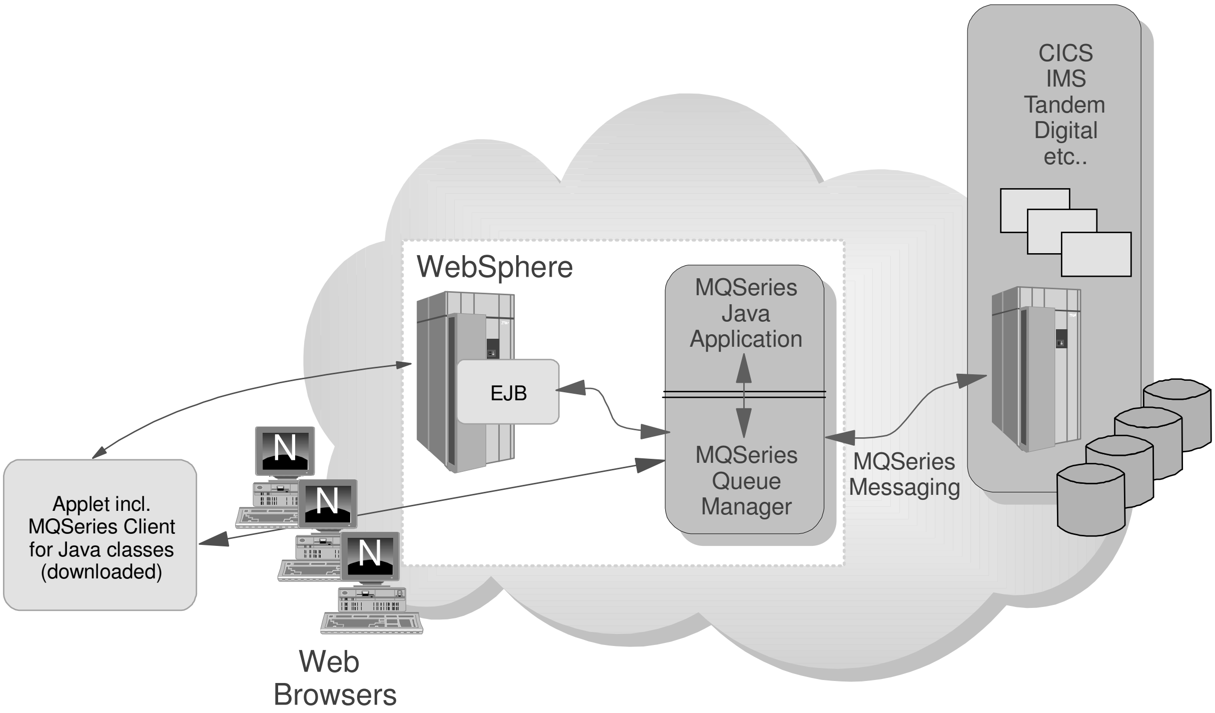
Messaging Flexibility and Loose Coupling


Application A
Application B
Application Programming Interface
Message Oriented Middleware/
Enterprise Messaging System

WebSphere MQ as a bridge between Java and EIS

- WebSphere MQ (messaging, integrator and workflow) can be used as a bridge between a Java client and an EIS
- Specific Java classes exist that can be used to code access to a queue server in local (bindings) or remote mode
- For CICS and IMS, bridges are available to trigger transactions based on an incoming message
  - becomes an even more interesting option when the EIS resides on a platform with little Java support
- Extends the J2EE infrastructure by supporting the JMS API

Using MQSeries to get from Java to EIS



CICS
IMS
Tandem
Digital
etc..

WebSphere

MQSeries Java Application

EJB

MQSeries Queue Manager

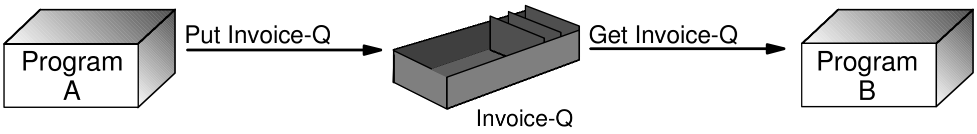
MQSeries Messaging

Applet incl. MQSeries Client for Java classes (downloaded)

Web Browsers

Models (Point to Point)

Send and Forget



Program A

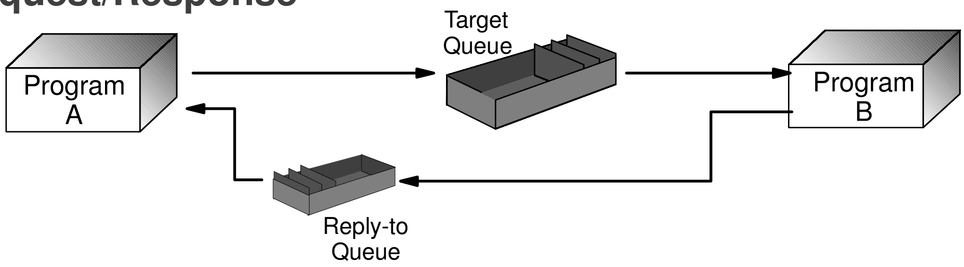
Put Invoice-Q

Get Invoice-Q

Invoice-Q

Program B

Request/Response

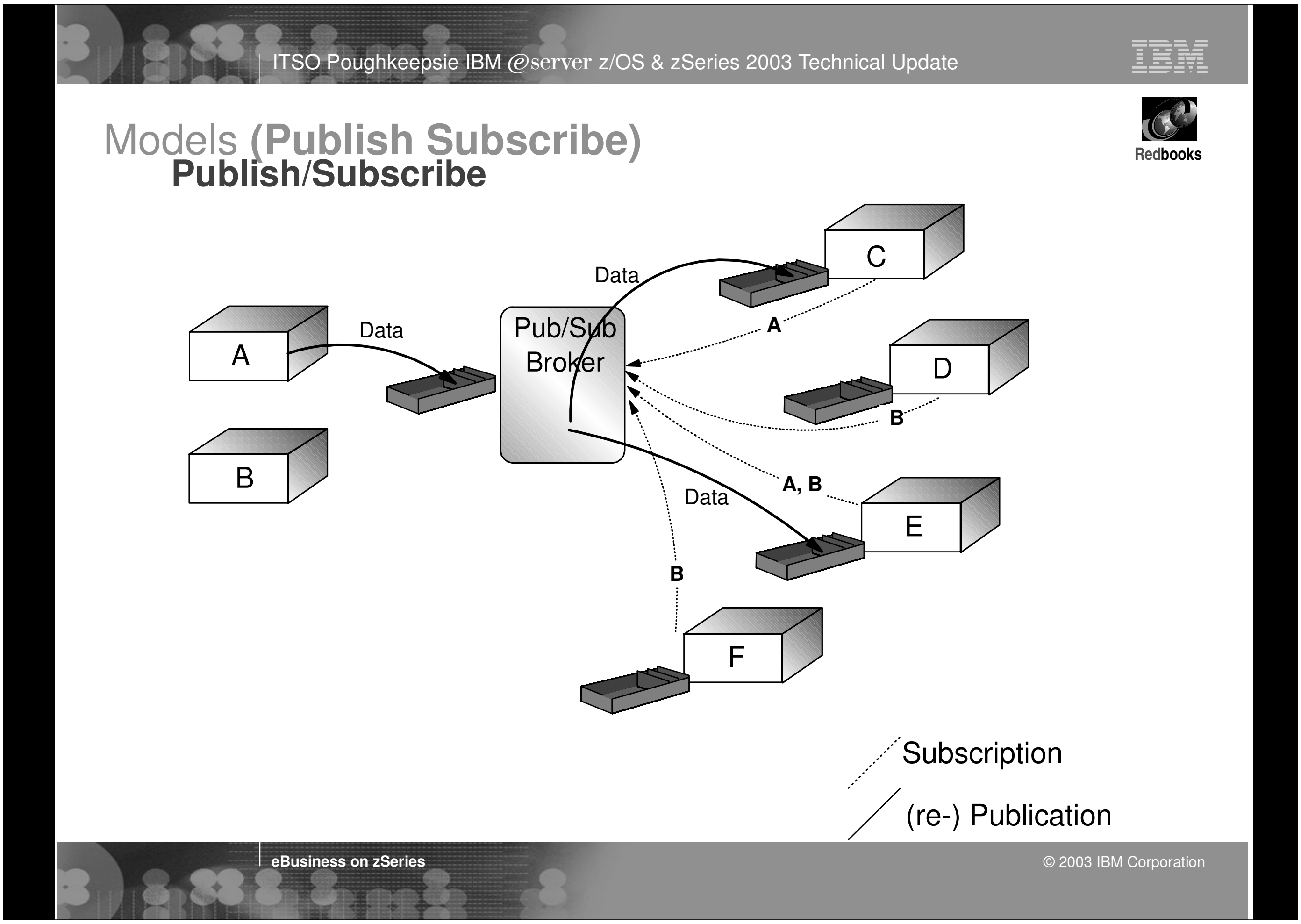


Program A

Target Queue

Program B

Reply-to Queue



Models (Publish Subscribe)
Publish/Subscribe

A

B

Data

Pub/Sub Broker

Data

C

A

D

B

A, B

Data

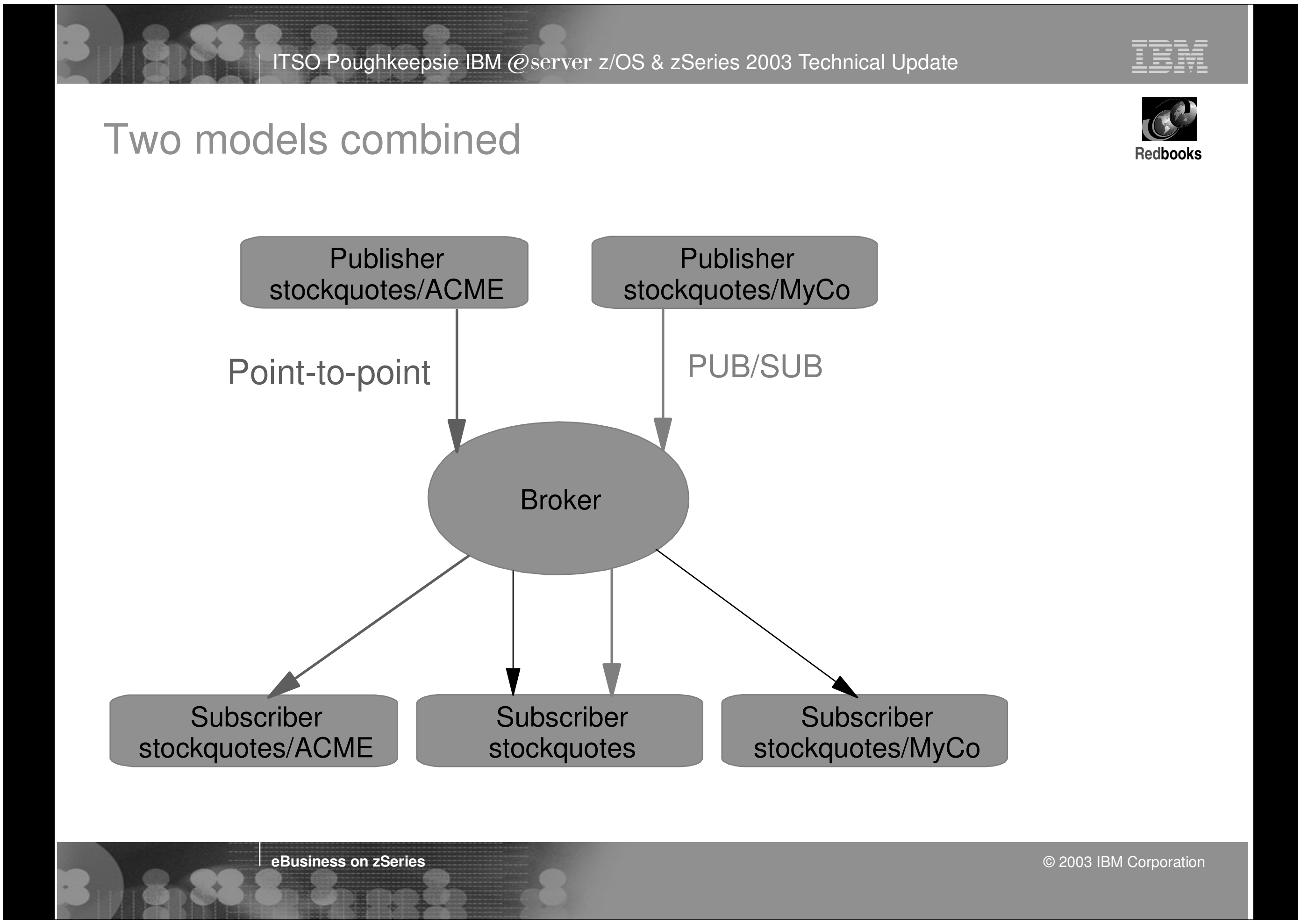
E

B

F

Subscription

(re-) Publication



Two models combined

Publisher stockquotes/ACME

Publisher stockquotes/MyCo

Point-to-point

PUB/SUB

Broker

Subscriber stockquotes/ACME

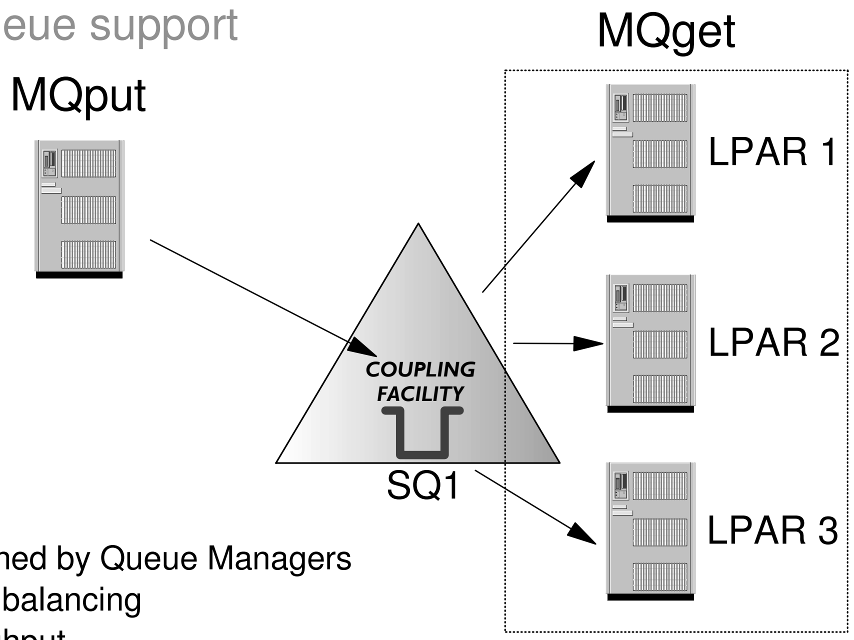
Subscriber stockquotes

Subscriber stockquotes/MyCo

What about MQseries on z/OS?

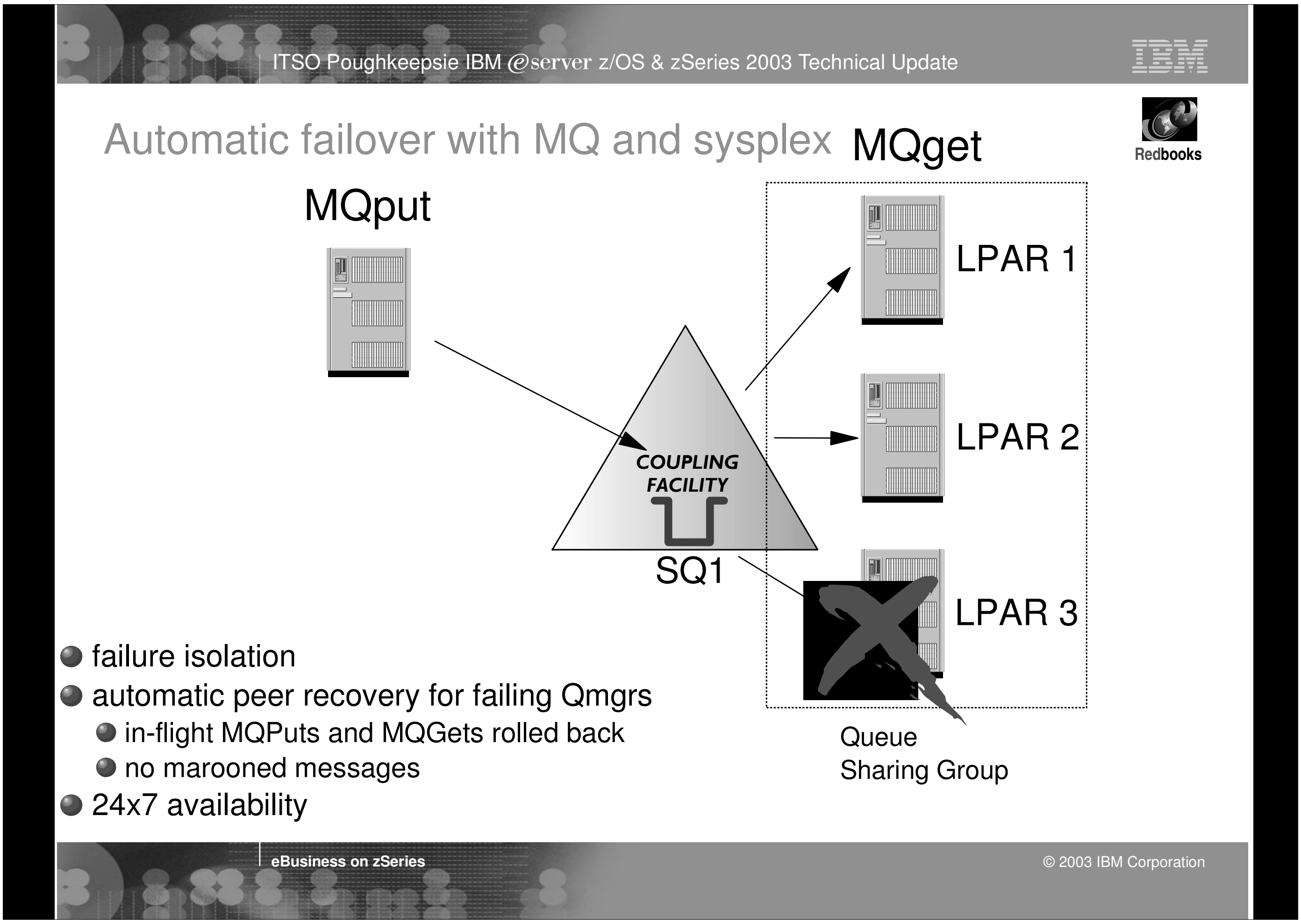
- Built from the "ground up" to exploit z/OS RAS capabilities and customer expectations
- Runs as formal MVS subsystem
- Exploits ARM, WLM and Parallel Sysplex
- Provides "PC" access from CICS, IMS, WAS, DB2 SPs, Batch and TSO application execution environments
- Specialized "bridge" support for CICS and IMS transactions
- Provides full participation in transactions coordinated by CICS, IMS, WebSphere and DB2 through RRS
- Capable of supporting 1000's of msgs/sec
- Many supporting vendor tools
  - Candle, BMC, CA

Shared queue support



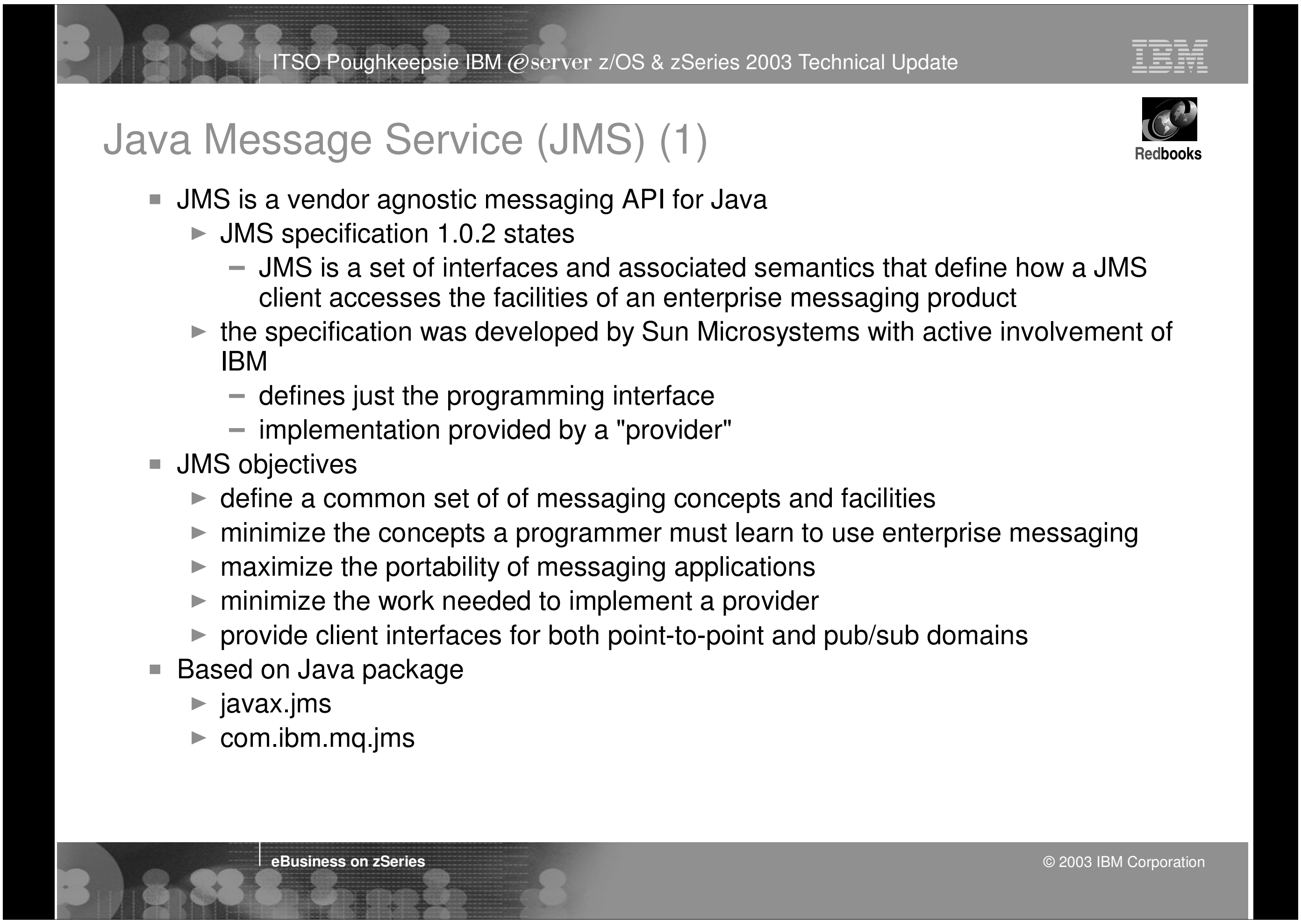
- queues not owned by Queue Managers
- automatic load balancing
- scalable throughput
- multiple processors can access the same queue
- queue sharing groups

