



Session 1107

WebSphere z/OS I'm No Longer a Dummy, But ...

John Hutchinson
hutchjm@us.ibm.com
IBM Advanced Technical Skills

© 2011 IBM Corporation

WebSphere Application Server Sessions



Room	Day	Time	Title	Speaker
208B	Monday	11:00	Lab	Stephen
201A	Monday	11:00	The Value of the WebSphere Application Server Job Manager	Loos
205A	Monday	4:30	WebSphere Application Server for z/OS -- I am No Longer a Dummy but...	Hutchinson
205B	Tuesday	9:30	Performance Tuning for WebSphere Application Server for z/OS - Practical Advice	Everett
205A	Wednesday	4:30	WebSphere Application Server for z/OS: Tools and Tricks (Potpourri)	Loos and Co.
205A	Wednesday	6:00	WebSphere Application Server for z/OS: Helping Customers Help Themselves	Stephen
206B	Thursday	8:00	Securing WebSphere Application Server for z/OS	O'Donnell
206B	Thursday	9:30	Application Improvement and Savings Through Simplification	McCorkle
206B	Thursday	11:00	WebSphere Application Server for z/OS: Batch	Bagwell
206A	Thursday	12:15	WebSphere Application Server 101	Stephen
206B	Thursday	1:30	WebSphere Application Server for z/OS: Availability Considerations	Bagwell
206B	Thursday	3:00	WebSphere Application Server: z/OS Exploitation/Differentiation	Follis
206B	Thursday	3:00	Performance Tuning for WebSphere Application Server for z/OS - WAS and WLM Interactions and Concepts	Follis

© 2011 IBM Corporation

Objective and Agenda

The objective of this session is to take you beyond knowing what WebSphere Application Server is, and seeing a bit more detail ... without taking you *too* far into complexity of it.

- Quick overview: role an “Application Server” and Standards
- Structure and architecture of WebSphere z/OS
- What an “Application” is and how it is packaged / deployed
- Accessing Data and Transactions (DB2, CICS, IMS, MQ)
- Administering WebSphere z/OS environment
- Review of key security topics

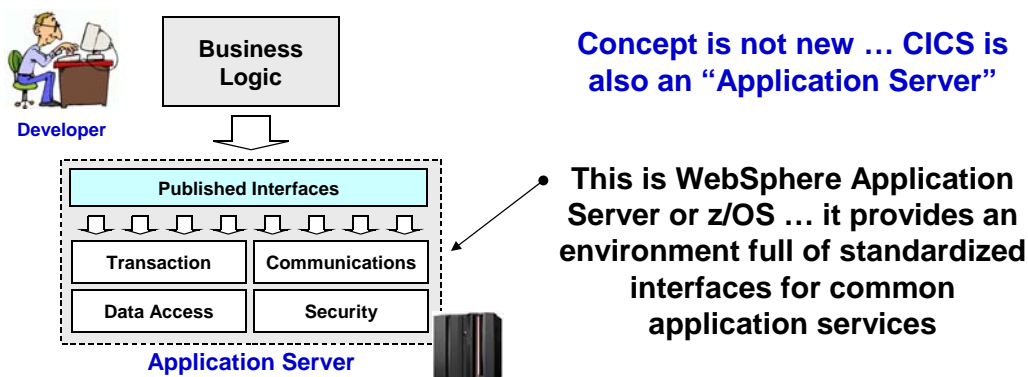
Purpose of an Application Server ...

IBM Advanced Technical Skills

© 2011 IBM CORPORATION

The Purpose of an “Application Server”

Purpose is to provide pre-packaged application support stuff so developers can focus on the main business task.



The key here is the use of open standards, which makes applications more portable across platforms

Open Standards ...

IBM Advanced Technical Skills

© 2011 IBM CORPORATION

Wide Array of Open Standards

Many open standards exist; WebSphere's role is to implement those standards

Specification or API	Version 6.1	Version 6.0	Version 5.1
Java 2 Platform, Enterprise Edition (J2EE) specification	J2EE 1.4	J2EE 1.4 New	J2EE 1.3
Java 2 Standard Edition	J2SE 1.4.2	J2SE 1.4.2 New	J2SE 1.3
ISO 8859 specification	ISO 8859 applies to these versions.		
Java Servlet specification (JSR 154)	Java Servlet 2.4	Java Servlet 2.4 New	Java Servlet 2.3
JavaServer Pages (JSP)	JSP 2.0	JSP 2.0 New	JSP 1.2
Enterprise JavaBeans (EJB) specification	EJB 2.1	EJB 2.1 New	EJB 2.0
Java Data Base Connect	JDBC 3.0	JDBC 3.0 New	JDBC 2.0
Java Message Service (JMS) specification	JMS 1.1	JMS 1.1 New	JMS 1.02

Key Points:

- Open standards are accepted way for computing nowadays
- IBM involved in most of the standards bodies, contributing to and helping shape standards
- You do not need to understand all the standards! Being aware they exist and knowing supported level is usually enough.
- WebSphere Application Server implements many dozen different standards
- The specification levels evolve over time and IBM implements new specs over time.