Queue
Sharing Group



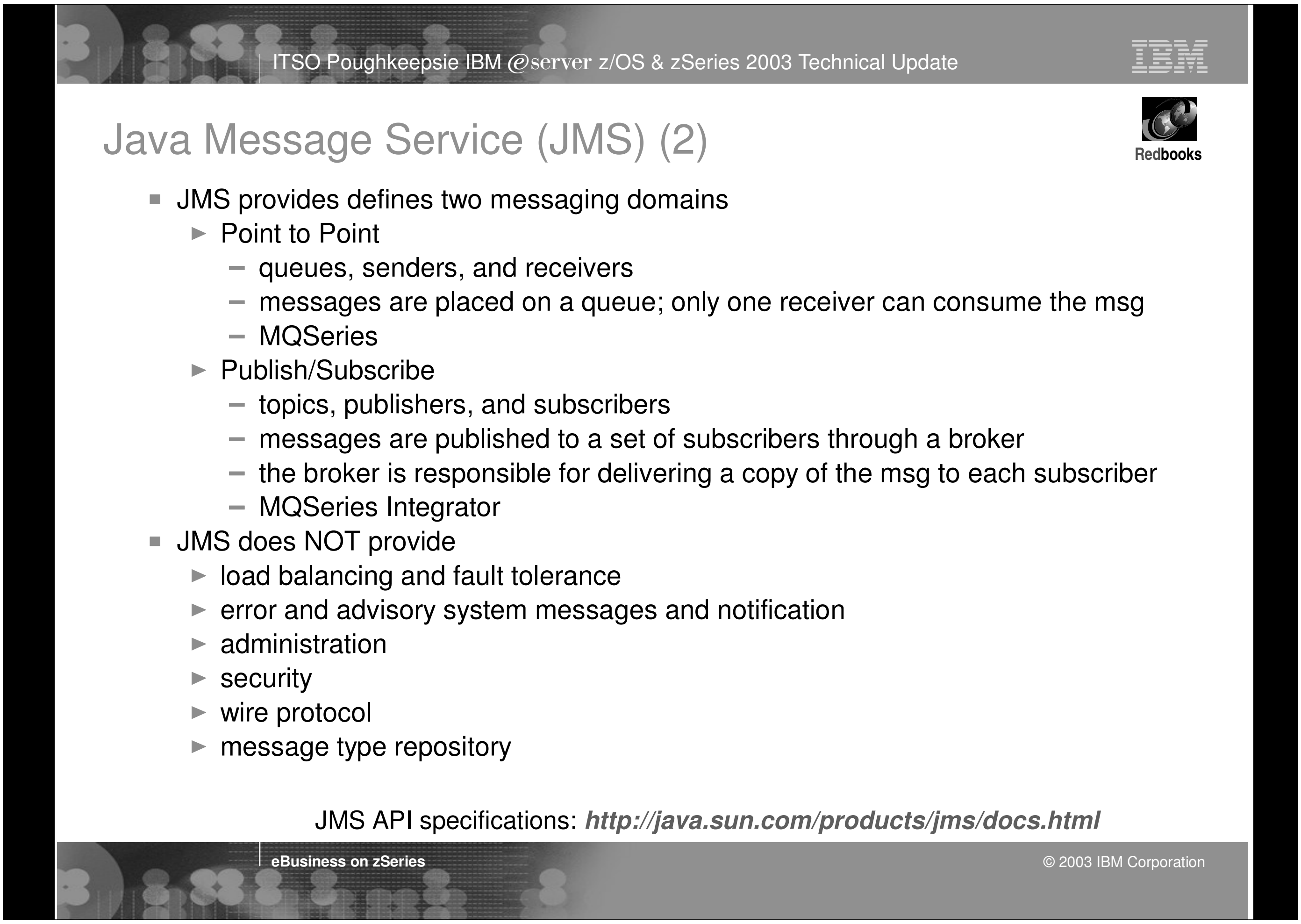
Automatic failover with MQ and sysplex

- failure isolation
- automatic peer recovery for failing Qmgrs
  - in-flight MQPuts and MQGets rolled back
  - no marooned messages
- 24x7 availability



Java Message Service (JMS) (1)

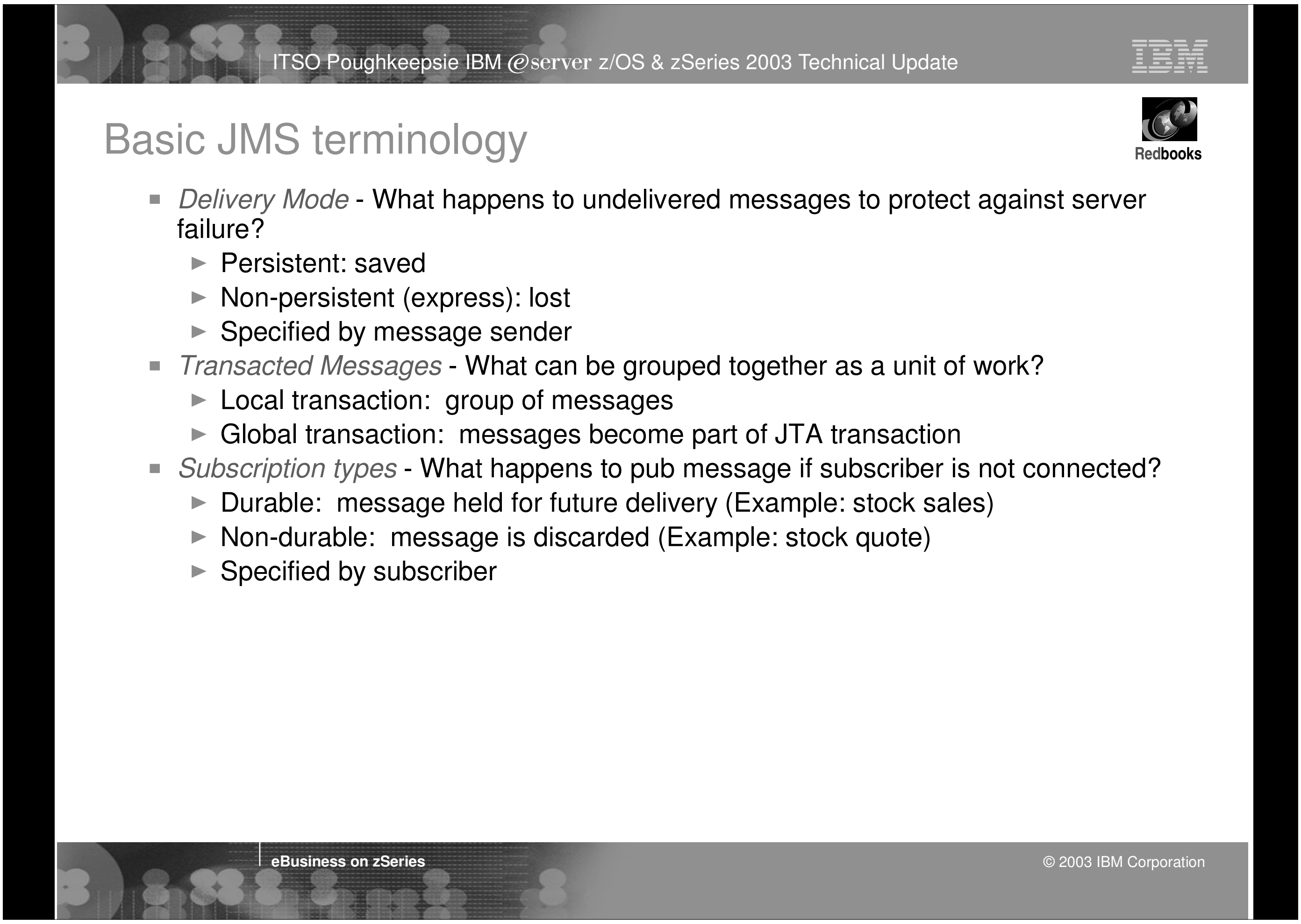
- JMS is a vendor agnostic messaging API for Java
  - JMS specification 1.0.2 states
    - JMS is a set of interfaces and associated semantics that define how a JMS client accesses the facilities of an enterprise messaging product
  - the specification was developed by Sun Microsystems with active involvement of IBM
    - defines just the programming interface
    - implementation provided by a "provider"
- JMS objectives
  - define a common set of of messaging concepts and facilities
  - minimize the concepts a programmer must learn to use enterprise messaging
  - maximize the portability of messaging applications
  - minimize the work needed to implement a provider
  - provide client interfaces for both point-to-point and pub/sub domains
- Based on Java package
  - javax.jms
  - com.ibm.mq.jms



Java Message Service (JMS) (2)

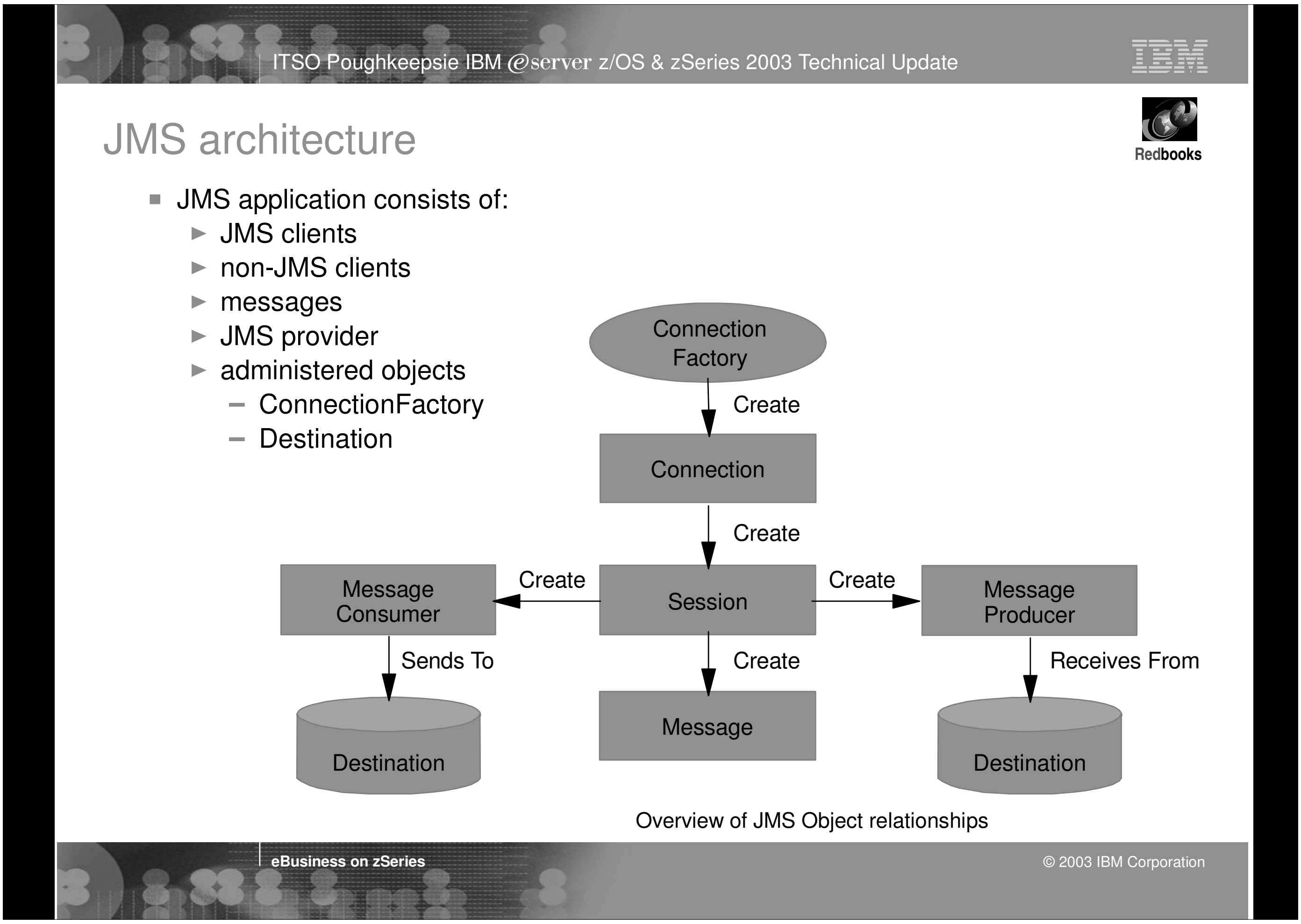
- JMS provides defines two messaging domains
  - Point to Point
    - queues, senders, and receivers
    - messages are placed on a queue; only one receiver can consume the msg
    - MQSeries
  - Publish/Subscribe
    - topics, publishers, and subscribers
    - messages are published to a set of subscribers through a broker
    - the broker is responsible for delivering a copy of the msg to each subscriber
    - MQSeries Integrator
- JMS does NOT provide
  - load balancing and fault tolerance
  - error and advisory system messages and notification
  - administration
  - security
  - wire protocol
  - message type repository

JMS API specifications: ***http://java.sun.com/products/jms/docs.html***



Basic JMS terminology

- *Delivery Mode* - What happens to undelivered messages to protect against server failure?
  - Persistent: saved
  - Non-persistent (express): lost
  - Specified by message sender
- *Transacted Messages* - What can be grouped together as a unit of work?
  - Local transaction: group of messages
  - Global transaction: messages become part of JTA transaction
- *Subscription types* - What happens to pub message if subscriber is not connected?
  - Durable: message held for future delivery (Example: stock sales)
  - Non-durable: message is discarded (Example: stock quote)
  - Specified by subscriber

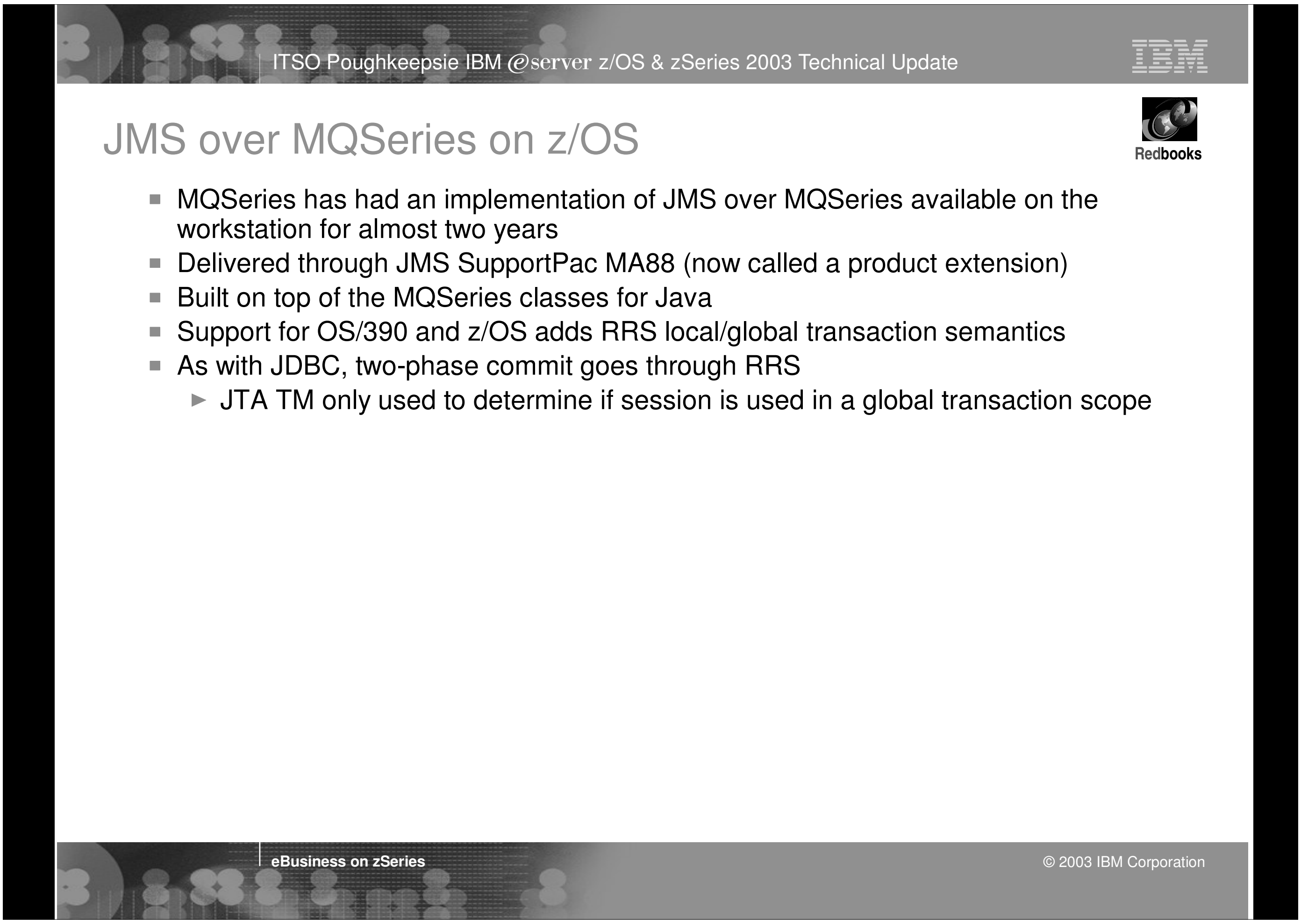


JMS architecture

- JMS application consists of:
  - JMS clients
  - non-JMS clients
  - messages
  - JMS provider
  - administered objects
    - ConnectionFactory
    - Destination

Connection Factory
Create
Connection
Create
Message Consumer
Create
Session
Create
Message Producer
Sends To
Create
Receives From
Destination
Message
Destination

Overview of JMS Object relationships



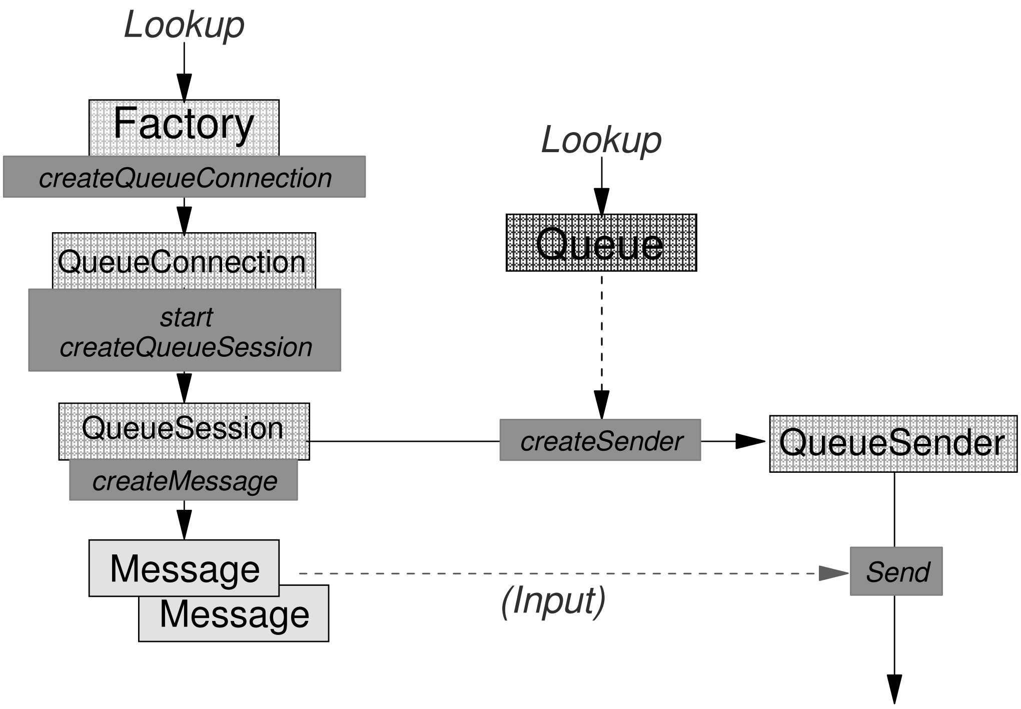
JMS over MQSeries on z/OS

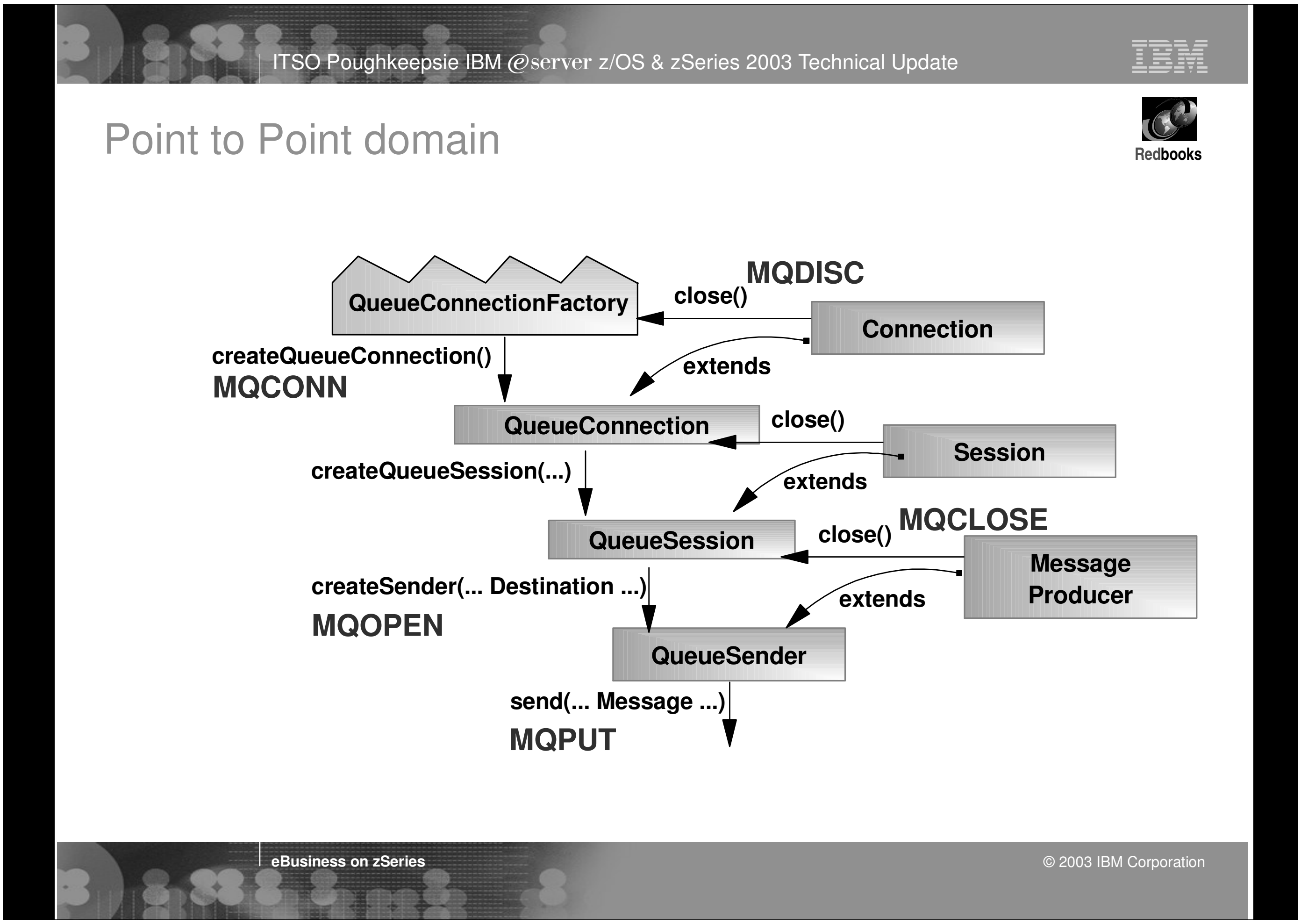
- MQSeries has had an implementation of JMS over MQSeries available on the workstation for almost two years
- Delivered through JMS SupportPac MA88 (now called a product extension)
- Built on top of the MQSeries classes for Java
- Support for OS/390 and z/OS adds RRS local/global transaction semantics
- As with JDBC, two-phase commit goes through RRS
  - JTA TM only used to determine if session is used in a global transaction scope

JMS Implementation in WebSphere V4.01 and V5

- Built on WebSphere MQ
  - numerous APIs:
    - Message Queue Interface (MQI)
    - Application Messaging Interface (AMI)
    - Java Messaging Interface (JMS)
  - numerous communication modes including
    - Point-to-Point
    - PUB/SUB
- Introduced with Java Message Support Pac MA88
- In WebSphere V4.01 and V5 for z/OS, a JMS client can participate in a global transaction via the Java Transaction API (JTA) using RRS

Point to Point - example





ITSO Poughkeepsie IBM @server z/OS & zSeries 2003 Technical Update

IBM

Redbooks

Point to Point domain

QueueConnectionFactory

close()

MQDISC

Connection

createQueueConnection()

MQCONN

extends

QueueConnection

close()

Session

createQueueSession(...)

extends

QueueSession

close()

MQCLOSE

Message Producer

createSender(... Destination ...)

MQOPEN

extends

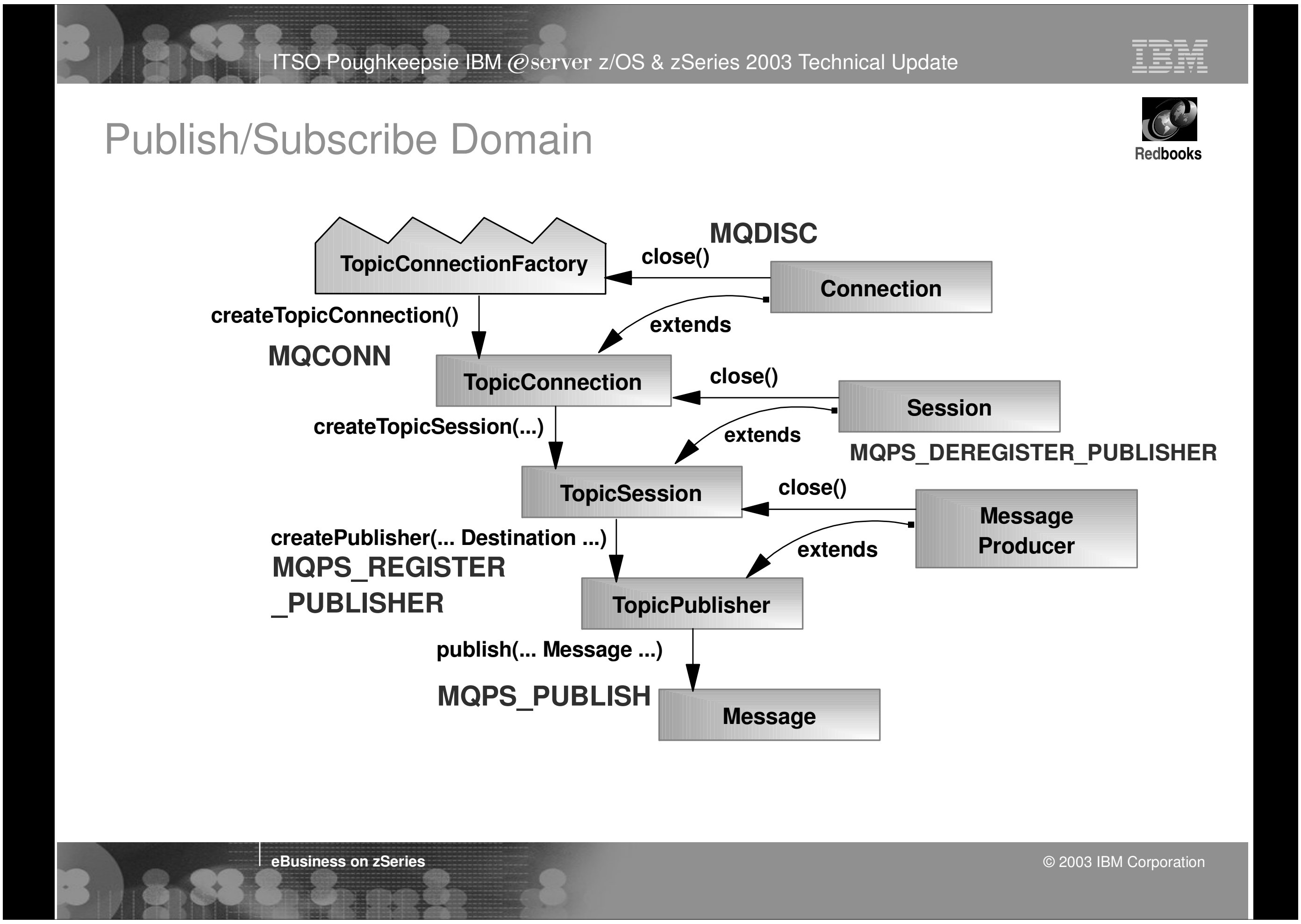
QueueSender

send(... Message ...)

MQPUT

eBusiness on zSeries

© 2003 IBM Corporation



ITSO Poughkeepsie IBM @server z/OS & zSeries 2003 Technical Update

IBM

Redbooks

Publish/Subscribe Domain

TopicConnectionFactory

close()

MQDISC

Connection

createTopicConnection()

MQCONN

extends

TopicConnection

close()

Session

createTopicSession(...)

extends

MQPS_DEREGISTER_PUBLISHER

TopicSession

close()

Message Producer

createPublisher(... Destination ...)

MQPS_REGISTER _PUBLISHER

extends

TopicPublisher

publish(... Message ...)

MQPS_PUBLISH

Message

eBusiness on zSeries

© 2003 IBM Corporation

Software prerequisites

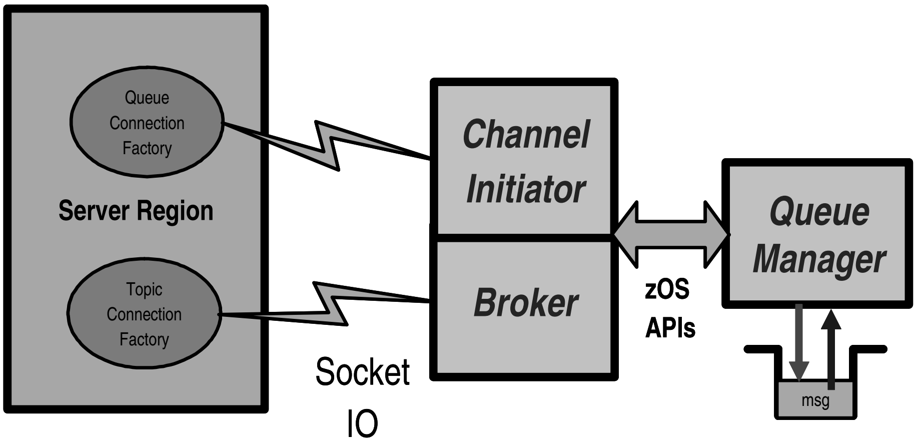
- WebSphere Application Server V4.01 for z/OS and OS/390
- WebSphere MQ V5.2 for z/OS and OS/390
  - JMS enabling PTF (PQ52271/UQ59032 sup by UQ65906)
  - Timer Service PTF (PQ52891/UQ59338
- Java Message Service Support Pac MA88 for z/OS and OS/390
  - http://www.ibm.com/software/ts/mqseries/txppacs/ma88.html
- SDK V1.3.1
- AAT and SMEUI

- WebSphere Application Server V5.0.1 for z/OS and OS/390
  - JMS provider ships limited function MQ 5.3.1. For full function, you need . .
- WebSphere MQ V5.3.1 for z/OS and OS/390

lote: *Current PTF maintenance is strongly recommended to avoid unnecessary problems !!!*

JMS Provider

- J2EE 1.3 CTS Compliant
  - Java connector providing the JMS Programming Model
- Fenced off zOS Qualitites of Service
  - TCP/IP Connectivity Only
    - Messages and other protocol traffic flow through socket.
  - XA Transactions
  - Both WebSphere zOS 5.0 and Queue Manager support XA Transactions.



Connection factories for WebSphere for z/OS

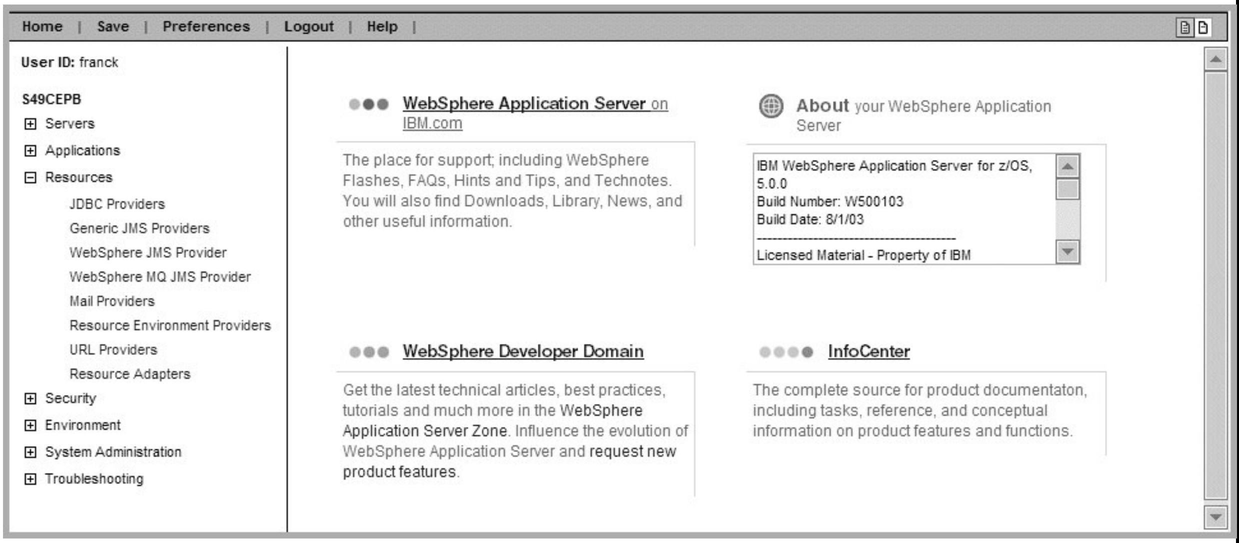
- RRS-enabled, capable of 2-Phase Commit
  - MQRRSQueueConnectionFactory
  - MQRRSTopicConnectionFactory
    - used for the J2EE programming Model
- MQ-specific capable of 0 / 1-Phase Commit semantics only
  - MQQueueConnectionFactory
  - MQTopicConnectionFactory

Agenda

- Overview Java Connector Architecture
- CICS Transaction Gateway
- IMS Connect
- ➡ Messaging
  - Overview
  - ➡ How To
- Connecting to databases

WebSphere V5 console - JMS Provider definition

Requestor tier, WebSphere MQ JMS Provider
Enterprise tier, WebSphere MQ JMS Provider and MDB listener



User ID: franck

S49CEPB
- ⊞ Servers
- ⊞ Applications
- ⊟ Resources
  - JDBC Providers
  - Generic JMS Providers
  - WebSphere JMS Provider
  - WebSphere MQ JMS Provider
  - Mail Providers
  - Resource Environment Providers
  - URL Providers
  - Resource Adapters
- ⊞ Security
- ⊞ Environment
- ⊞ System Administration
- ⊞ Troubleshooting

WebSphere Application Server on IBM.com

The place for support; including WebSphere Flashes, FAQs, Hints and Tips, and Technotes. You will also find Downloads, Library, News, and other useful information.

About your WebSphere Application Server

IBM WebSphere Application Server for z/OS, 5.0.0
Build Number: W500103
Build Date: 8/1/03

Licensed Material - Property of IBM

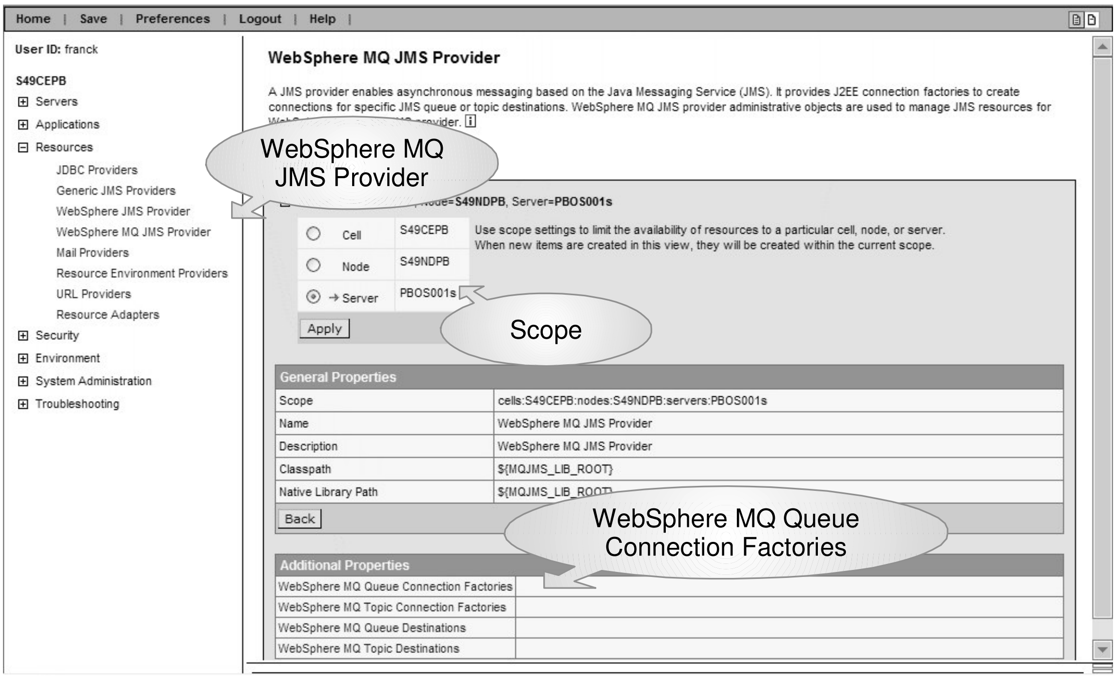
WebSphere Developer Domain

Get the latest technical articles, best practices, tutorials and much more in the WebSphere Application Server Zone. Influence the evolution of WebSphere Application Server and request new product features.

InfoCenter

The complete source for product documentaton, including tasks, reference, and conceptual information on product features and functions.

WebSphere MQ JMS Provider - Queue Conn. Factory


Home | Save | Preferences | Logout | Help |
WebSphere MQ JMS Provider
Scope
WebSphere MQ Queue Connection Factories

User ID: franck

S49CEPB
- ⊞ Servers
- ⊞ Applications
- ⊟ Resources
  - JDBC Providers
  - Generic JMS Providers
  - WebSphere JMS Provider
  - WebSphere MQ JMS Provider
  - Mail Providers
  - Resource Environment Providers
  - URL Providers
  - Resource Adapters
- ⊞ Security
- ⊞ Environment
- ⊞ System Administration
- ⊞ Troubleshooting

WebSphere MQ JMS Provider

A JMS provider enables asynchronous messaging based on the Java Messaging Service (JMS). It provides J2EE connection factories to create connections for specific JMS queue or topic destinations. WebSphere MQ JMS provider administrative objects are used to manage JMS resources for WebSphere MQ JMS provider.

Node=**S49NDPB**, Server=**PBOS001s**

○ Cell	S49CEPB	Use scope settings to limit the availability of resources to a particular cell, node, or server. When new items are created in this view, they will be created within the current scope.
● → Server	PBOS001s	
○ Node	S49NDPB	

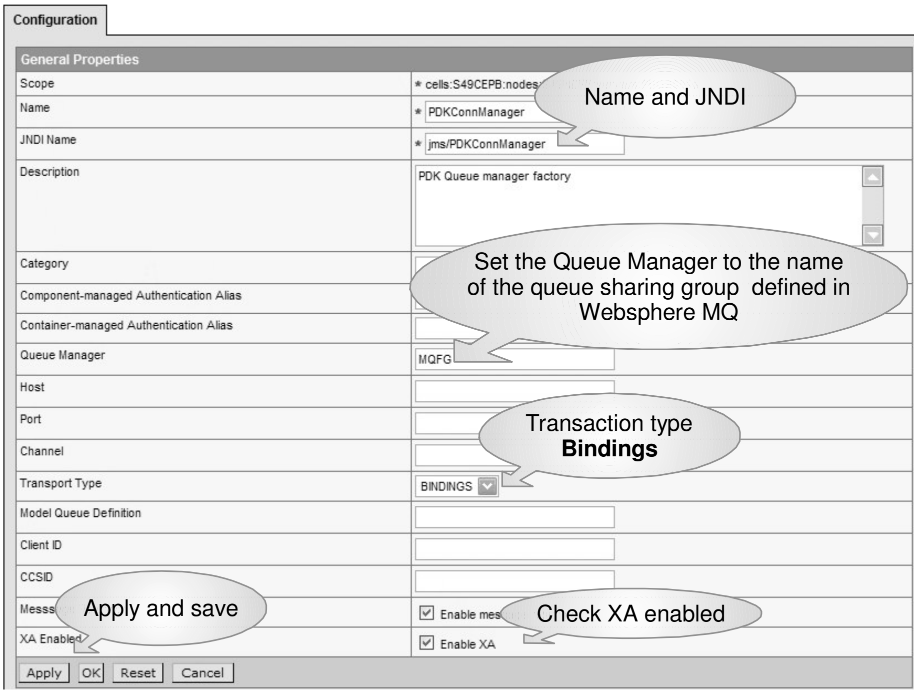
Apply

General Properties	
Scope	cells:S49CEPB:nodes:S49NDPB:servers:PBOS001s
Name	WebSphere MQ JMS Provider
Description	WebSphere MQ JMS Provider
Classpath	${MQJMS_LIB_ROOT}
Native Library Path	${MQJMS_LIB_ROOT}

Back

Additional Properties
WebSphere MQ Queue Connection Factories
WebSphere MQ Topic Connection Factories
WebSphere MQ Queue Destinations
WebSphere MQ Topic Destinations

WebSphere MQ JMS Provider - Queue Conn. Factory



Configuration

General Properties	
Scope	* cells:S49CEPB:nodes:
Name	* PDKConnManager
JNDI Name	* jms/PDKConnManager
Description	PDK Queue manager factory
Category	
Component-managed Authentication Alias	
Container-managed Authentication Alias	
Queue Manager	MQFG
Host	
Port	
Channel	
Transport Type	BINDINGS
Model Queue Definition	
Client ID	
CCSID	
Mess	☑ Enable mes
XA Enabled	☑ Enable XA

Name and JNDI

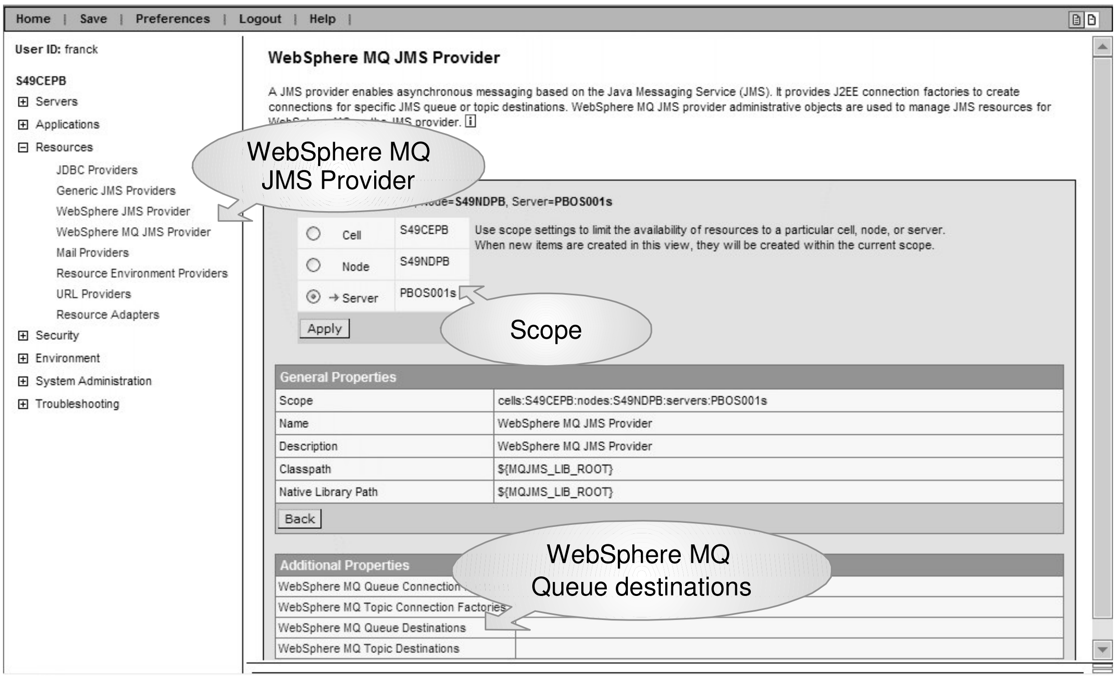
Set the Queue Manager to the name of the queue sharing group defined in Websphere MQ

Transaction type **Bindings**

Apply and save

Check XA enabled

WebSphere MQ JMS Provider - Queue destinations



WebSphere MQ JMS Provider

A JMS provider enables asynchronous messaging based on the Java Messaging Service (JMS). It provides J2EE connection factories to create connections for specific JMS queue or topic destinations. WebSphere MQ JMS provider administrative objects are used to manage JMS resources for the JMS provider.

ode=S49NDPB, Server=PBOS001s

○ Cell	S49CEPB	Use scope settings to limit the availability of resources to a particular cell, node, or server. When new items are created in this view, they will be created within the current scope.
○ Node	S49NDPB	
● → Server	PBOS001s	

General Properties	
Scope	cells:S49CEPB:nodes:S49NDPB:servers:PBOS001s
Name	WebSphere MQ JMS Provider
Description	WebSphere MQ JMS Provider
Classpath	${MQJMS_LIB_ROOT}
Native Library Path	${MQJMS_LIB_ROOT}

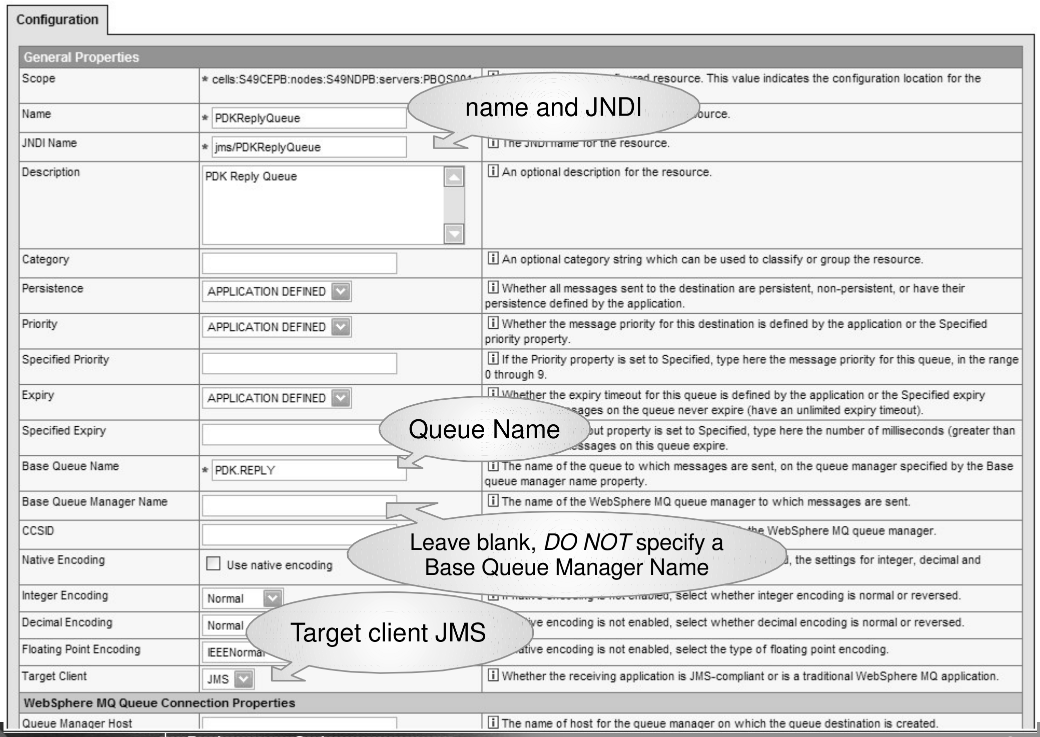
Additional Properties
WebSphere MQ Queue Connection
WebSphere MQ Topic Connection Factories
WebSphere MQ Queue Destinations
WebSphere MQ Topic Destinations

WebSphere MQ JMS Provider

Scope

WebSphere MQ Queue destinations

WebSphere MQ destination queue - reply queue



Configuration

General Properties		
Scope	* cells:S49CEPB:nodes:S49NDPB:servers:PBOS001	red resource. This value indicates the configuration location for the
Name	* PDKReplyQueue	ource.
JNDI Name	* jms/PDKReplyQueue	The JNDI name for the resource.
Description	PDK Reply Queue	An optional description for the resource.
Category		An optional category string which can be used to classify or group the resource.
Persistence	APPLICATION DEFINED	Whether all messages sent to the destination are persistent, non-persistent, or have their persistence defined by the application.
Priority	APPLICATION DEFINED	Whether the message priority for this destination is defined by the application or the Specified priority property.
Specified Priority		If the Priority property is set to Specified, type here the message priority for this queue, in the range 0 through 9.
Expiry	APPLICATION DEFINED	Whether the expiry timeout for this queue is defined by the application or the Specified expiry sages on the queue never expire (have an unlimited expiry timeout).
Specified Expiry		out property is set to Specified, type here the number of milliseconds (greater than essages on this queue expire.
Base Queue Name	* PDK.REPLY	The name of the queue to which messages are sent, on the queue manager specified by the Base queue manager name property.
Base Queue Manager Name		The name of the WebSphere MQ queue manager to which messages are sent.
CCSID		the WebSphere MQ queue manager.
Native Encoding	☐ Use native encoding	, the settings for integer, decimal and
Integer Encoding	Normal	select whether integer encoding is normal or reversed.
Decimal Encoding	Normal	ive encoding is not enabled, select whether decimal encoding is normal or reversed.
Floating Point Encoding	IEEENormal	ative encoding is not enabled, select the type of floating point encoding.
Target Client	JMS	Whether the receiving application is JMS-compliant or is a traditional WebSphere MQ application.
WebSphere MQ Queue Connection Properties		
Queue Manager Host		The name of host for the queue manager on which the queue destination is created.