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.zseries.doc/info/zseries/ae/rovr_specs.html

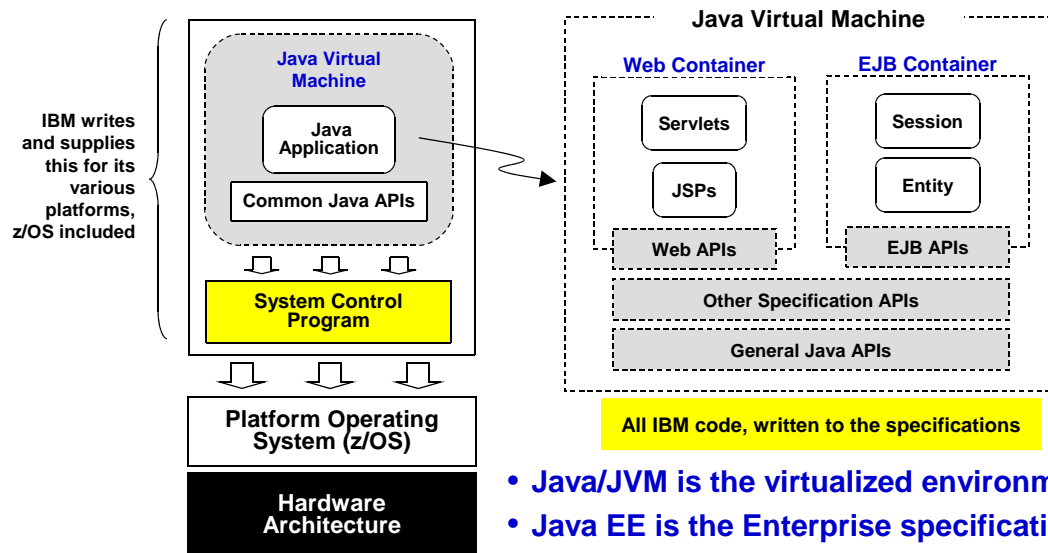
Java and Java EE ...

IBM Advanced Technical Skills

© 2011 IBM CORPORATION

Java, Java EE and "Containers"

This is the fundamental heart of WebSphere ... it is the software implementation of the standards to provide the "application server"



- Java/JVM is the virtualized environment
- Java EE is the Enterprise specification
- "Containers" are the implemented specification APIs within the JVM

Structure of WAS on z/OS ...

IBM Advanced Technical Skills

© 2011 IBM CORPORATION

Structure and Architecture of WebSphere for z/OS

IBM Advanced Technical Skills

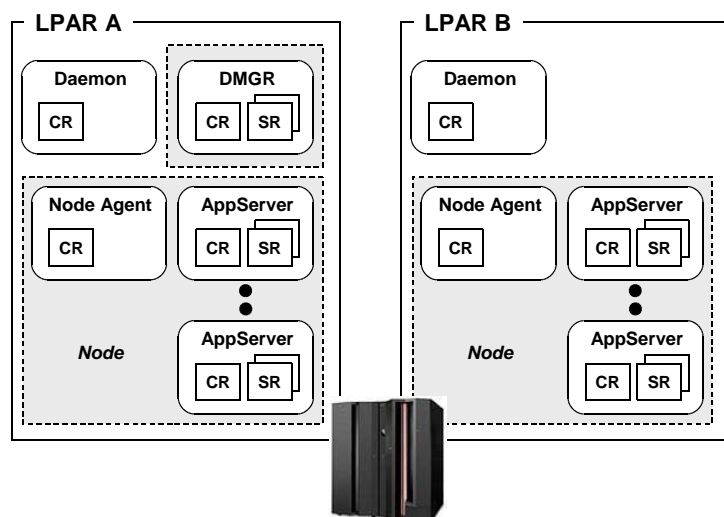
© 2011 IBM CORPORATION

SHARE 116, Spring 2011, Anaheim, CA



Typical Picture Used to Represent WebSphere z/OS Cell

You may have seen a picture like this to represent a WebSphere z/OS “cell” ... we'll use it in this presentation



This is made up of a number of started tasks and address spaces

We're assuming you know the basics of this

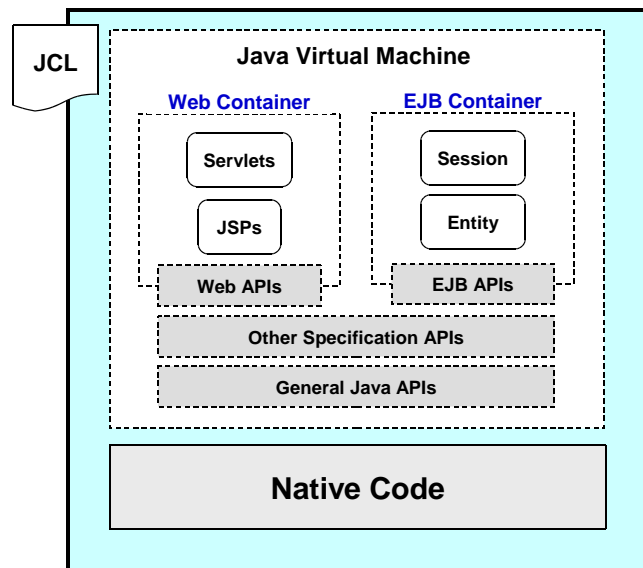
Implemented as started task ...