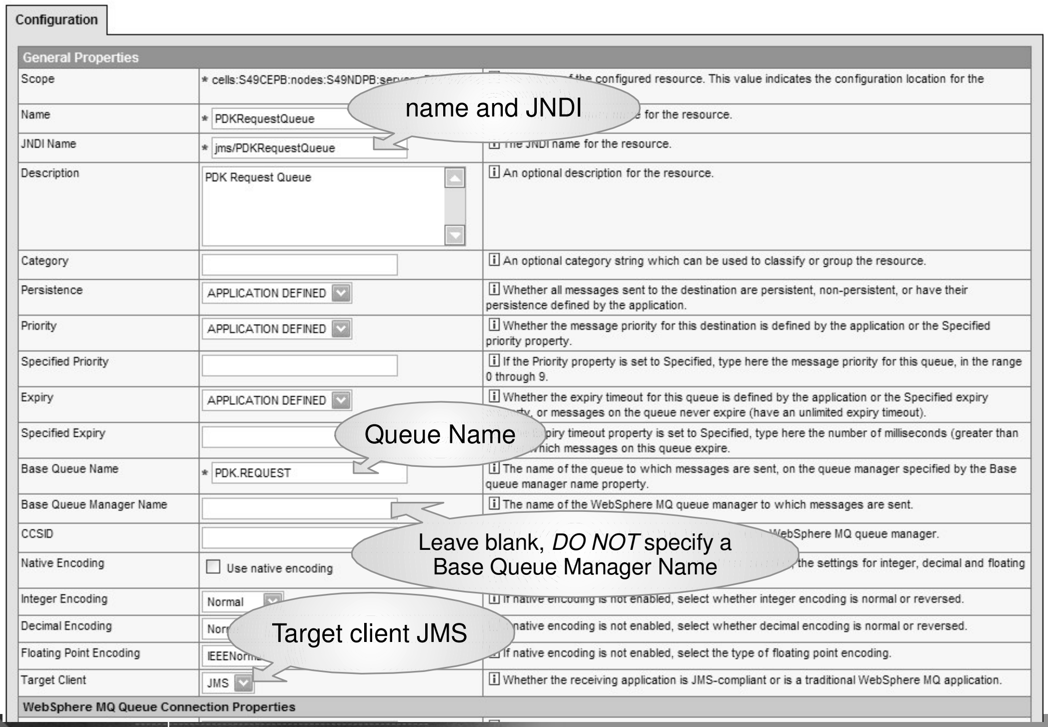
name and JNDI

Queue Name

Leave blank, *DO NOT* specify a Base Queue Manager Name

Target client JMS

WebSphere MQ destination queue - request queue



Configuration

General Properties		
Scope	* cells:S49CEPB:nodes:S49NDPB:serv	f the configured resource. This value indicates the configuration location for the
Name	* PDKRequestQueue	for the resource.
JNDI Name	* jms/PDKRequestQueue	The JNDI name for the resource.
Description	PDK Request Queue	An optional description for the resource.
Category		An optional category string which can be used to classify or group the resource.
Persistence	APPLICATION DEFINED	Whether all messages sent to the destination are persistent, non-persistent, or have their persistence defined by the application.
Priority	APPLICATION DEFINED	Whether the message priority for this destination is defined by the application or the Specified priority property.
Specified Priority		If the Priority property is set to Specified, type here the message priority for this queue, in the range 0 through 9.
Expiry	APPLICATION DEFINED	Whether the expiry timeout for this queue is defined by the application or the Specified expiry rty, or messages on the queue never expire (have an unlimited expiry timeout).
Specified Expiry		piry timeout property is set to Specified, type here the number of milliseconds (greater than hich messages on this queue expire.
Base Queue Name	* PDK.REQUEST	The name of the queue to which messages are sent, on the queue manager specified by the Base queue manager name property.
Base Queue Manager Name		The name of the WebSphere MQ queue manager to which messages are sent.
CCSID		WebSphere MQ queue manager.
Native Encoding	☐ Use native encoding	the settings for integer, decimal and floating
Integer Encoding	Normal	If native encoding is not enabled, select whether integer encoding is normal or reversed.
Decimal Encoding	Nor	native encoding is not enabled, select whether decimal encoding is normal or reversed.
Floating Point Encoding	IEEENor	If native encoding is not enabled, select the type of floating point encoding.
Target Client	JMS	Whether the receiving application is JMS-compliant or is a traditional WebSphere MQ application.
WebSphere MQ Queue Connection Properties		

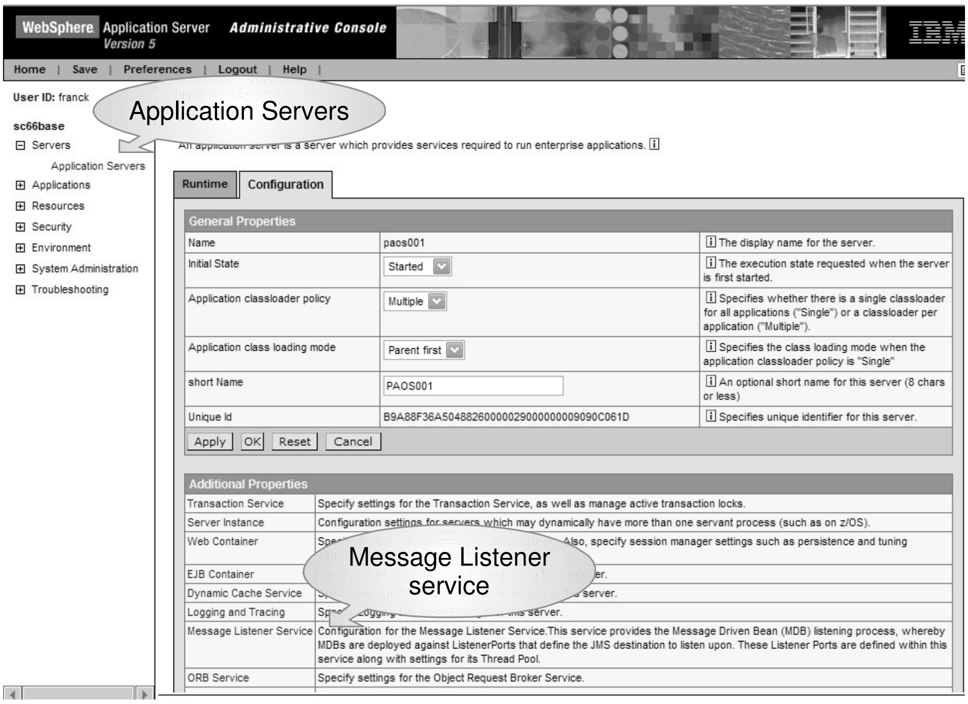
name and JNDI

Queue Name

Leave blank, *DO NOT* specify a Base Queue Manager Name

Target client JMS

WebSphere MQ - Message Listener Service


WebSphere Application Server Version 5 Administrative Console
Home | Save | Preferences | Logout | Help |
Application Servers
Message Listener service

User ID: franck

sc66base
- ⊟ Servers
  - Application Servers
- ⊞ Applications
- ⊞ Resources
- ⊞ Security
- ⊞ Environment
- ⊞ System Administration
- ⊞ Troubleshooting

An application server is a server which provides services required to run enterprise applications.

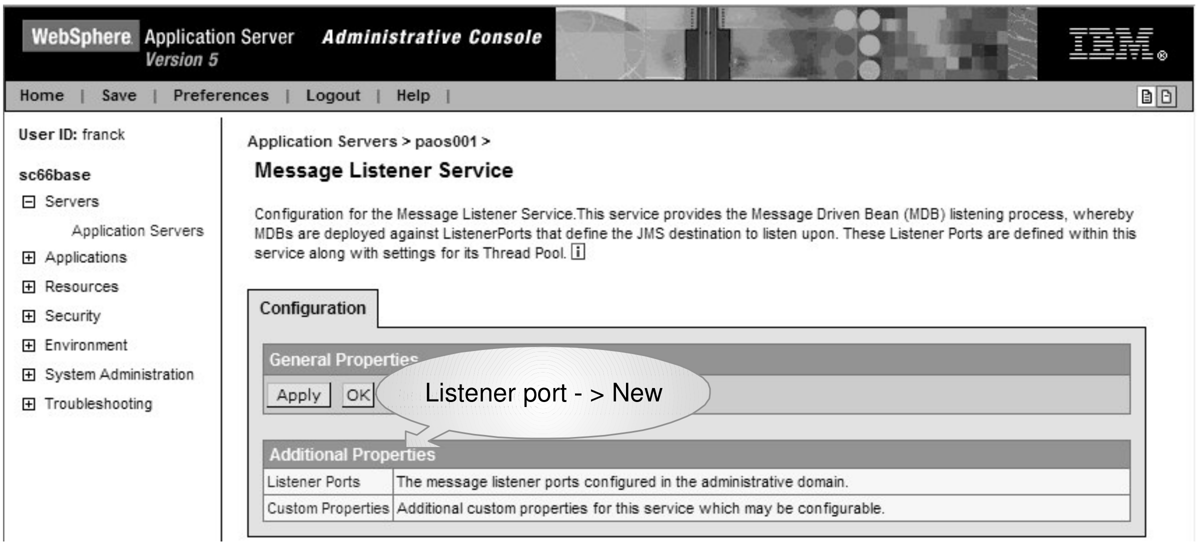
Runtime | Configuration

General Properties		
Name	paos001	The display name for the server.
Initial State	Started	The execution state requested when the server is first started.
Application classloader policy	Multiple	Specifies whether there is a single classloader for all applications ("Single") or a classloader per application ("Multiple").
Application class loading mode	Parent first	Specifies the class loading mode when the application classloader policy is "Single"
short Name	PAOS001	An optional short name for this server (8 chars or less)
Unique Id	B9A88F36A5048826000002900000000909 0C061D	Specifies unique identifier for this server.

Apply | OK | Reset | Cancel

Additional Properties	
Transaction Service	Specify settings for the Transaction Service, as well as manage active transaction locks.
Server Instance	Configuration settings for servers which may dynamically have more than one servant process (such as on z/OS).
Web Container	Spe Also, specify session manager settings such as persistence and tuning
EJB Container	er.
Dynamic Cache Service	S server.
Logging and Tracing	Sp Logging this server.
Message Listener Service	Configuration for the Message Listener Service.This service provides the Message Driven Bean (MDB) listening process, whereby MDBs are deployed against ListenerPorts that define the JMS destination to listen upon. These Listener Ports are defined within this service along with settings for its Thread Pool.
ORB Service	Specify settings for the Object Request Broker Service.

WebSphere MQ - Message Listener Service


WebSphere Application Server Version 5 Administrative Console
Home | Save | Preferences | Logout | Help |
Listener port - > New

User ID: franck

sc66base
- ⊟ Servers
  - Application Servers
- ⊞ Applications
- ⊞ Resources
- ⊞ Security
- ⊞ Environment
- ⊞ System Administration
- ⊞ Troubleshooting

Application Servers > paos001 >

Message Listener Service

Configuration for the Message Listener Service.This service provides the Message Driven Bean (MDB) listening process, whereby MDBs are deployed against ListenerPorts that define the JMS destination to listen upon. These Listener Ports are defined within this service along with settings for its Thread Pool.

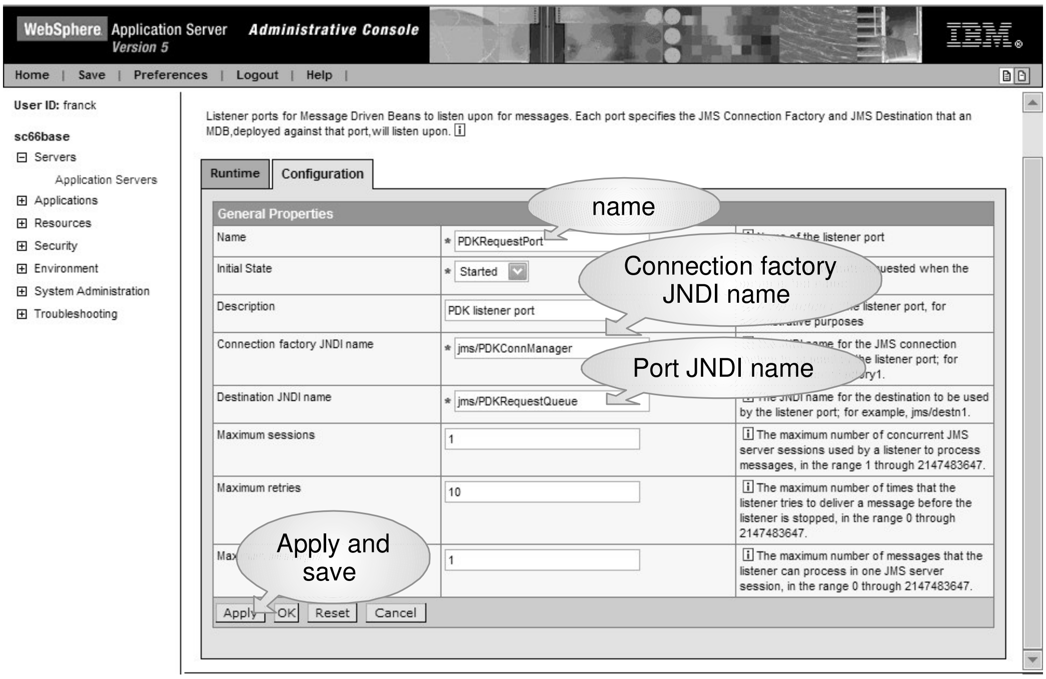
Configuration

General Properties

Apply | OK

Additional Properties	
Listener Ports	The message listener ports configured in the administrative domain.
Custom Properties	Additional custom properties for this service which may be configurable.

WebSphere MQ - Message Listener Service



Listener ports for Message Driven Beans to listen upon for messages. Each port specifies the JMS Connection Factory and JMS Destination that an MDB,deployed against that port,will listen upon.

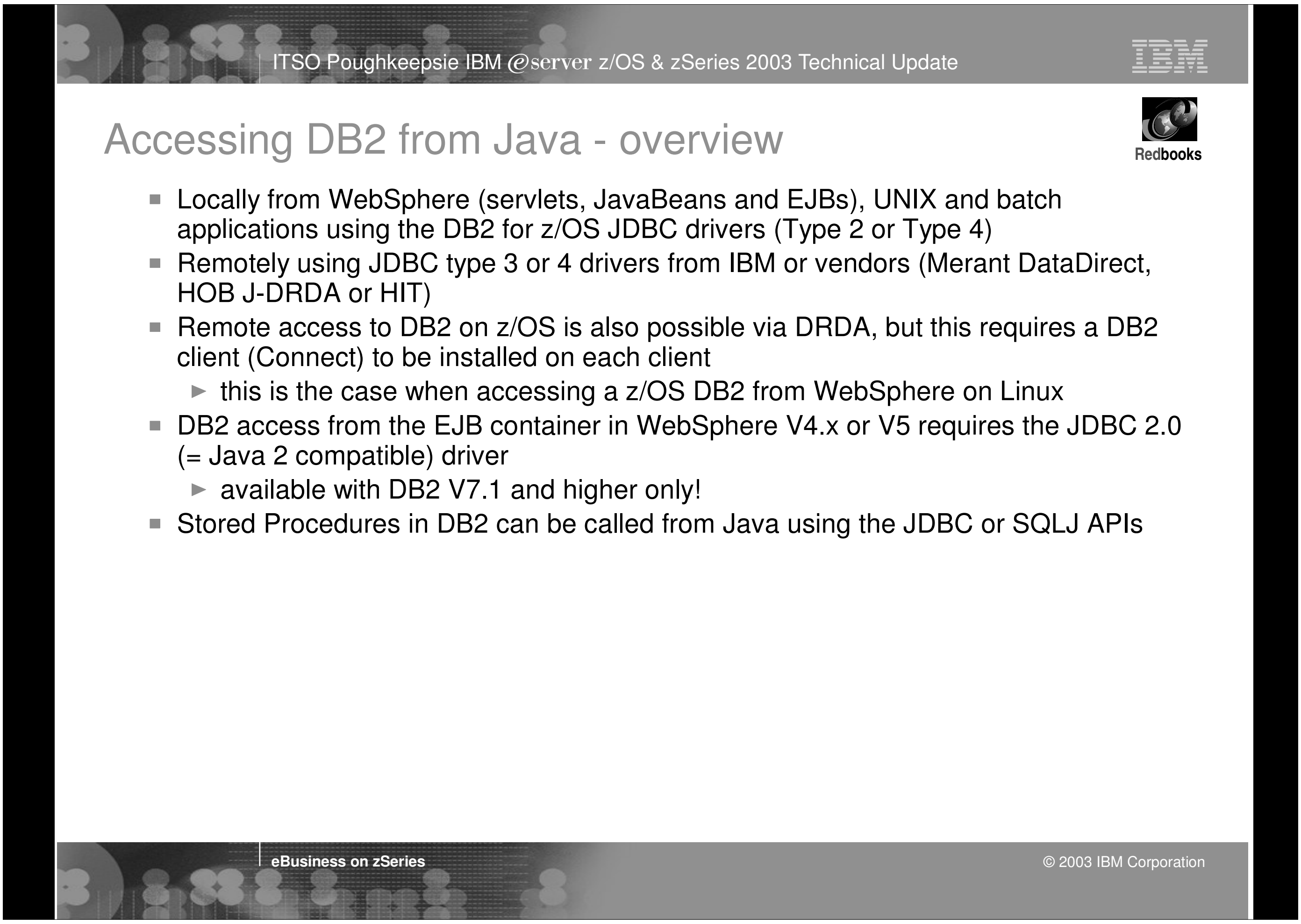
General Properties

Name	* PDKRequestPort	… of the listener port
Initial State	* Started	…uested when the
Description	PDK listener port	…e listener port, for …trative purposes
Connection factory JNDI name	* jms/PDKConnManager	…ame for the JMS connection …e listener port; for …ry1.
Destination JNDI name	* jms/PDKRequestQueue	…e JNDI name for the destination to be used by the listener port; for example, jms/destn1.
Maximum sessions	1	The maximum number of concurrent JMS server sessions used by a listener to process messages, in the range 1 through 2147483647.
Maximum retries	10	The maximum number of times that the listener tries to deliver a message before the listener is stopped, in the range 0 through 2147483647.
Max…	1	The maximum number of messages that the listener can process in one JMS server session, in the range 0 through 2147483647.

Apply | OK | Reset | Cancel

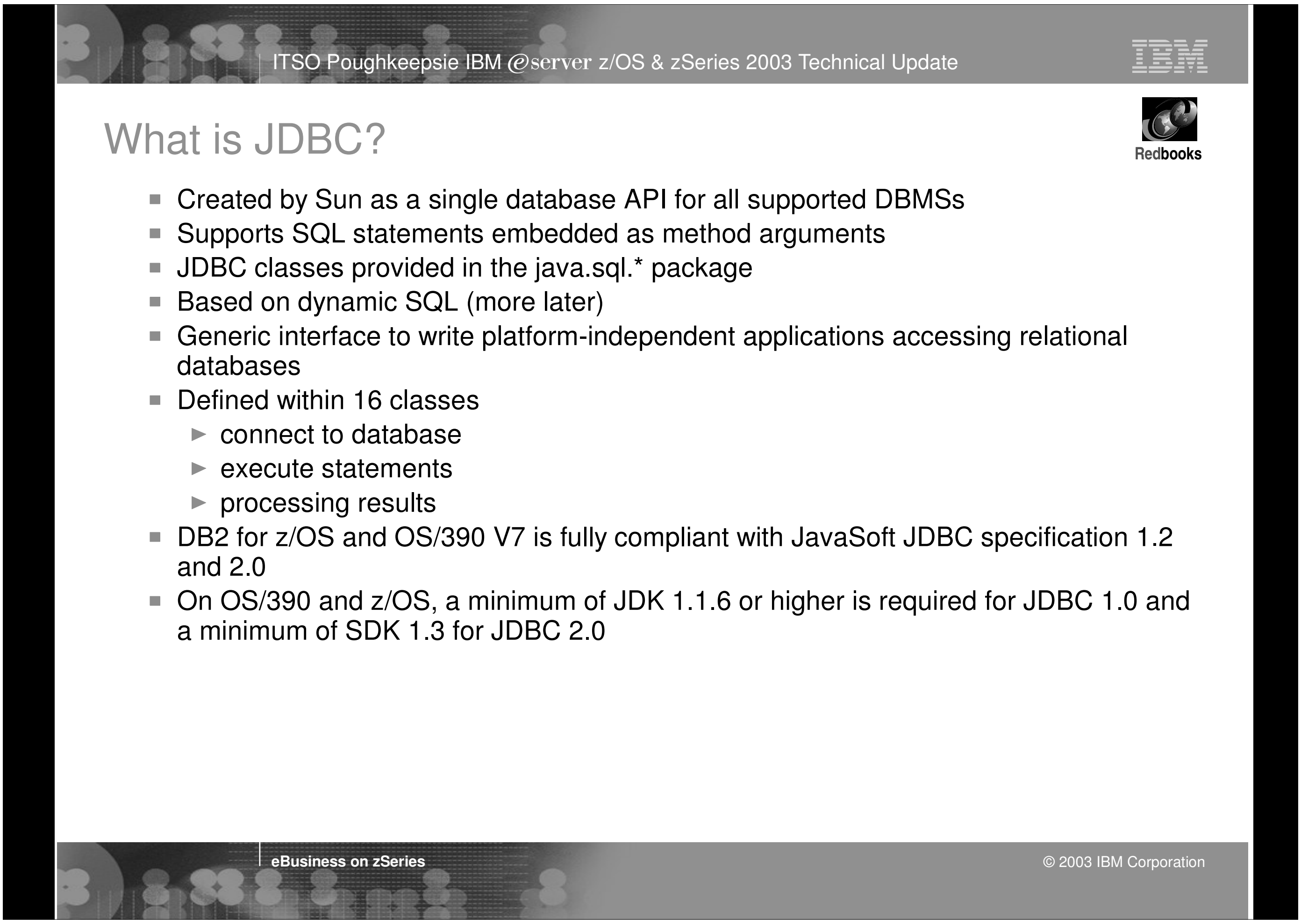
Agenda

- Overview Java Connector Architecture
- CICS Transaction Gateway
- IMS Connect
- Java Message Service
- ➡ Connecting to databases
  - ➡ JDBC and SQLJ overview
  - Best practices
  - How to information



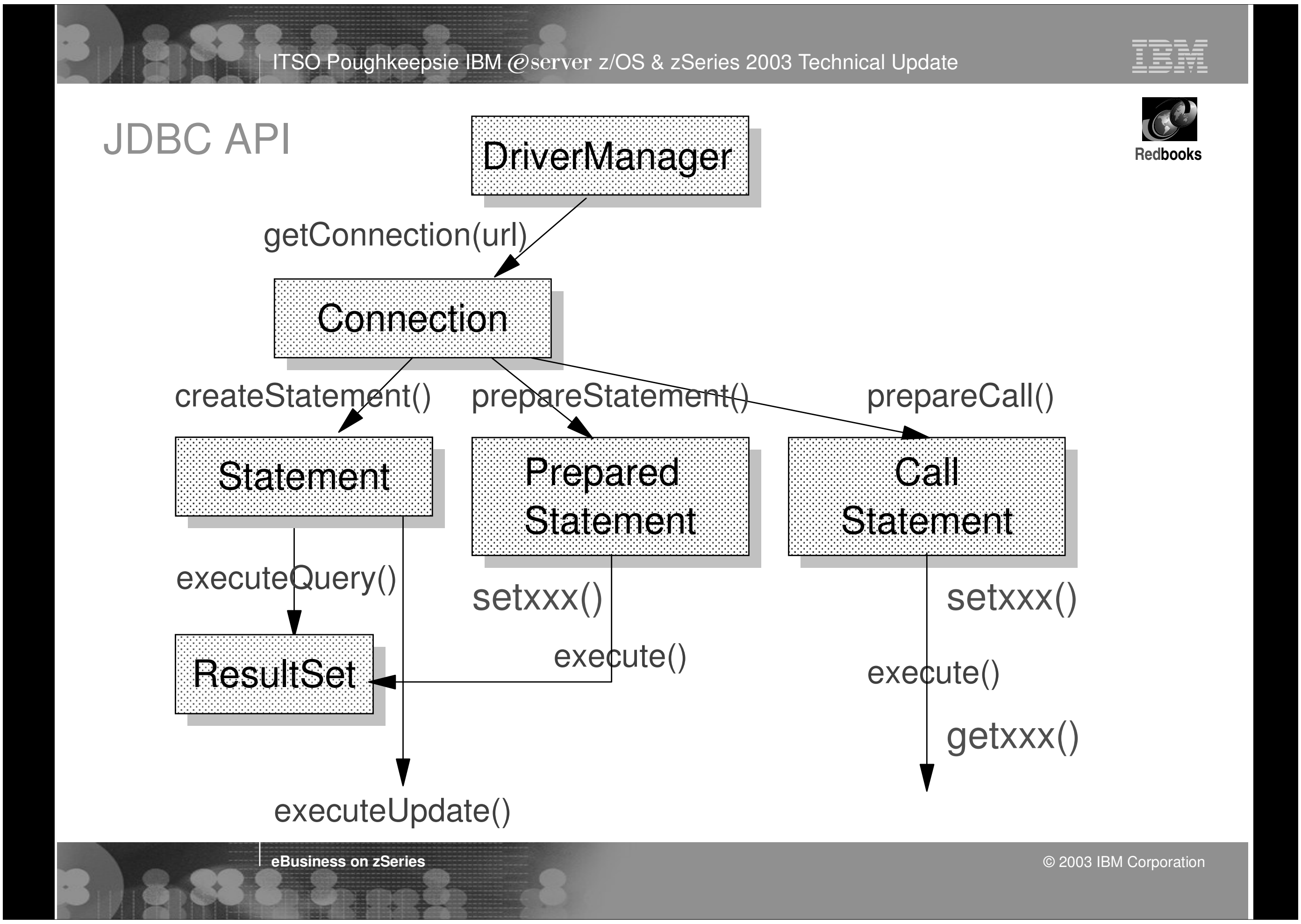
Accessing DB2 from Java - overview

- Locally from WebSphere (servlets, JavaBeans and EJBs), UNIX and batch applications using the DB2 for z/OS JDBC drivers (Type 2 or Type 4)
- Remotely using JDBC type 3 or 4 drivers from IBM or vendors (Merant DataDirect, HOB J-DRDA or HIT)
- Remote access to DB2 on z/OS is also possible via DRDA, but this requires a DB2 client (Connect) to be installed on each client
  - ► this is the case when accessing a z/OS DB2 from WebSphere on Linux
- DB2 access from the EJB container in WebSphere V4.x or V5 requires the JDBC 2.0 (= Java 2 compatible) driver
  - ► available with DB2 V7.1 and higher only!
- Stored Procedures in DB2 can be called from Java using the JDBC or SQLJ APIs



What is JDBC?

- Created by Sun as a single database API for all supported DBMSs
- Supports SQL statements embedded as method arguments
- JDBC classes provided in the java.sql.* package
- Based on dynamic SQL (more later)
- Generic interface to write platform-independent applications accessing relational databases
- Defined within 16 classes
  - ► connect to database
  - ► execute statements
  - ► processing results
- DB2 for z/OS and OS/390 V7 is fully compliant with JavaSoft JDBC specification 1.2 and 2.0
- On OS/390 and z/OS, a minimum of JDK 1.1.6 or higher is required for JDBC 1.0 and a minimum of SDK 1.3 for JDBC 2.0



JDBC API

DriverManager

getConnection(url)

Connection

createStatement()

prepareStatement()

prepareCall()

Statement

Prepared Statement

Call Statement

executeQuery()

setxxx()

setxxx()

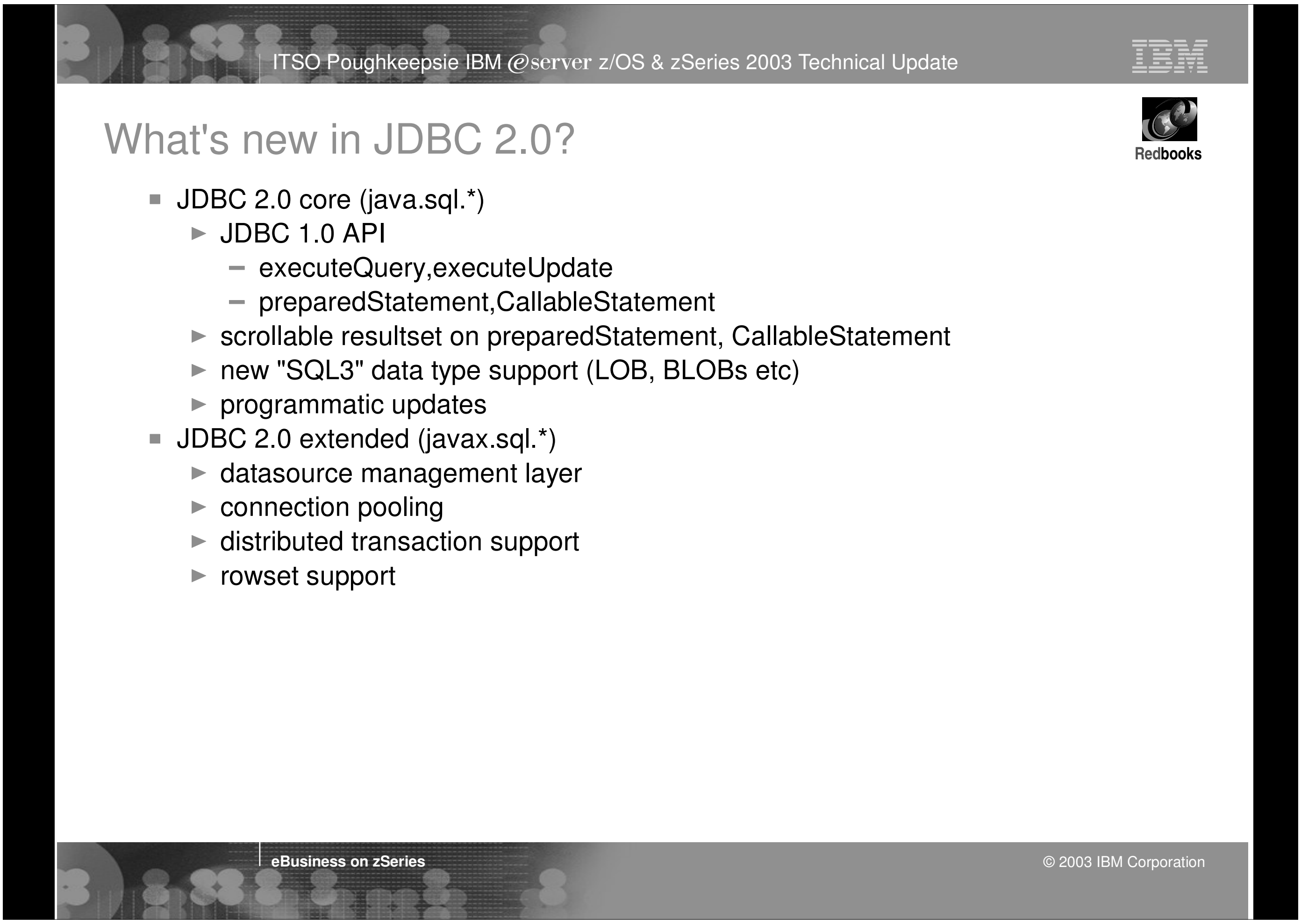
ResultSet

execute()

execute()

getxxx()

executeUpdate()

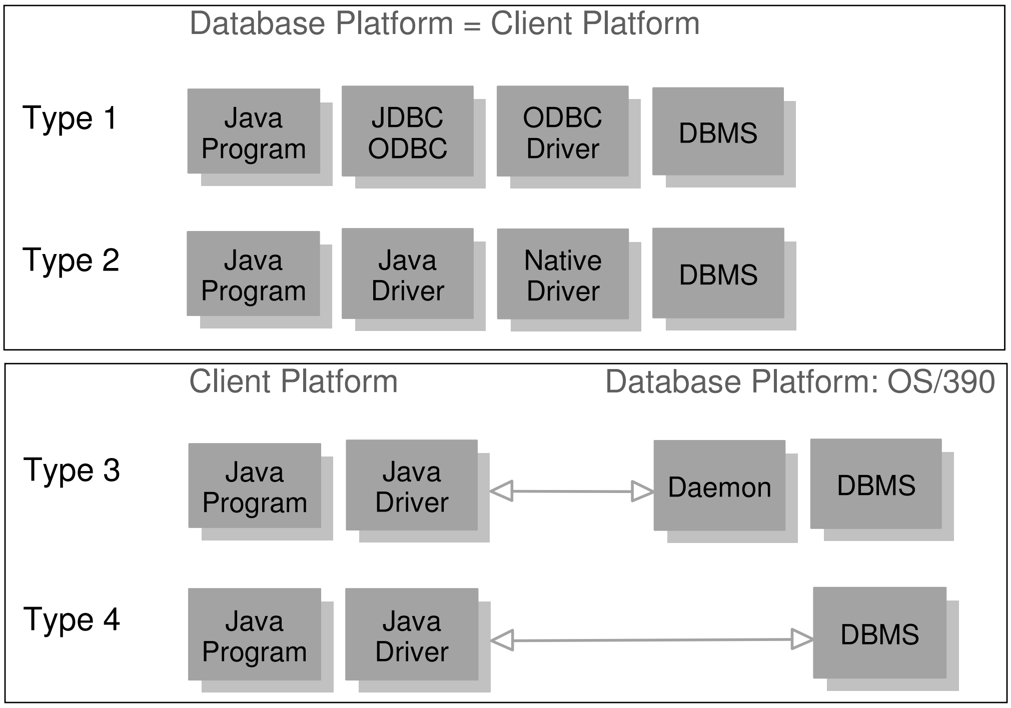


What's new in JDBC 2.0?

- JDBC 2.0 core (java.sql.*)
  - JDBC 1.0 API
    - executeQuery,executeUpdate
    - preparedStatement,CallableStatement
  - scrollable resultset on preparedStatement, CallableStatement
  - new "SQL3" data type support (LOB, BLOBs etc)
  - programmatic updates
- JDBC 2.0 extended (javax.sql.*)
  - datasource management layer
  - connection pooling
  - distributed transaction support
  - rowset support

Redbooks

JDBC - driver types



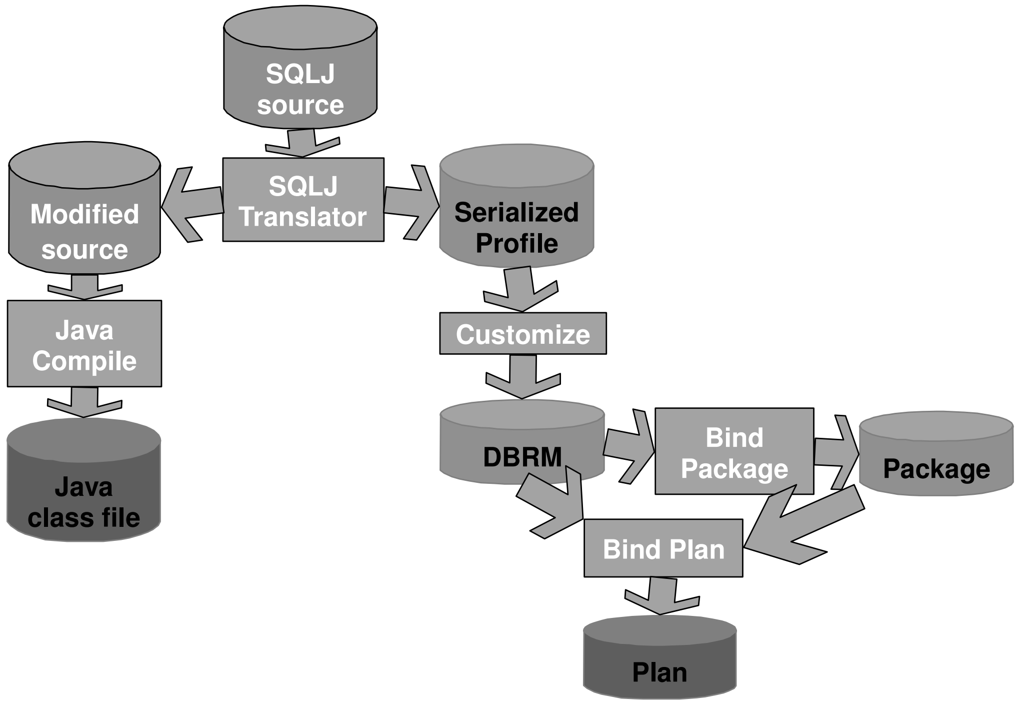
Redbooks

What is SQLJ?

- Wide DBMS vendor acceptance
  - IBM, Oracle, Sybase, Informix, Compaq (Tandem)
  - SQLJ has been accepted by ANSI and ISO
  - IBM and Oracle to push for SQLJ to be included in J2EE and JDK
- Support for embedded static SQL in java applications and servlets, but also works from session beans and BMP entity beans
- SQLJ specifications consist of three parts:
  - Database Languages - SQL - Part 0: Object Language Bindings
  - Database Languages - SQL - Part 1: SQL Routines using the Java Programming Language
  - Database Languages - SQL - Part 2: Types using the Java Programming Language
- DB2 for OS/390 and z/OS implementation of SQLJ supports:
  - Part 0
  - Part 1: invoke a java static method as a stored procedure

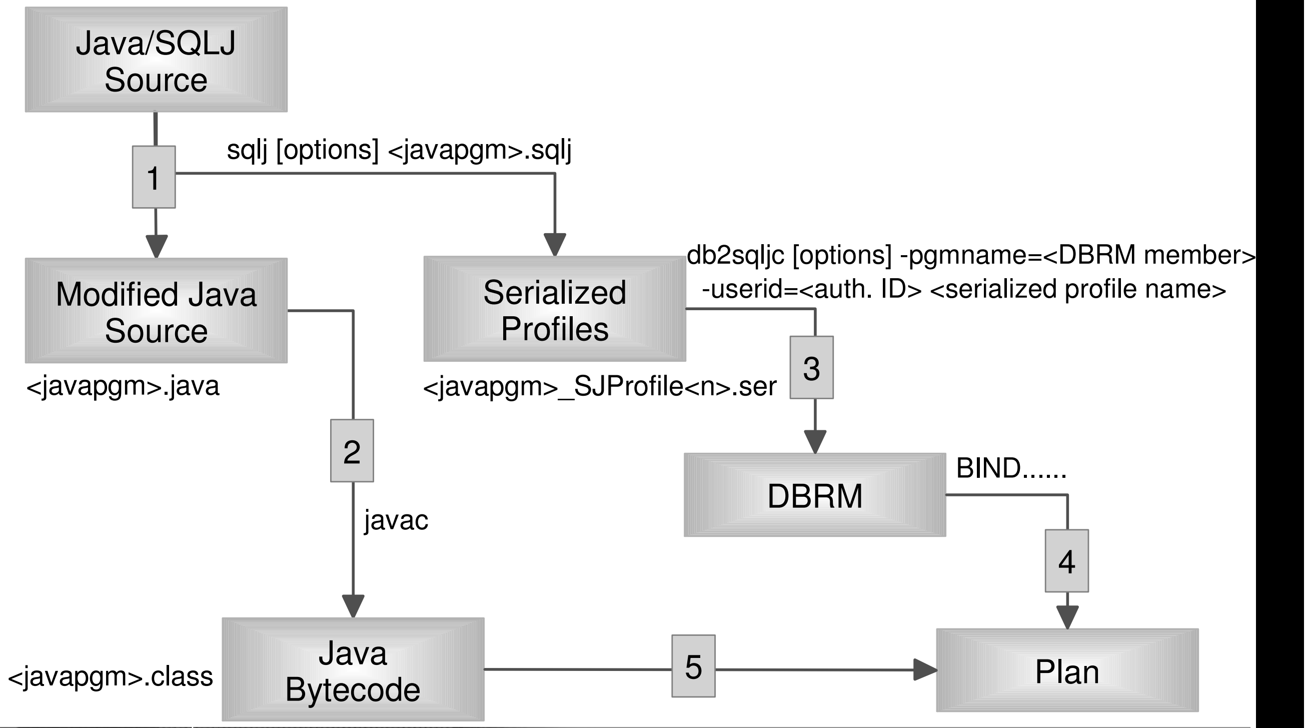
Redbooks

SQLJ preparation steps



Redbooks

Preparing an SQLJ Program for OS/390



sqlj [options] <javapgm>.sqlj

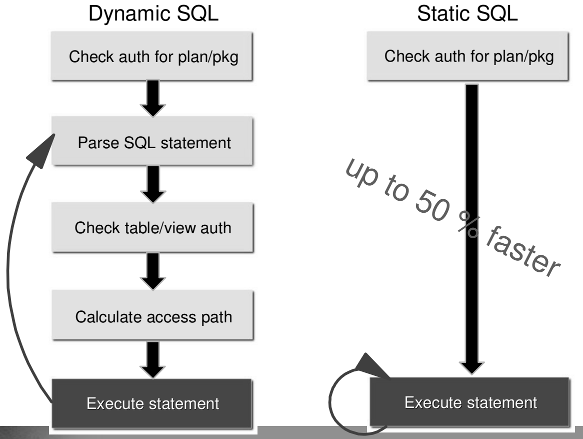
db2sqljc [options] -pgmname=<DBRM member>
-userid=<auth. ID> <serialized profile name>

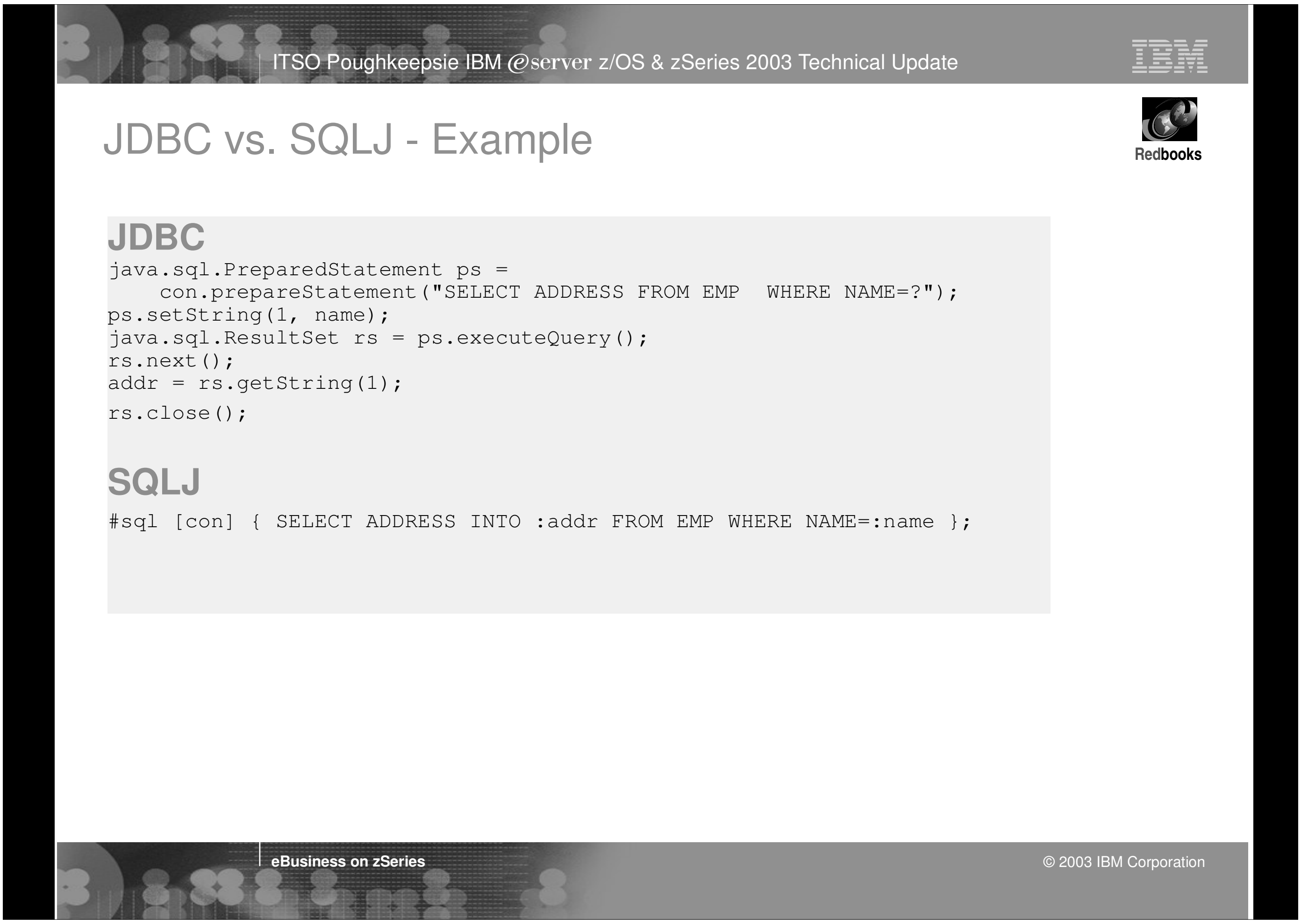
JDBC vs. SQLJ

- Reasons to use SQLJ:
  - more concise and less complex to code relative to JDBC
  - for DB2 (with optional customization step)
    - *reduced CPU resource consumption* (not prepared at run-time)
    - users can be authorized to run programs, not access tables
  - optional SQL checking during program preparation
    - syntax
    - type mappings
  - SQLJ is the strategic way to access relational data from Java
- Reasons you might not use SQLJ:
  - lack of tooling support and/or more steps in application build process
  - need flexibility in some cases to dynamically build SQL request at run time
- SQLJ and JDBC interoperability
  - can 'mix and match' SQLJ and JDBC in the same application if you need to selectively use dynamic SQL
  - SQLJ and JDBC can share the same JDBC connection
  - JDBC result sets can be turned into SQLJ iterators, and vice versa

Static SQL is usually faster...