IBM Advanced Technical Skills

© 2011 IBM CORPORATION

Implementation of Java Within z/OS Address Space

On z/OS the WebSphere Application Server product is implemented as a started task



Key Points:

- Could have been done as a straight port running on Unix Systems Services, but it wasn't
- WebSphere z/OS has some unique code designed to exploit the strengths of the platform
- The started task structure is familiar to z/OS system administrators ... maps well to existing skills and tools

A WebSphere z/OS runtime environment consists of multiple started tasks and multiple address spaces

Let's dig under the covers

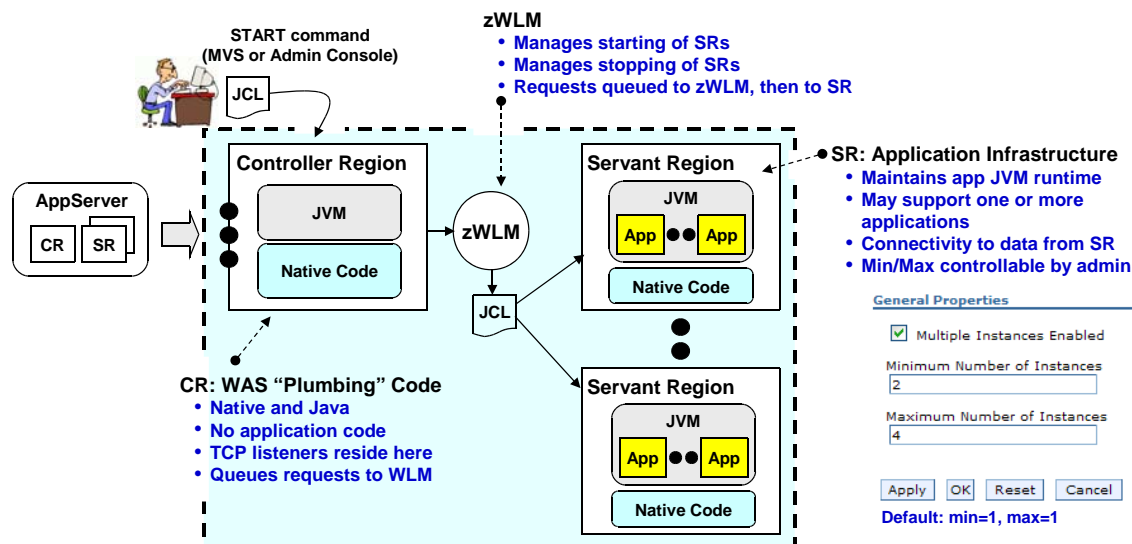
CR/SR structure ...

IBM Advanced Technical Skills

© 2011 IBM CORPORATION

The WAS z/OS Unique "CR/SR" architecture

The first thing we'll look at is the "Application Server," which is made up of two address spaces and maybe more:



Okay ... but why?

Value of CR/SR ...

IBM Advanced Technical Skills

© 2011 IBM CORPORATION

Value of z/OS Controller/Servant Architecture

It provides four areas of value:

1. Provides a form of “vertical scaling”
 - Allow WLM to spawn additional servant regions based on incoming work, or
 - Pre-set number of servant regions to more than one and allow
 - If multiple servants, outage of one does not impact others
2. Separates “plumbing” code from user application code
 - The CR has only IBM code; no user applications run here (runs Key 2 authorized)
 - Outage of SR because of user application will not affect WAS plumbing in CR
3. Queuing structure between CR & SR prevents overwhelming SR with work
 - Work destined for SR is queued to WLM first, then taken off queue by SR when able
 - Allows intelligent load balancing when multiple SRs present
4. Queues work to separate SRs based on transaction classification (TCLASS)
 - Using a “Workload Classification Document”
InfoCenter and search on [trun_wlm_tclass.html](#)
 - With a front-end device such as the HTTP Server or the On Demand Router

An example of exploitation of inherent strength of the platform

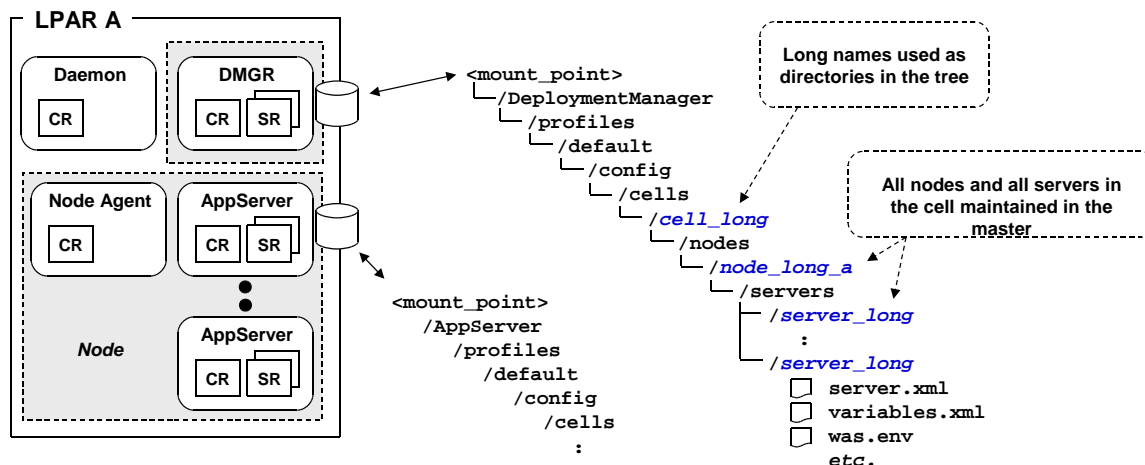
Configuration file structure ...

IBM Advanced Technical Skills

© 2011 IBM CORPORATION

Configuration File Systems and Synchronization

A Network Deployment configuration has at least two file systems to keep in mind -- the “master” and one for each node:



Key Points:

- Full knowledge of the details of this not needed ... awareness of basic concepts is all that's needed
- The DMGR maintains the “master,” which contains everything for the whole cell
- Each node's configuration has what it needs for itself, and an awareness of the configuration of other nodes
- Synchronization involves copying changed files from DMGR master down to node configuration via Node Agent

Applications ...

IBM Advanced Technical Skills

© 2011 IBM CORPORATION

Applications

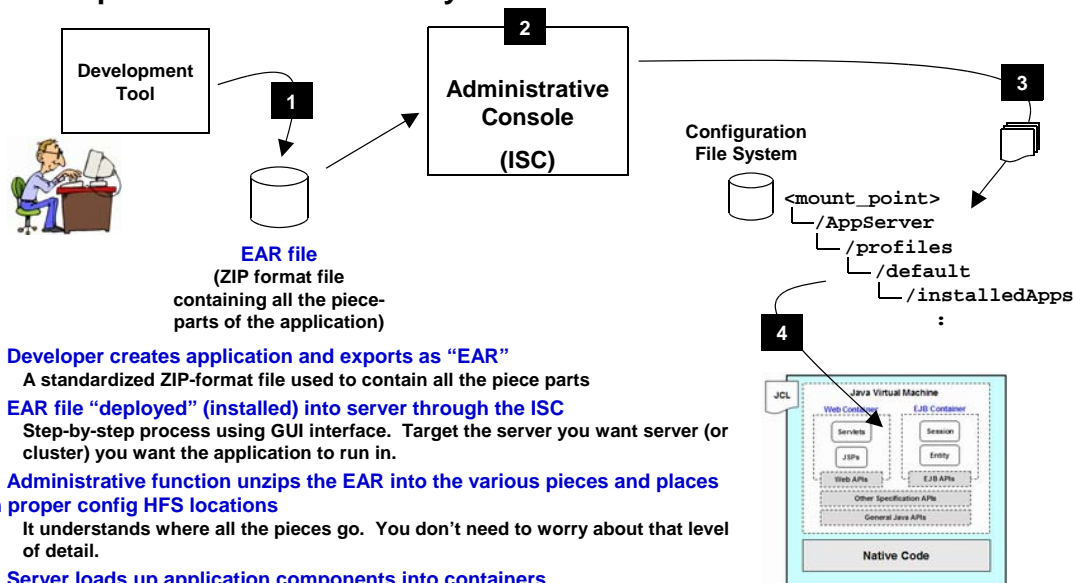
How They're Packaged and Deployed

IBM Advanced Technical Skills

© 2011 IBM CORPORATION

Applications - Overview

A WebSphere “application” is a collection of Java classes and other pieces that make up the business solution you’ve built



1. Developer creates application and exports as “EAR”

A standardized ZIP-format file used to contain all the piece parts

2. EAR file “deployed” (installed) into server through the ISC

Step-by-step process using GUI interface. Target the server you want server (or cluster) you want the application to run in.

3. Administrative function unzips the EAR into the various pieces and places in proper config HFS locations

It understands where all the pieces go. You don’t need to worry about that level of detail.

4. Server loads up application components into containers

Webapp pieces into web container; EJB pieces into EJB container -- all automatic based on internal-WAS knowledge of this stuff

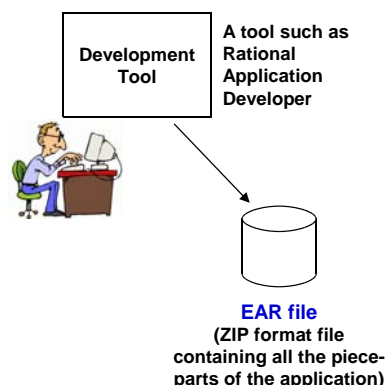
Packaging ...