- Static SQL
  - syntax of embedded SQL is fully known at precompile time
  - executable form of statement created and stored in the package in the database before runtime
  - authorization of the person binding is used -> end user does not require direct privileges to execute statements in the package
- Dynamic SQL: syntax is not known until runtime





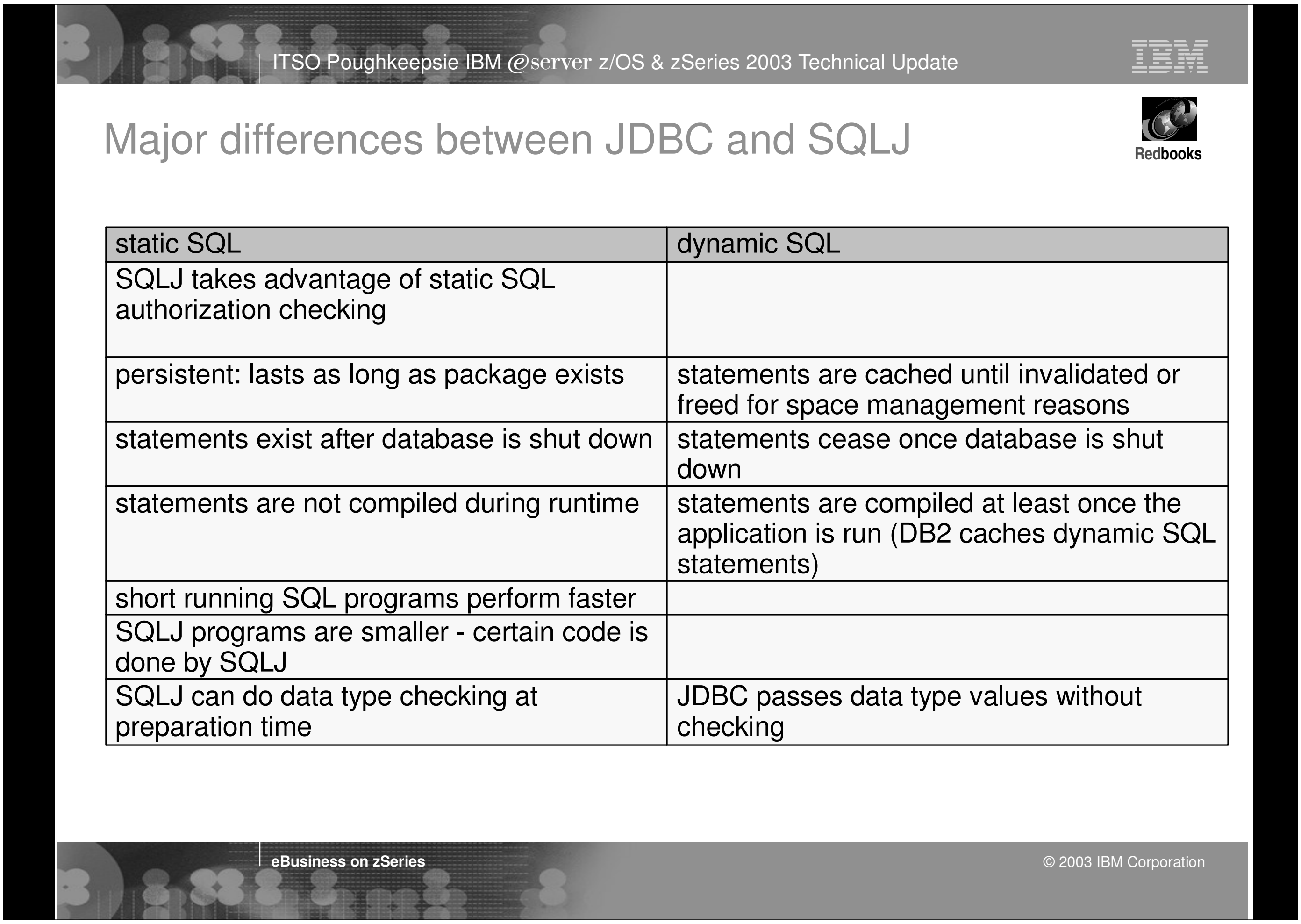
JDBC vs. SQLJ - Example

JDBC

```
java.sql.PreparedStatement ps =
    con.prepareStatement("SELECT ADDRESS FROM EMP  WHERE NAME=?");
ps.setString(1, name);
java.sql.ResultSet rs = ps.executeQuery();
rs.next();
addr = rs.getString(1);
rs.close();
```

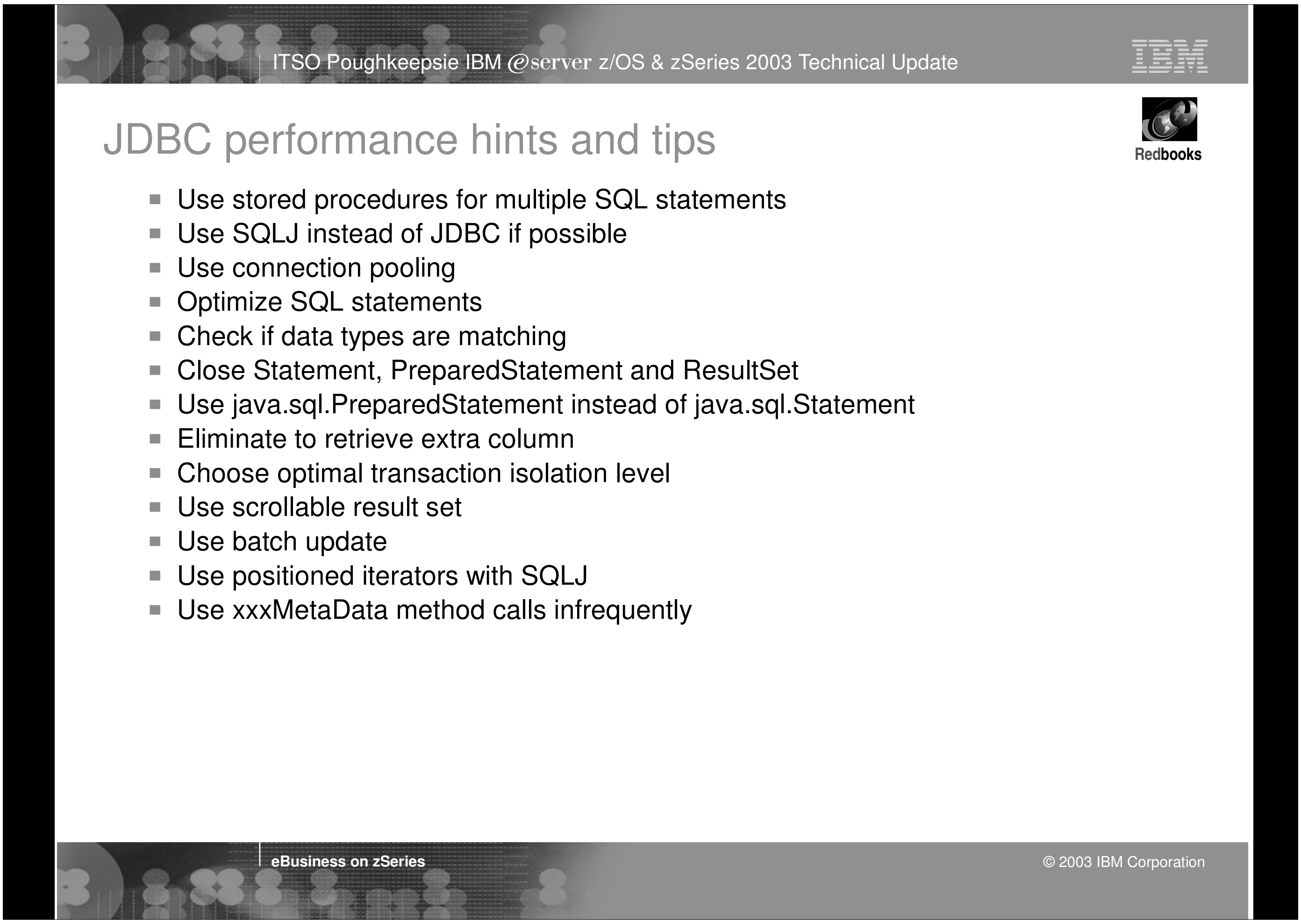
SQLJ

```
#sql [con] { SELECT ADDRESS INTO :addr FROM EMP WHERE NAME=:name };
```



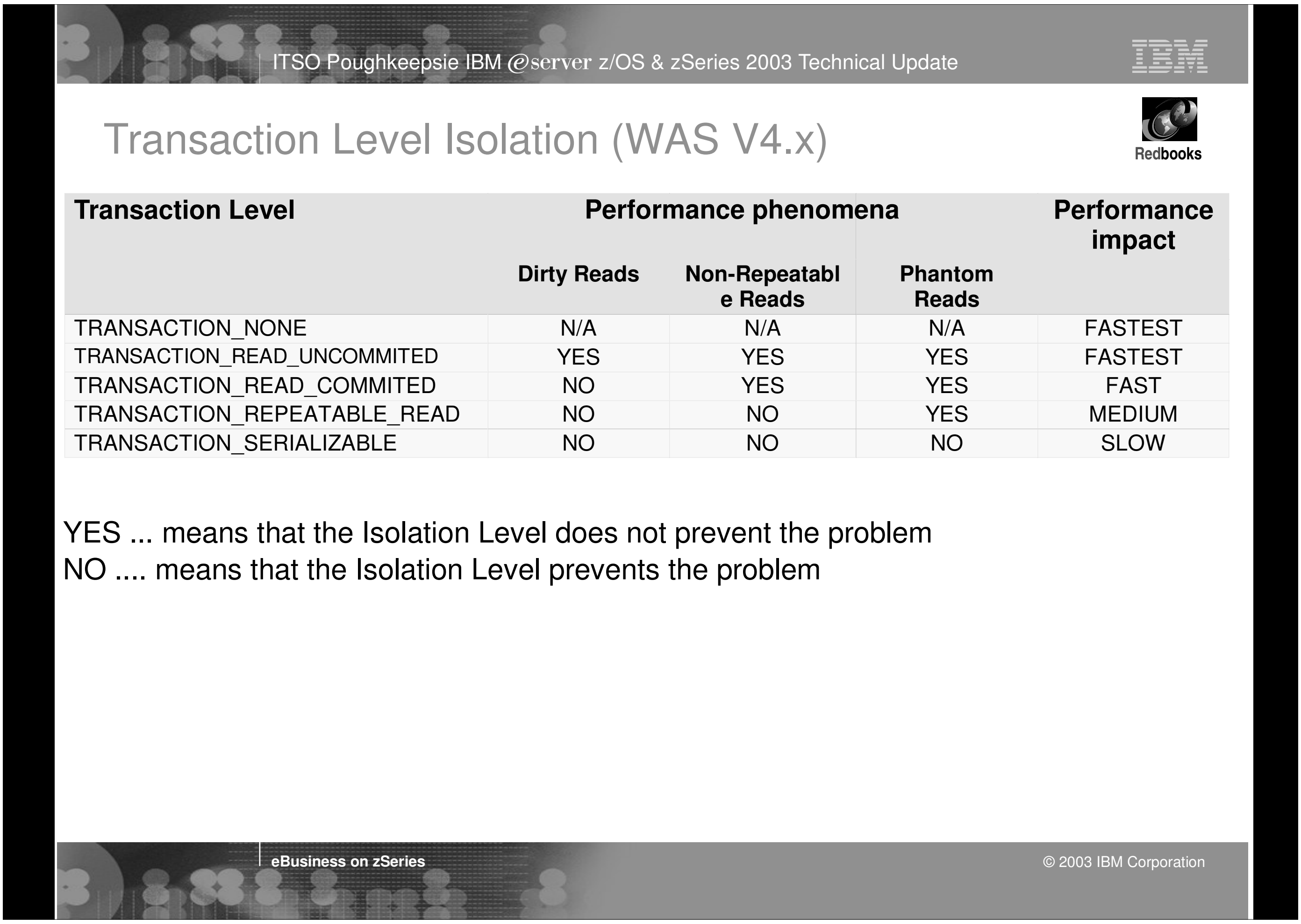
Major differences between JDBC and SQLJ

static SQL	dynamic SQL
SQLJ takes advantage of static SQL authorization checking	
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
statements are not compiled during runtime	statements are compiled at least once the application is run (DB2 caches dynamic SQL statements)
short running SQL programs perform faster	
SQLJ programs are smaller - certain code is done by SQLJ	
SQLJ can do data type checking at preparation time	JDBC passes data type values without checking



JDBC performance hints and tips

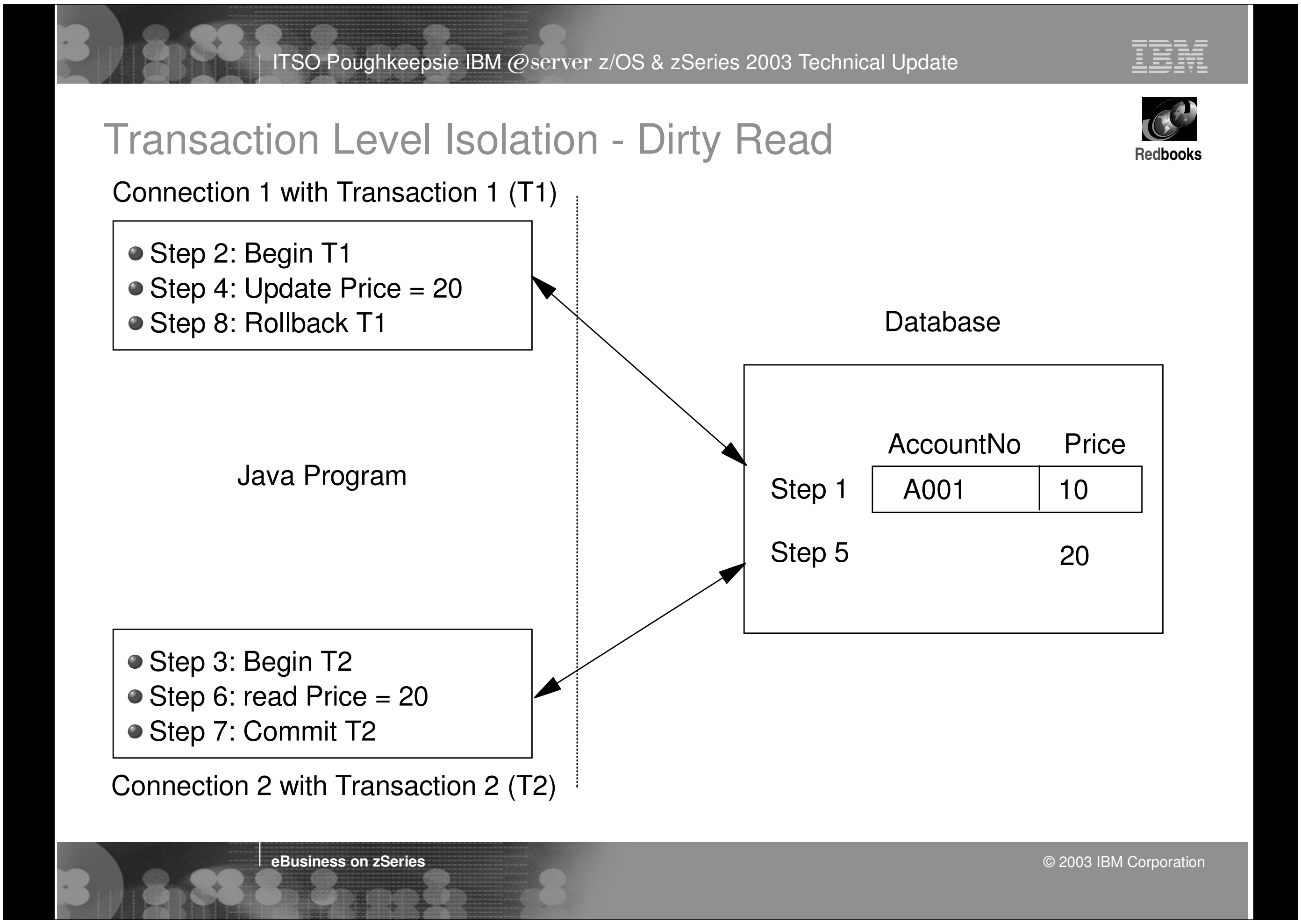
- Use stored procedures for multiple SQL statements
- Use SQLJ instead of JDBC if possible
- Use connection pooling
- Optimize SQL statements
- Check if data types are matching
- Close Statement, PreparedStatement and ResultSet
- Use java.sql.PreparedStatement instead of java.sql.Statement
- Eliminate to retrieve extra column
- Choose optimal transaction isolation level
- Use scrollable result set
- Use batch update
- Use positioned iterators with SQLJ
- Use xxxMetaData method calls infrequently



Transaction Level Isolation (WAS V4.x)

Transaction Level	Performance phenomena			Performance impact
	Dirty Reads	Non-Repeatable Reads	Phantom Reads	
TRANSACTION_NONE	N/A	N/A	N/A	FASTEST
TRANSACTION_READ_UNCOMMITED	YES	YES	YES	FASTEST
TRANSACTION_READ_COMMITED	NO	YES	YES	FAST
TRANSACTION_REPEATABLE_READ	NO	NO	YES	MEDIUM
TRANSACTION_SERIALIZABLE	NO	NO	NO	SLOW

YES ... means that the Isolation Level does not prevent the problem
NO .... means that the Isolation Level prevents the problem



ITSO Poughkeepsie IBM @server z/OS & zSeries 2003 Technical Update

Transaction Level Isolation - Dirty Read

Connection 1 with Transaction 1 (T1)

- Step 2: Begin T1
- Step 4: Update Price = 20
- Step 8: Rollback T1

Java Program

- Step 3: Begin T2
- Step 6: read Price = 20
- Step 7: Commit T2

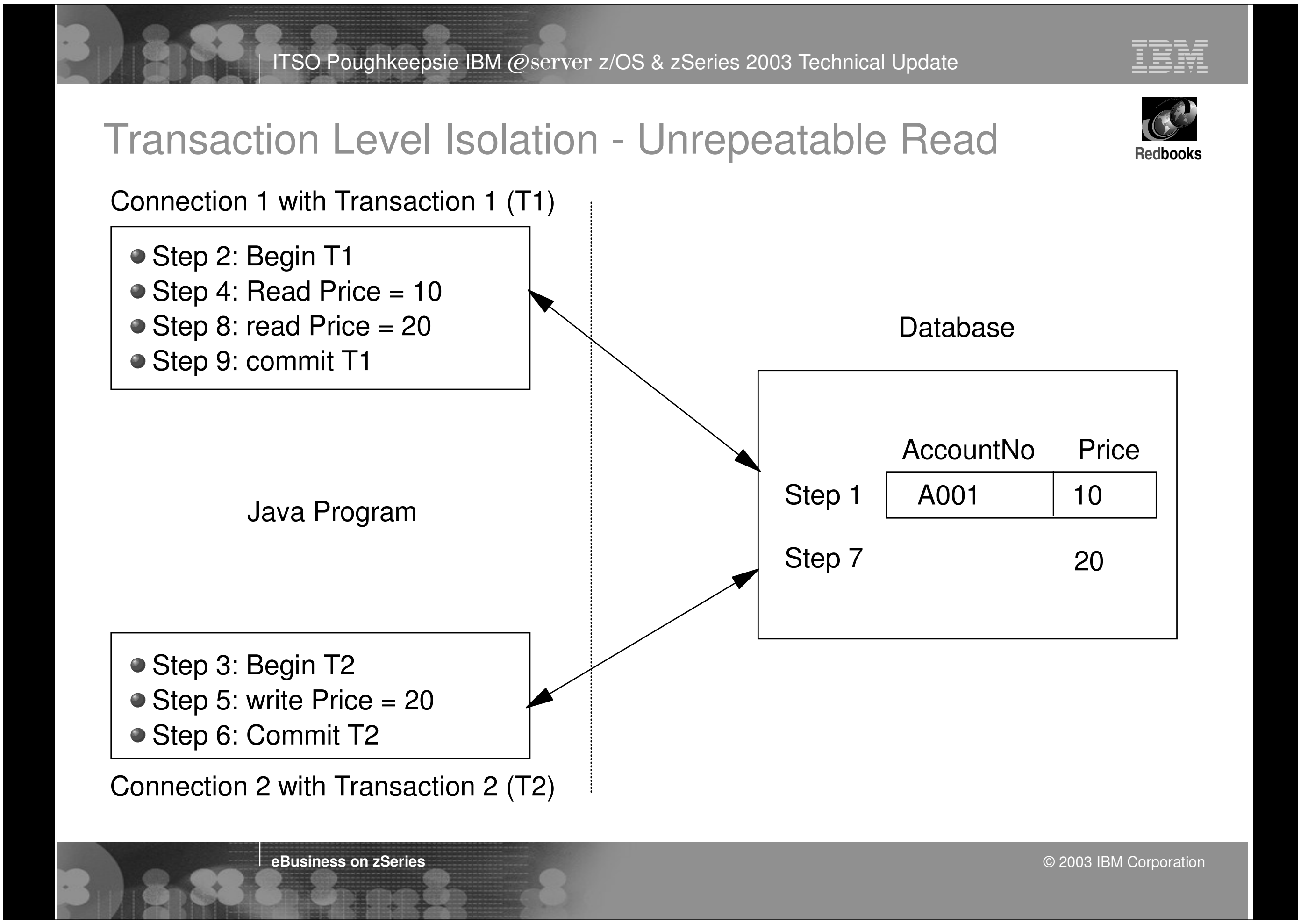
Connection 2 with Transaction 2 (T2)