IBM Advanced Technical Skills

© 2011 IBM CORPORATION

Application Packaging

WebSphere Applications are typically packaged in an “EAR” file format, which is a standardized ZIP-format file containing many smaller pieces inside



JAR File(s)

- Hold EJB application components
- Are themselves a standardized ZIP-format file
- Contain Java class files and XML “deployment descriptors”
- May have zero, one or more than one JAR file in EAR

WAR File(s) *

- Hold web application components
- Are themselves a standardized ZIP-format file
- Contain Java class files and XML “deployment descriptors”
- May have zero, one or more than one WAR file in EAR

Application Deployment Descriptors

- XML and XMI files
- Contain information about the application
- Example: application.xml

Other Files

- Basic stuff -- MANIFEST.MF, and development tool info files

WebSphere Application Server for z/OS is designed to understand this standardized format. Key point here -- EAR will contain what the developer created for the application

* Yes, is possible to deploy a WAR without having it packaged as EAR

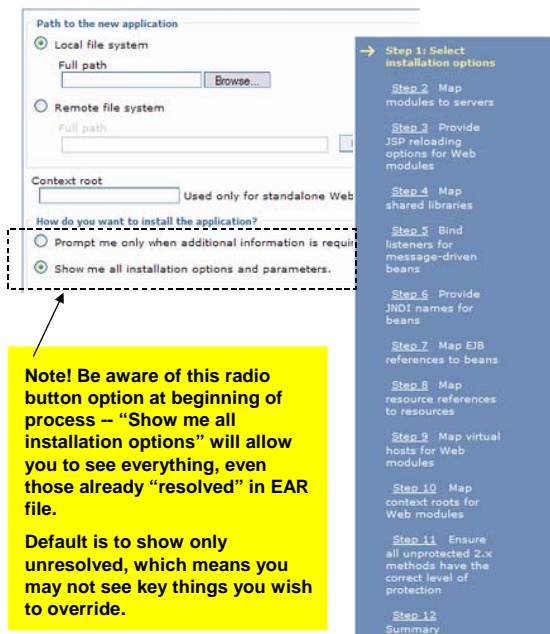
Admin Console and EAR ...

IBM Advanced Technical Skills

© 2011 IBM CORPORATION

Installation Function and What It Sees in the EAR File

The installation function within the Administrative Console is shaped by what it sees in the EAR file being deployed. It reads the deployment descriptors.



Key Points:

- The step-by-step instructions you see is very much determined by what the Admin function sees in the EAR file
- Purpose of this is to supply information that allows key values to be “resolved” to resources defined in WebSphere ... data connectors, etc.
- It is possible the developer resolved some of this in the EAR file. See note to the left.
- With information supplied, WebSphere z/OS can install the application into the runtime and provide all the necessary information about how to connect the application to needed resources.

Application states ...

IBM Advanced Technical Skills

© 2011 IBM CORPORATION

Application “States”

This is a simple thought exercise to help you understand the various “states” of an application as it goes from developer EAR to application ready to access

1. Application not yet deployed

The developer has delivered the EAR, but you’ve not yet deployed it into WAS.

2. Deployed, but not yet saved

There is a point in the process where the Administrative Console will report the application is “installed successfully.” But until you save it, the files are in a temporary holding location.

3. Saved, but not yet synchronized

In an Network Deployment environment synchronization is key to getting application files from DMGR “master configuration” out to the node where the server actually resides. Synchronization may or may not be automatic at time of save.

4. Synchronized, but not yet started

An application can be synchronized out to the node, but by default the application will be in a stopped state (red X) unless you start it or recycle the server. Stopped applications can’t be accessed.

5. Started and ready to access

The application is in a true “ready” state.

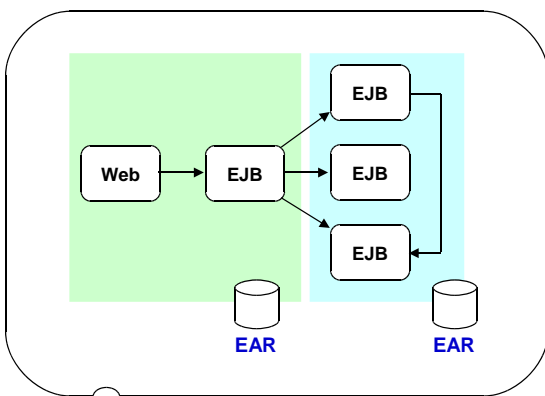
Business applications ...

IBM Advanced Technical Skills

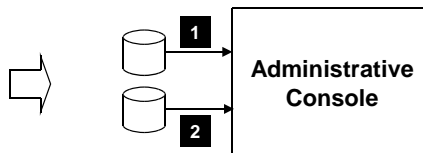
© 2011 IBM CORPORATION

Applications Comprised of Multiple EAR Files

Be aware that the term “application” might refer to a combination of EAR files. It’s not common, but it’s starting to come up more and more.



The developer may architect the “application” to be packaged as separate EAR files



Implies two separate installations through Administrative Console

Key Points:

- Be aware this possibility exists
- If you suspect this is what people mean by “application” ... ask and clarify
- If multiple EARs comprise “application” then talk to developer and get good sense for how they are to be deployed -- same server? etc.

Data access ...

IBM Advanced Technical Skills

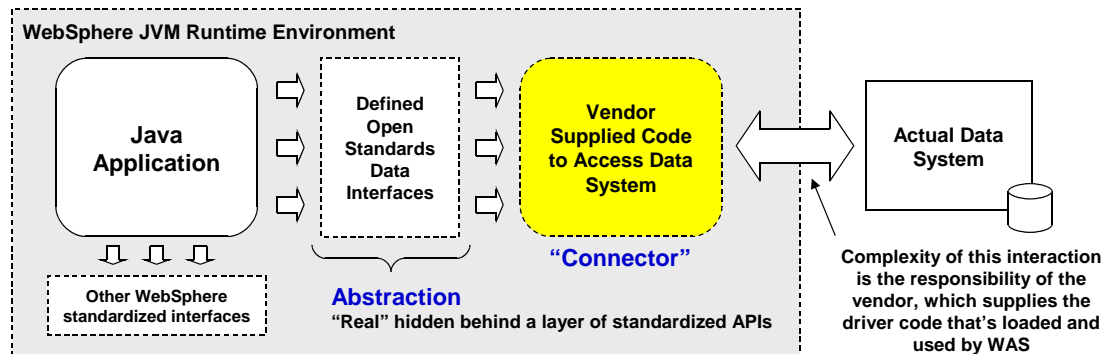
© 2011 IBM CORPORATION

Data Access

Accessing Data - Overview

The key is to understand that what Java EE is trying to do is shield developer from having to know the details of each vendor's implementation.

It does this by *abstracting the actual data implementing using standardized interfaces*



Two basic question to answer here:

1. What kind of data access connectors are available?
2. Where does the “vendor supplied code” (connector code) come from?

The specifics of how to configure the connectors into WebSphere is beyond time limit of this presentation

Four kinds ...

Four Basic Kinds of Data Access

This chart summarizes the four basic data access methods out of WebSphere:

1. Relational Database

- Examples: DB2, Oracle, Sybase, Informix, Cloudscape, Derby
- Uses **JDBC** ... Java Data Base Connectivity
- For DB2 -- JDBC drivers come with installed DB2. Configure WAS to load drivers.

2. Non-relational Data Systems

- Examples: CICS, IMS, SAP, PeopleSoft
- Uses **JCA** ... Java Connector Architecture
- For CICS -- resource adapter comes with separate product: "CICS Transaction Gateway"
- For IMS -- downloadable from ibm.com website

3. Messaging Systems

- Examples: **JMS** or MQ
- Uses ... see upcoming chart on the "JMS/MQ Confusion" issue
- Default messaging provider simply part of the WebSphere z/OS product.
- WebSphere z/OS comes with code that allows client mode connection to MQ
- Otherwise, you can configure in connection code using what's supplied with MQ installation

4. Other

- Things like Web Services (SOAP), or calling another EJB (IIOP)
- We won't cover these in this presentation

The first three are what we'll focus on

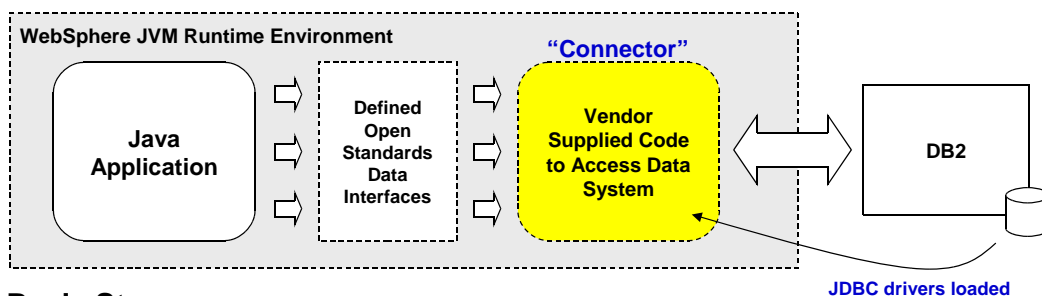
Example: JDBC ...

IBM Advanced Technical Skills

© 2011 IBM CORPORATION

Example: JDBC Access to DB2

To give you a sense for this, here's how you access DB2 using "JDBC Drivers" supplied by IBM DB2:



Basic Steps:

1. In ISC define a "Provider," which tells WebSphere what kind of database will be accessed (DB2) and location in the HFS where the DB2 installation can be found

- JDBC drivers supplied as a set of Java class files in the DB2 installation directories
- Key -- this tells WebSphere where to get the driver code

2. In ISC define a "Data Source," which tells WebSphere the specifics of the connection -

- DB2 location name, the type of connection (local or IP remote), etc.
- Key -- this tells WebSphere what relational data system to connect to

3. At application deployment time, map application "Resource Reference" to the data source,

- Provides the connection -- App to abstraction, abstraction to real.
- Really good analogy -- DDNAMEs. COBOL source has DDNAME; DDNAME resolved to real in JCL. Same concept here.