Database

	AccountNo	Price
Step 1	A001	10
Step 5		20

eBusiness on zSeries

© 2003 IBM Corporation



ITSO Poughkeepsie IBM @server z/OS & zSeries 2003 Technical Update

Transaction Level Isolation - Unrepeatable Read

Connection 1 with Transaction 1 (T1)

- Step 2: Begin T1
- Step 4: Read Price = 10
- Step 8: read Price = 20
- Step 9: commit T1

Java Program

- Step 3: Begin T2
- Step 5: write Price = 20
- Step 6: Commit T2

Connection 2 with Transaction 2 (T2)

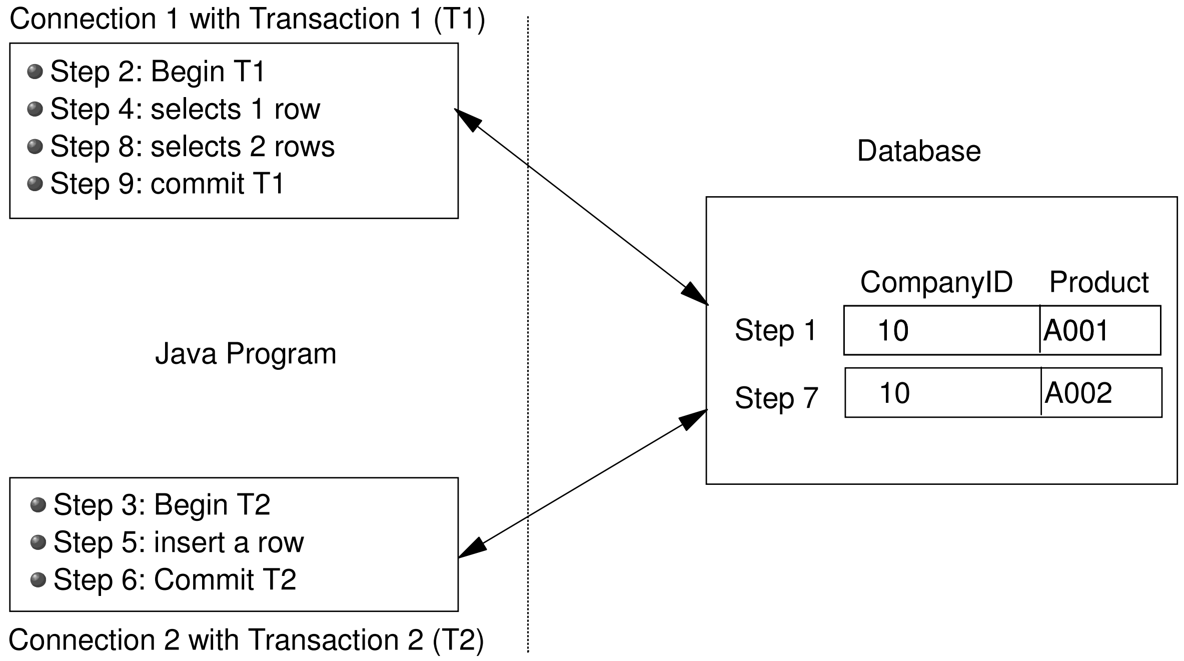
Database

	AccountNo	Price
Step 1	A001	10
Step 7		20

eBusiness on zSeries

© 2003 IBM Corporation

Transaction Level Isolation - Phantom Read



Connection 1 with Transaction 1 (T1)

- Step 2: Begin T1
- Step 4: selects 1 row
- Step 8: selects 2 rows
- Step 9: commit T1

Java Program

- Step 3: Begin T2
- Step 5: insert a row
- Step 6: Commit T2

Connection 2 with Transaction 2 (T2)

Database

	CompanyID	Product
Step 1	10	A001
Step 7	10	A002

Agenda

- Overview Java Connector Architecture
- CICS Transaction Gateway
- IMS Connect
- Java Message Service
- ➡ Connecting to databases
  - JDBC and SQLJ overview
  - Best practices
  - ➡ How to information

Steps to set up JDBC for DB2 on z/OS

- Update the procedures for the servant address spaces to include the DB2 libraries (optional)
- Create/update the db2sqljjdbc.properties file to indicate the location of the DSNJDBC_JDBCProfile.ser file
- Create/update WebSphere environment variables so the WebSphere server can locate the DB2 home directory and the JDBC driver can locate the db2sqljjdbc.properties file.
- Define J2C Authentication Aliases to provide userid with password to be used when connecting to DB2 (optional)
- Define a JDBC Provider for DB2 for z/OS access.
- Define a DataSource specifying the JAAS aliases for component and container connections (optional).
- Update the custom properties to indicate the DB2 location name to be associated with this DataSource.

Please refer to the document Using DB2 for z/OS in WebSphere for z/OS Version 5 published by Washington System Center.

http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD101072

Important info for the MVS syspro...

If your installation does not have

DB2 SDSNEXIT, SDSNLOAD and SDSNLOD2 in the linklist,

update the STEPLIB DD concatenation for the servant address space

```
EDIT       SYS1.PROCLIB(PBO5ASRZ) - 01.02                  Columns 00001 00072
Command ===>                                                  Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001 //*
000002 //* Output DDs
000004 //CEEDUMP   DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
000005 //SYSOUT    DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
000006 //SYSPRINT  DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
000007 //*
000008 //*Steplib Setup
000015 //STEPLIB  DD DISP=SHR,DSN=DB7T7.SDSNEXIT
000016 //         DD DISP=SHR,DSN=DB7T7.SDSNLOAD
000017 //         DD DISP=SHR,DSN=DB7T7.SDSNLOD2
000018 //         DD DISP=SHR,DSN=CICSTS22.CICS.SDFHEXCI
```

The db2sqljjdbc.properties file

Create/update the db2sqljjdbc.properties file.

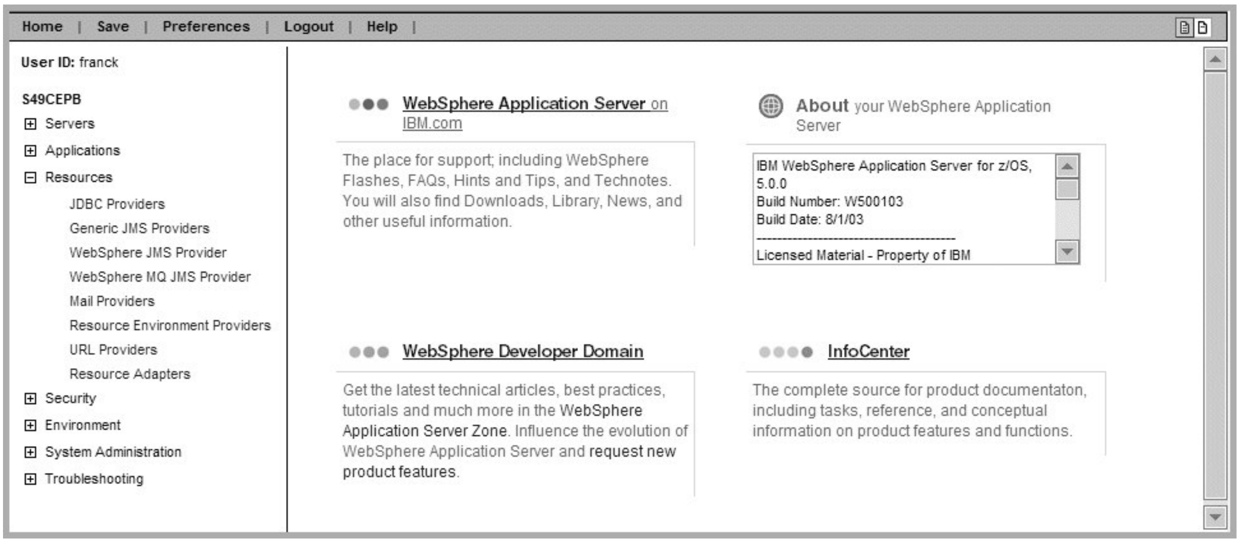
- ✓ The db2sqljjdbc.properties file contains DB2 JDBC driver configuration info
  - Points to location of DSNJDBC_JDBCProfile.ser file as of APAR PQ69861
  - db2sqljjdbc.properties must be in codepage Cp1047 (i.e. EBCDIC).
- ✓ DSNJDBC_JDBCProfile.ser must be placed in the HFS in a directory that is readable by all servant address spaces on the z/OS image

```
EDIT       /SC49/db2v7/UQ77539/classes/db2sqljjdbc.properties  Columns 00001 00072
Command ===>                                                   Scroll ===> CSR
****** ***************************** Top of Data ********************************
000001 # Any lines starting with the pound sign '#' are comments.
000002 # Please see the DB2 for OS/390 Application Programming Guide and Reference
000004 # for Java for the d[illegible]ettings.
000005 #
000006 # SC49 DB2 for PDK ap[illegible]
000007 DB2SQLJSSID=DB7T
000009 # Default values
000010 #DB2SQLJPLANNAME=DSNJDBC
000011 #DB2SQLJ_TRACE_FILENAME=/tmp/mytrc
000012 #DB2CURSORHOLD=YES
000013 #DB2SQLJMULTICONTEXT=YES
000014 #DB2SQLJATTACHTYPE=RRSAF
000015 db2.connpool.max.size=100
000016 db2.jdbc.profile.pathname=/usr/lpp/db2/db2710/classes/DSNJDBC_JDBCProfile.ser
000017 #DB2SQLJMULTICONTEXT=YES
****** ***************************** Bottom of Data ********************************
```

DB2 subsystem ID

Points to DSNJDBC_JDBCProfile

WebSphere V5 console - Resources



The place for support; including WebSphere Flashes, FAQs, Hints and Tips, and Technotes. You will also find Downloads, Library, News, and other useful information.

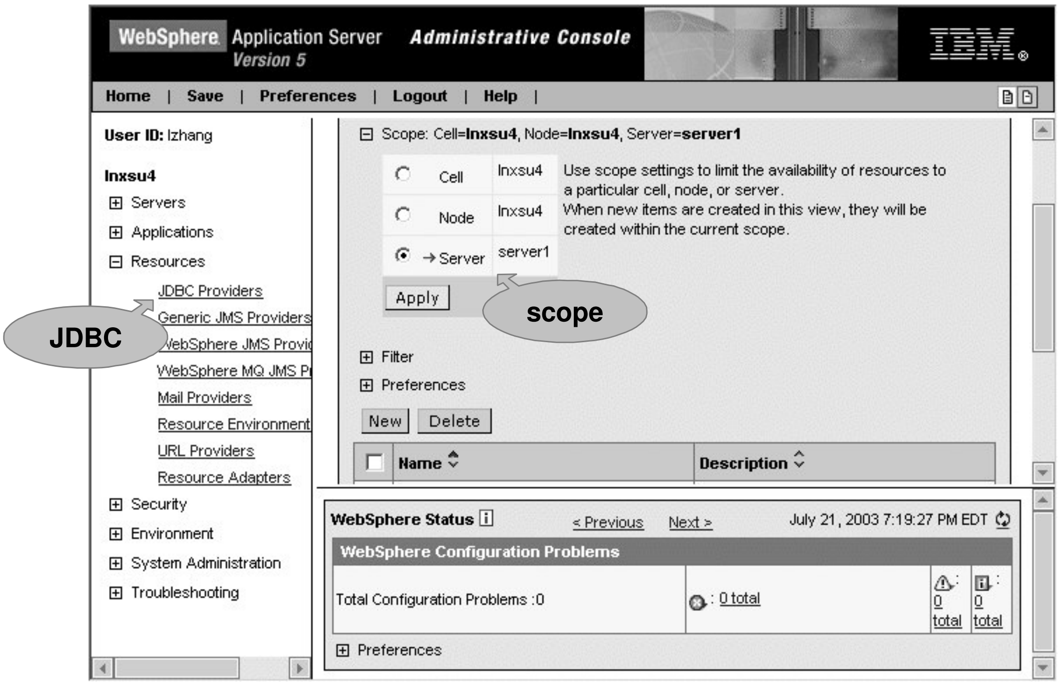
IBM WebSphere Application Server for z/OS, 5.0.0
Build Number: W500103
Build Date: 8/1/03

Licensed Material - Property of IBM

Get the latest technical articles, best practices, tutorials and much more in the WebSphere Application Server Zone. Influence the evolution of WebSphere Application Server and request new product features.

The complete source for product documentaton, including tasks, reference, and conceptual information on product features and functions.

WebSphere V5 console - JDBC resources



⊟ Scope: Cell=**lnxsu4**, Node=**lnxsu4**, Server=**server1**

○ Cell	lnxsu4	Use scope settings to limit the availability of resources to a particular cell, node, or server. When new items are created in this view, they will be created within the current scope.
○ Node	lnxsu4	
● → Server	server1	

July 21, 2003 7:19:27 PM EDT

WebSphere Configuration Problems

Total Configuration Problems :0	: 0 total	0 total	0 total

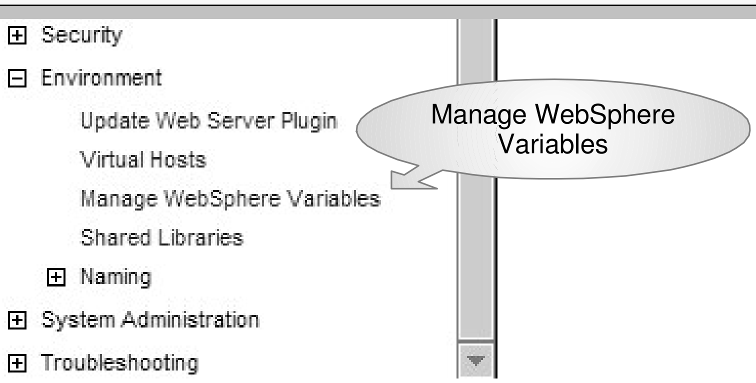
WebSphere V5 console - setting JDBC env. variables

Create/update WebSphere environment variables

- ✓ The WebSphere for z/OS configuration needs to have WebSphere and POSIX environment variables set so that the DB2 home directory and the db2sqljjdbc.properties file can be located.

Click on Environment --> Manage WebSphere Variables.

- ✓ Select the node level view and press “Apply”.



WebSphere V5 console - setting JDBC env. variables

Set DB2390_JDBC_DRIVER_PATH –

- ✓ This variable will be present but have no assigned value. Update this value to the directory in which the DB2 code resides.

  For example: **/usr/lpp/db2/db2710**

Set DB2SQLJPROPERTIES

- ✓ Add this value to indicate the location and name of **db2sqljjdbc.properties** file created for this node.

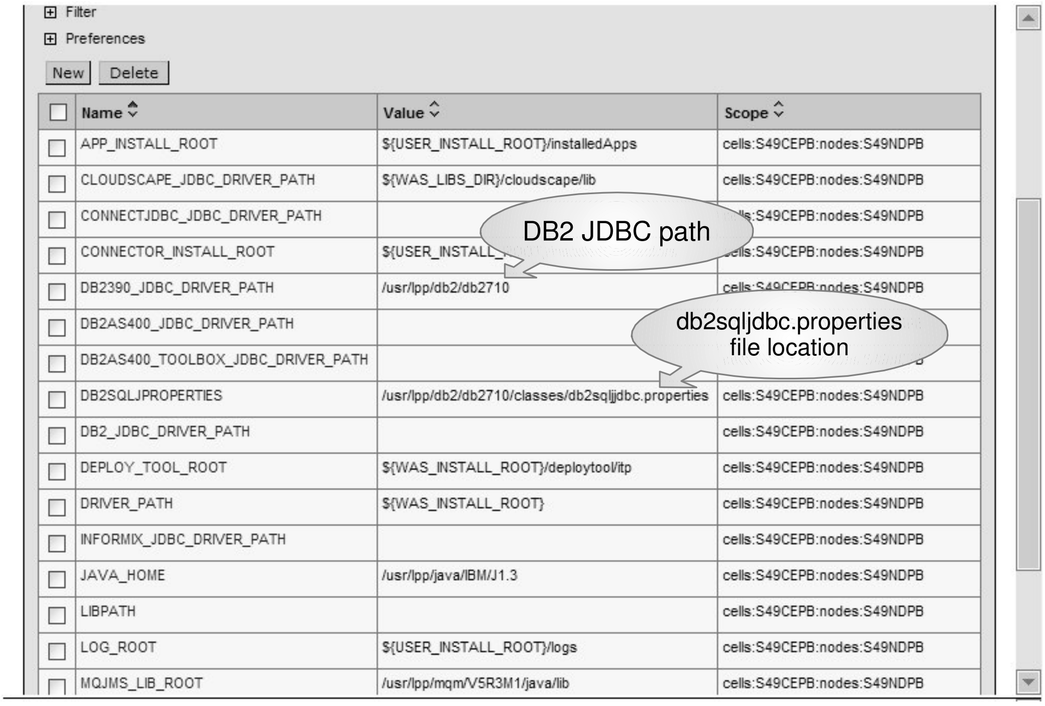
  For example: **/SC49/db2v7/UQ77539/classes/db2sqljjdbc.properties**

  Note - these variables are set at the node level and thus are applicable to all servers in the node residing on this system. They can be set at the server level instead of or in addition to setting them at the node level.

Save changes

Stop/restart the server

WebSphere V5 console - setting JDBC env. variables

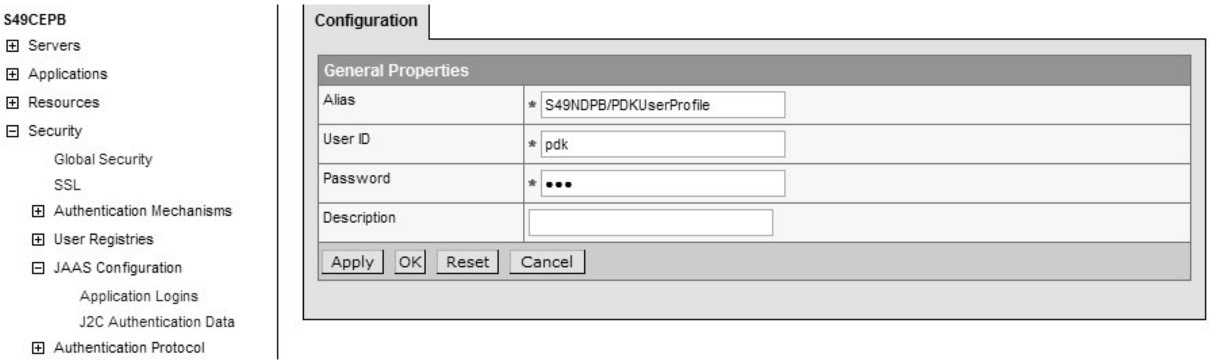


☐	Name	Value	Scope
☐	APP_INSTALL_ROOT	${USER_INSTALL_ROOT}/installedApps	cells:S49CEPB:nodes:S49NDPB
☐	CLOUDSCAPE_JDBC_DRIVER_PATH	${WAS_LIBS_DIR}/cloudscape/lib	cells:S49CEPB:nodes:S49NDPB
☐	CONNECTJDBC_JDBC_DRIVER_PATH		ls:S49CEPB:nodes:S49NDPB
☐	CONNECTOR_INSTALL_ROOT	${USER_INSTALL_	ells:S49CEPB:nodes:S49NDPB
☐	DB2390_JDBC_DRIVER_PATH	/usr/lpp/db2/db2710	cells:S49CEPB:nodes:S49NDPB
☐	DB2AS400_JDBC_DRIVER_PATH		
☐	DB2AS400_TOOLBOX_JDBC_DRIVER_PATH		B
☐	DB2SQLJPROPERTIES	/usr/lpp/db2/db2710/classes/db2sqljjdbc.properties	cells:S49CEPB:nodes:S49NDPB
☐	DB2_JDBC_DRIVER_PATH		cells:S49CEPB:nodes:S49NDPB
☐	DEPLOY_TOOL_ROOT	${WAS_INSTALL_ROOT}/deploytool/itp	cells:S49CEPB:nodes:S49NDPB
☐	DRIVER_PATH	${WAS_INSTALL_ROOT}	cells:S49CEPB:nodes:S49NDPB
☐	INFORMIX_JDBC_DRIVER_PATH		cells:S49CEPB:nodes:S49NDPB
☐	JAVA_HOME	/usr/lpp/java/IBM/J1.3	cells:S49CEPB:nodes:S49NDPB
☐	LIBPATH		cells:S49CEPB:nodes:S49NDPB
☐	LOG_ROOT	${USER_INSTALL_ROOT}/logs	cells:S49CEPB:nodes:S49NDPB
☐	MQJMS_LIB_ROOT	/usr/lpp/mqm/V5R3M1/java/lib	cells:S49CEPB:nodes:S49NDPB

WebSphere V5 console - setting JAAS security for JDBC

Define J2C Authentication aliases

- ✓ If you do not want to connect to DB2 with the servant region's userid, it is necessary to define an alias which holds a userid and password to be used on the getConnection( ) method.
- ✓ Click on Security -> JAAS Configuration -> J2C Authentication Data
  Specify an alias, a userid and a password that can be passed to DB2 .
  Then press "Apply"



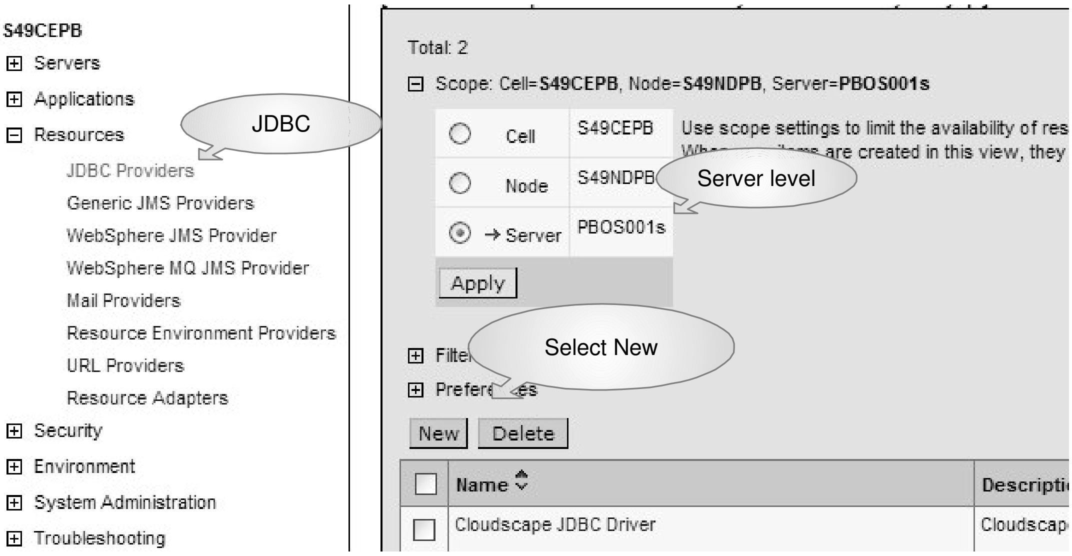
General Properties

Alias	* S49NDPB/PDKUserProfile
User ID	* pdk
Password	* •••
Description	

WebSphere V5 console - Define a JDBC provider (1)