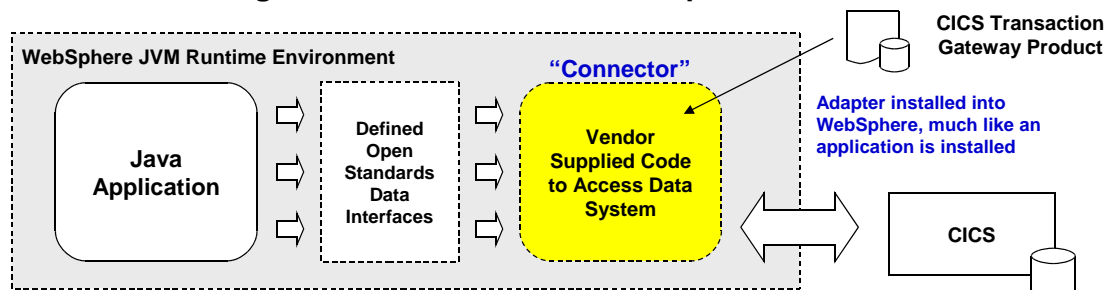
CICS Resource Adapter ...

IBM Advanced Technical Skills

© 2011 IBM CORPORATION

Example: JCA Access to CICS

Similar in concept to JDBC, it uses a different approach to supplying access code ... something called a “JCA Resource Adapter”



Basic Steps:

1. In ISC install the CICS Transaction Gateway resource adapter into WebSphere.

- Packaged as a RAR file (Resource Archive ... another zip-format file)
- Resource Adapter contains code that implements standard programming interface and CICS access capability
- This is similar in concept to the JDBC “Provider” -- it defines where WebSphere gets the access code

2. In ISC define a “Connection Factory,” which tells WebSphere the specifics of the connection

- CICS region - type of connection (local or IP remote), etc.
- This is similar in concept to the JDBC “Data Source” -- it defines where and how to connect.

3. At application deployment time, map application “Resource Reference” to the connection factory

- Provides the connection -- App to abstraction, abstraction to real.
- Same DDNAME analogy applies here.

IMS works in a way very similar to this

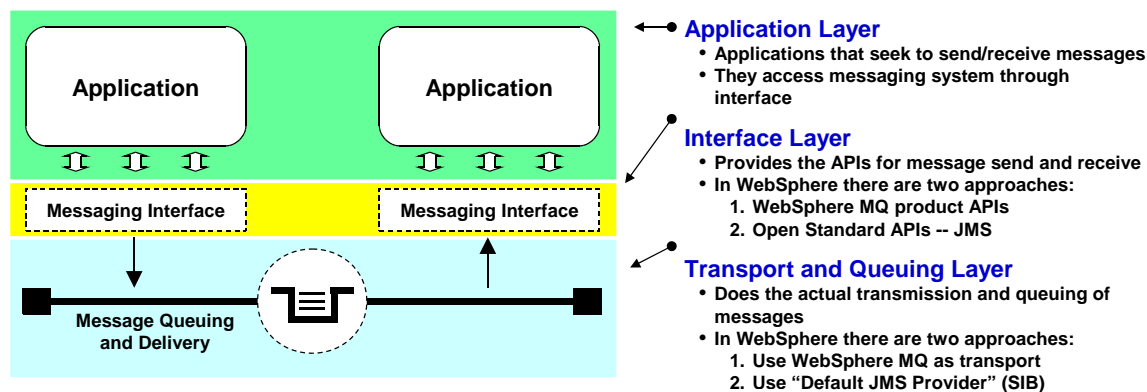
JMS/MQ confusion ...

IBM Advanced Technical Skills

© 2011 IBM CORPORATION

The JMS/MQ Confusion Issue

Let’s try to clear up the confusion here ... key is understanding the “layers” of a messaging architecture



API	Transport
Direct use of MQ product APIs	Must use MQ as the transport since product (not JMS) APIs in use
Use of standard JMS APIs	Transport: MQ -- configure MQ as the “JMS Provider” and define JMS destinations so they map to actual MQ queues in real QMGR.
	Transport: SIB -- configure Service Integration Bus and destinations. Messages handled by Java-implemented transport inside of WAS

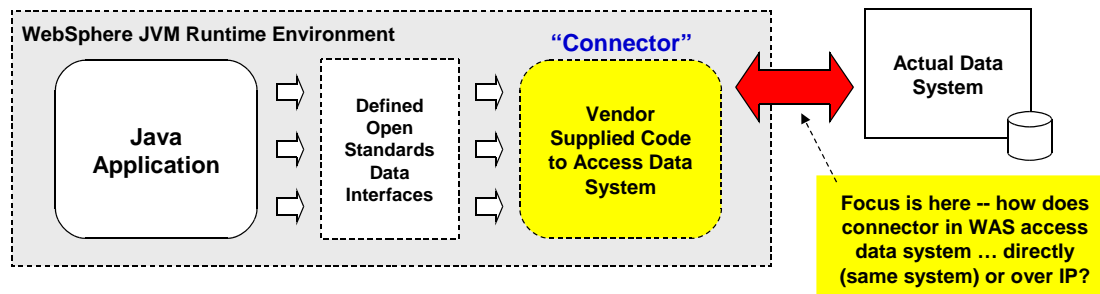
Proximity ...