Define a JDBC provider

- ✓ Select Resources --> JDBC Providers.
- ✓ Select the server in which you wish to install a JDBC provider -> press "Apply"
- ✓ Select "New"



Total: 2

⊟ Scope: Cell=**S49CEPB**, Node=**S49NDPB**, Server=**PBOS001s**

○ Cell	S49CEPB	Use scope settings to limit the availability of res
○ Node	S49NDPB	
● → Server	PBOS001s	

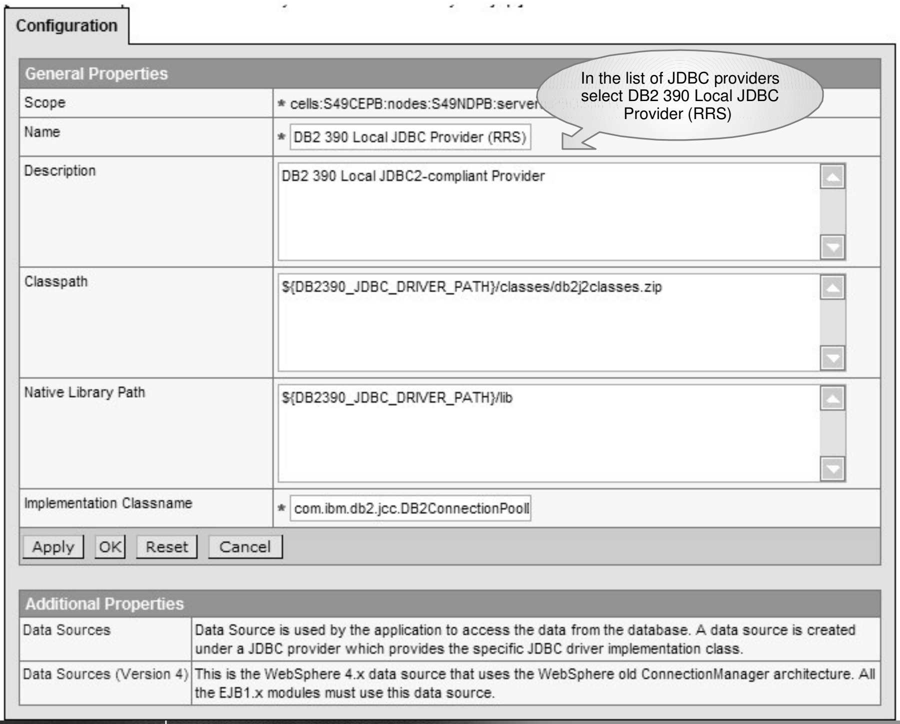
□	Name	Descripti
□	Cloudscape JDBC Driver	Cloudscap

JDBC

Server level

Select New

WebSphere V5 console - Define a JDBC provider (2)



Configuration

General Properties

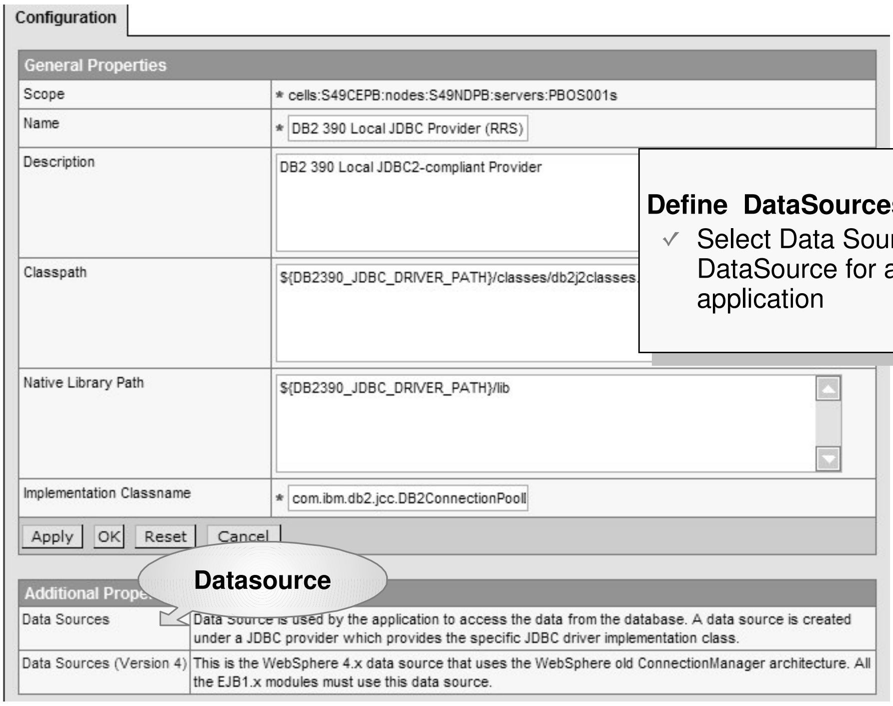
Scope	* cells:S49CEPB:nodes:S49NDPB:server
Name	* DB2 390 Local JDBC Provider (RRS)
Description	DB2 390 Local JDBC2-compliant Provider
Classpath	${DB2390_JDBC_DRIVER_PATH}/classes/db2j2classes.zip
Native Library Path	${DB2390_JDBC_DRIVER_PATH}/lib
Implementation Classname	* com.ibm.db2.jcc.DB2ConnectionPooll

Additional Properties

Data Sources	Data Source is used by the application to access the data from the database. A data source is created under a JDBC provider which provides the specific JDBC driver implementation class.
Data Sources (Version 4)	This is the WebSphere 4.x data source that uses the WebSphere old ConnectionManager architecture. All the EJB1.x modules must use this data source.

In the list of JDBC providers select DB2 390 Local JDBC Provider (RRS)

WebSphere V5 console - Define a Data Source (1)



Configuration

General Properties

Scope	* cells:S49CEPB:nodes:S49NDPB:servers:PBOS001s
Name	* DB2 390 Local JDBC Provider (RRS)
Description	DB2 390 Local JDBC2-compliant Provider
Classpath	${DB2390_JDBC_DRIVER_PATH}/classes/db2j2classes.
Native Library Path	${DB2390_JDBC_DRIVER_PATH}/lib
Implementation Classname	* com.ibm.db2.jcc.DB2ConnectionPooll

Additional Prope

Data Sources	Data Source is used by the application to access the data from the database. A data source is created under a JDBC provider which provides the specific JDBC driver implementation class.
Data Sources (Version 4)	This is the WebSphere 4.x data source that uses the WebSphere old ConnectionManager architecture. All the EJB1.x modules must use this data source.

Datasource

Define DataSources

- ✓ Select Data Sources to create a DataSource for a J2EE 1.3 compliant application

WebSphere V5 console - Define a Data Source (2)

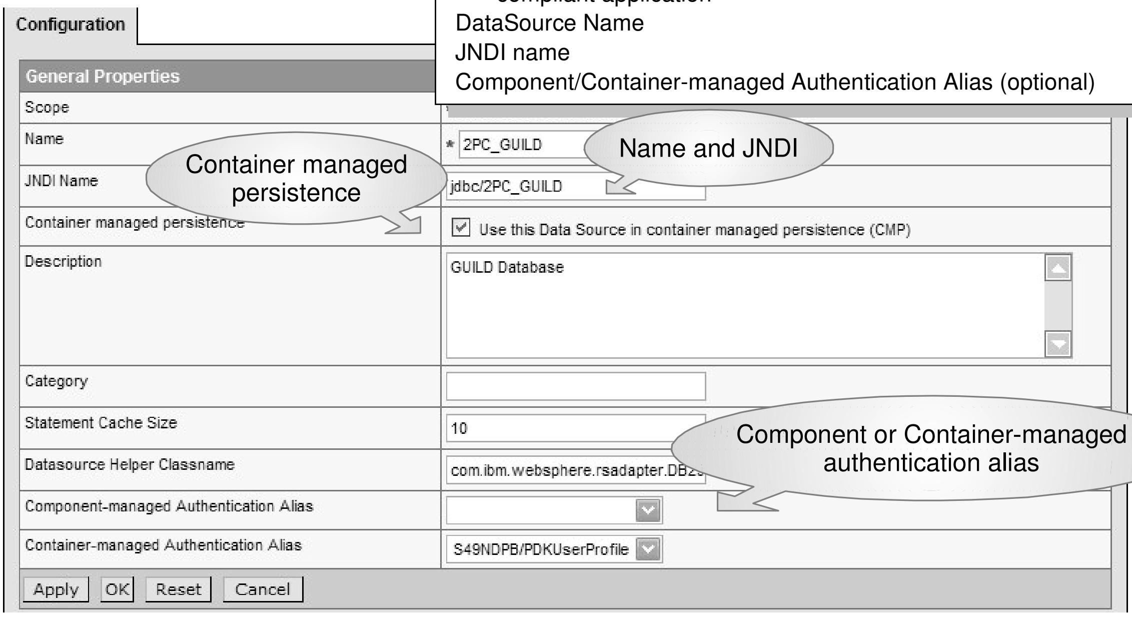
Define DataSources

- ✓ Select DataSource to create a DataSource to for a J2EE 1.3 compliant application

DataSource Name

JNDI name

Component/Container-managed Authentication Alias (optional)



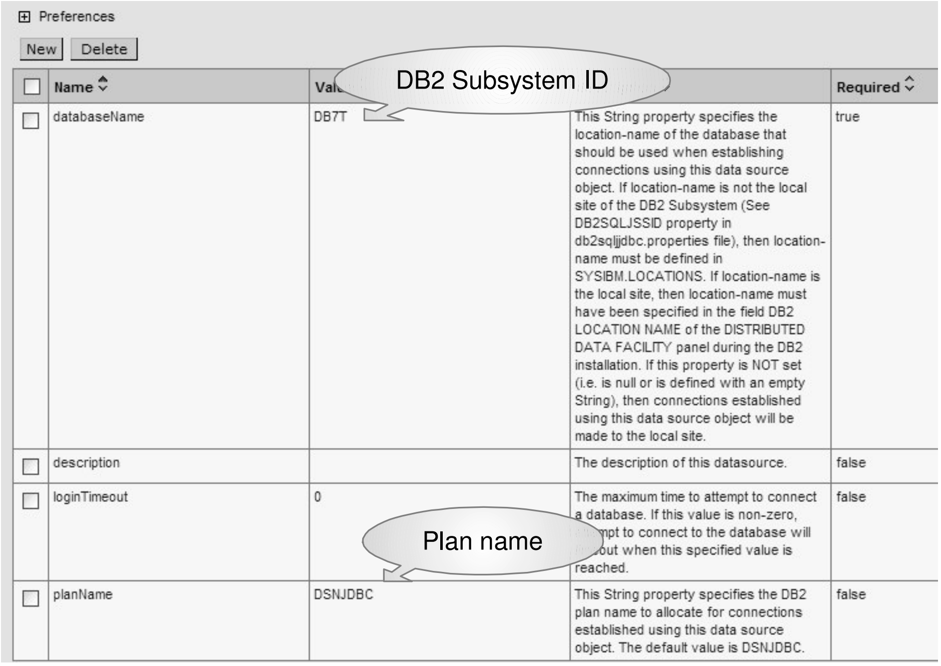
Configuration

General Properties

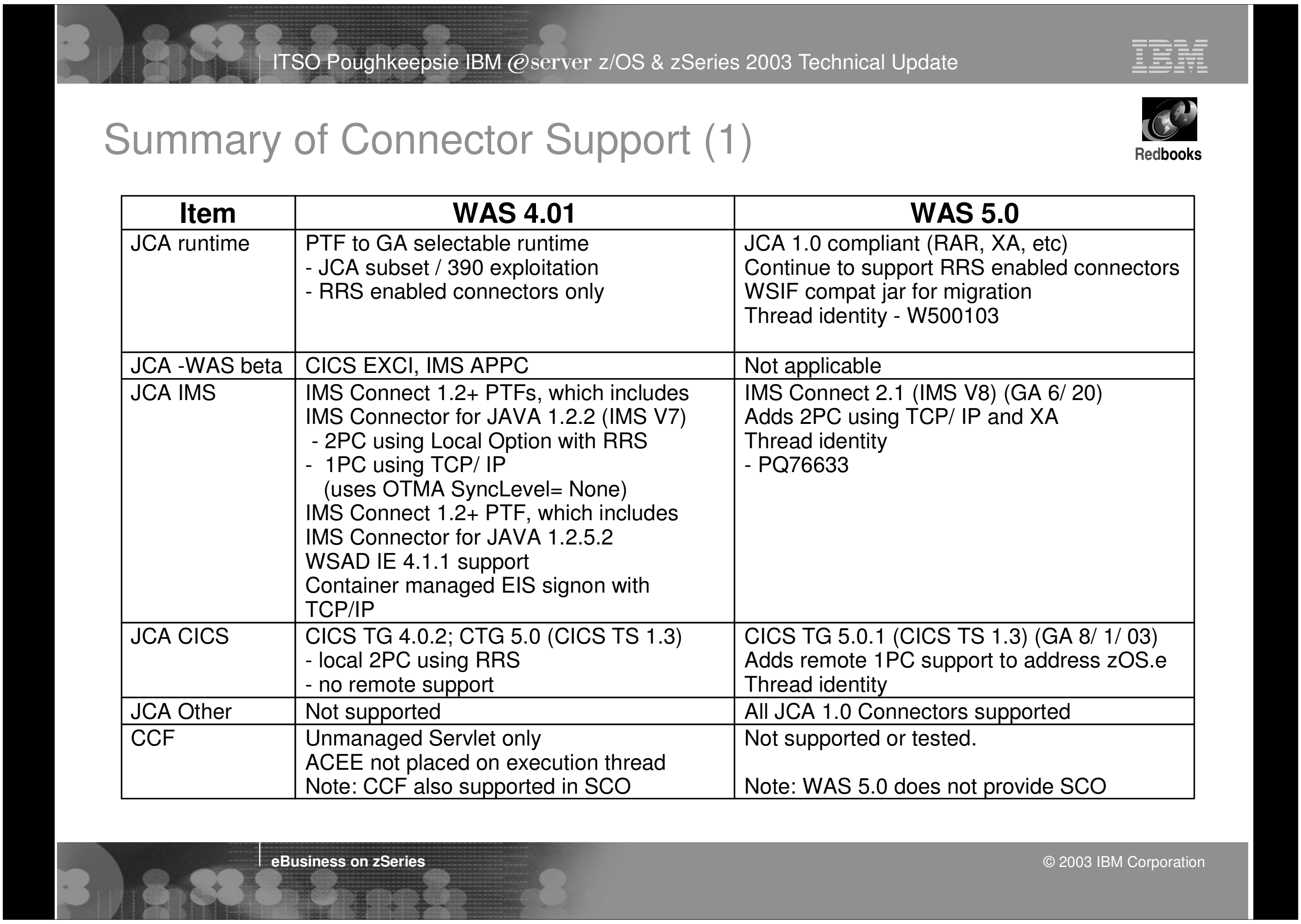
Scope	
Name	* 2PC_GUILD
JNDI Name	jdbc/2PC_GUILD
Container managed persistence	☑ Use this Data Source in container managed persistence (CMP)
Description	GUILD Database
Category	
Statement Cache Size	10
Datasource Helper Classname	com.ibm.websphere.rsadapter.DB2
Component-managed Authentication Alias	
Container-managed Authentication Alias	S49NDPB/PDKUserProfile

Apply
OK
Reset
Cancel
Container managed persistence
Name and JNDI
Component or Container-managed authentication alias

WebSphere V5 console - Define SSID and Plan name

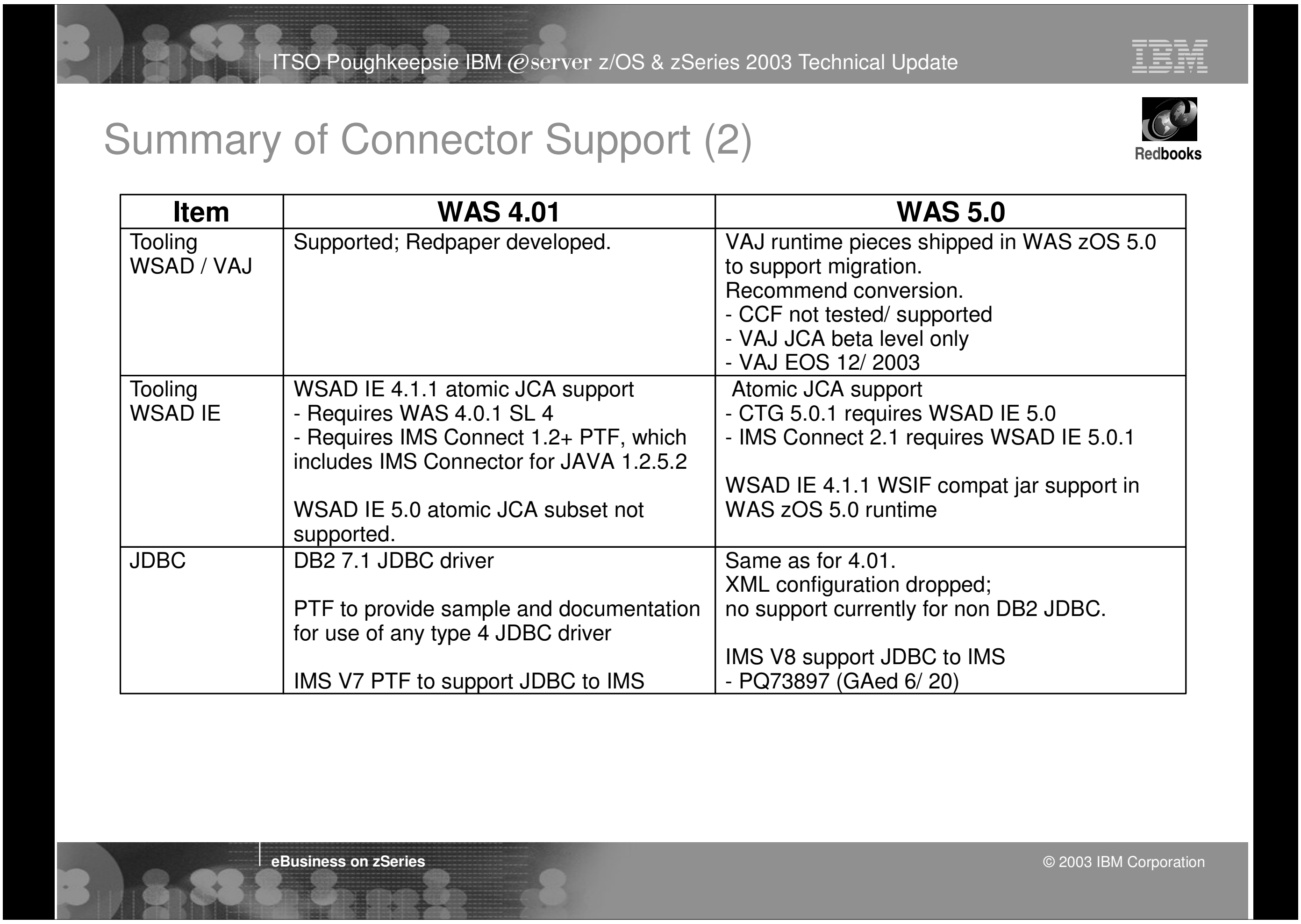


□	Name	Val		Required
□	databaseName	DB7T	This String property specifies the location-name of the database that should be used when establishing connections using this data source object. If location-name is not the local site of the DB2 Subsystem (See DB2SQLJSSID property in db2sqljjdbc.properties file), then location-name must be defined in SYSIBM.LOCATIONS. If location-name is the local site, then location-name must have been specified in the field DB2 LOCATION NAME of the DISTRIBUTED DATA FACILITY panel during the DB2 installation. If this property is NOT set (i.e. is null or is defined with an empty String), then connections established using this data source object will be made to the local site.	true
□	description		The description of this datasource.	false
□	loginTimeout	0	The maximum time to attempt to connect a database. If this value is non-zero, mpt to connect to the database will out when this specified value is reached.	false
□	planName	DSNJDBC	This String property specifies the DB2 plan name to allocate for connections established using this data source object. The default value is DSNJDBC.	false



Summary of Connector Support (1)

Item	WAS 4.01	WAS 5.0
JCA runtime	PTF to GA selectable runtime - JCA subset / 390 exploitation - RRS enabled connectors only	JCA 1.0 compliant (RAR, XA, etc) Continue to support RRS enabled connectors WSIF compat jar for migration Thread identity - W500103
JCA -WAS beta	CICS EXCI, IMS APPC	Not applicable
JCA IMS	IMS Connect 1.2+ PTFs, which includes IMS Connector for JAVA 1.2.2 (IMS V7) - 2PC using Local Option with RRS - 1PC using TCP/ IP (uses OTMA SyncLevel= None) IMS Connect 1.2+ PTF, which includes IMS Connector for JAVA 1.2.5.2 WSAD IE 4.1.1 support Container managed EIS signon with TCP/IP	IMS Connect 2.1 (IMS V8) (GA 6/ 20) Adds 2PC using TCP/ IP and XA Thread identity - PQ76633
JCA CICS	CICS TG 4.0.2; CTG 5.0 (CICS TS 1.3) - local 2PC using RRS - no remote support	CICS TG 5.0.1 (CICS TS 1.3) (GA 8/ 1/ 03) Adds remote 1PC support to address zOS.e Thread identity
JCA Other	Not supported	All JCA 1.0 Connectors supported
CCF	Unmanaged Servlet only ACEE not placed on execution thread Note: CCF also supported in SCO	Not supported or tested. Note: WAS 5.0 does not provide SCO



Summary of Connector Support (2)

Item	WAS 4.01	WAS 5.0
Tooling WSAD / VAJ	Supported; Redpaper developed.	VAJ runtime pieces shipped in WAS zOS 5.0 to support migration. Recommend conversion. - CCF not tested/ supported - VAJ JCA beta level only - VAJ EOS 12/ 2003
Tooling WSAD IE	WSAD IE 4.1.1 atomic JCA support - Requires WAS 4.0.1 SL 4 - Requires IMS Connect 1.2+ PTF, which includes IMS Connector for JAVA 1.2.5.2 WSAD IE 5.0 atomic JCA subset not supported.	Atomic JCA support - CTG 5.0.1 requires WSAD IE 5.0 - IMS Connect 2.1 requires WSAD IE 5.0.1 WSAD IE 4.1.1 WSIF compat jar support in WAS zOS 5.0 runtime
JDBC	DB2 7.1 JDBC driver PTF to provide sample and documentation for use of any type 4 JDBC driver IMS V7 PTF to support JDBC to IMS	Same as for 4.01. XML configuration dropped; no support currently for non DB2 JDBC. IMS V8 support JDBC to IMS - PQ73897 (GAed 6/ 20)

Summary of Connector Support (3)

Item	WAS 4.01	WAS 5.0
JMS	Full implementation using MQ stack - MQSeries 5.2 PTF, MA88, MQ SI 2.1	Integrated JMS provider - Selectable function - Based on MQ products with limitations Full function queue manager - MQ 5.3.1 - shared message queue - bindings mode - CICS/ IMS bridge MQ 5.2/ 5.3 can be utilized - Not J2EE 1.3 compliant - No JAVA client or XA capability. Full function broker - WBI Event Broker for z/OS V5.0 GAed (6/2003)

More Information

- SG24-6541-00, Writing Optimized Java Applications for z/OS
- SG24-5980-01, e-business Cookbook for z/OS Volume III: Java Development

- http://developer.java.sun.com/developer/onlineTraining/