IBM Advanced Technical Skills

© 2011 IBM CORPORATION

The Question of Proximity -- WAS to the Data

This affects performance, security implications and how the connectors are configured:



Connector	Same System*	Over IP
JDBC	“Type 2”	“Type 4”
JCA	“Local”	“Remote”
JMS over MQ	“Bindings Mode”	“Client Mode”

Objective here -- align concepts with terminology used. All the details about configuration, security implications, performance impacts, etc., just can't be covered in this hour.

* Is possible to do IP on same system

Java and Native code ...

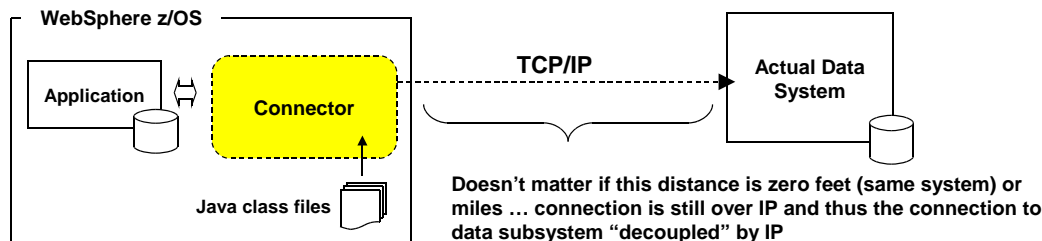
IBM Advanced Technical Skills

© 2011 IBM CORPORATION

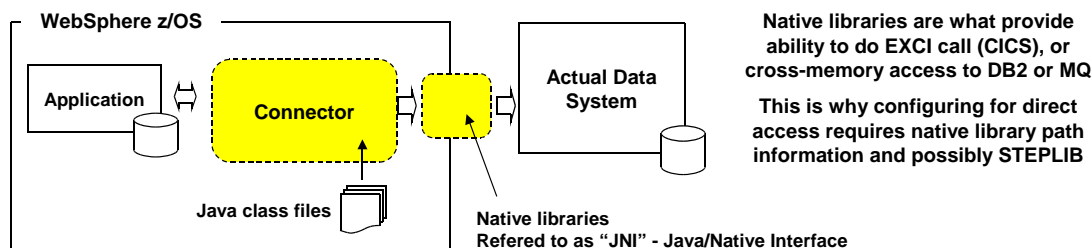
Java Code and Native Code -- When They Mix

When data access involves directly accessing the data system, then native code comes into play. Remote requires only Java.

When the data access is over IP -- just Java class files



When the data access is direct (DB2 Type 2; MQ Bindings, etc.) -- native libraries come into play



Administration ...

IBM Advanced Technical Skills

© 2011 IBM CORPORATION

Administering WebSphere

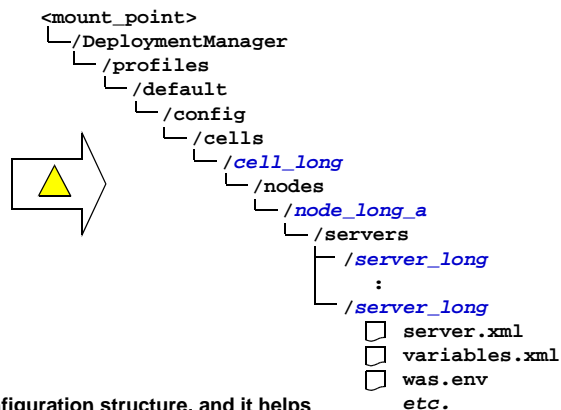
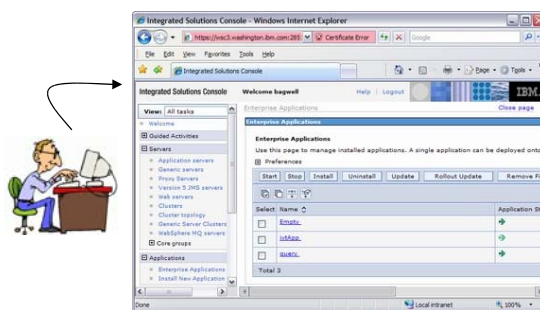
Administrative Console a.k.a. Integrated Solutions Console (ISC)

IBM Advanced Technical Skills

© 2011 IBM CORPORATION

Using the Administrative Console

The Administrative Console is an application supplied with WebSphere that understands how to translate your mouse-clicks into XML file changes:



Key Points:

- This eliminates you having to know details of the configuration structure, and it helps reduce human error that would happen with direct edit of XML
- Administrative Console runs in DMGR if ND configuration; Standalone server if that type of configuration
- Don't forget -- if ND, then there's the master-configuration / synchronize issue keep in mind
- Learning all the options within Admin Console takes time ... can't learn it all at one time
- Downside -- hard to exactly duplicate customization cell to cell ... which is why management interface to WebSphere is available

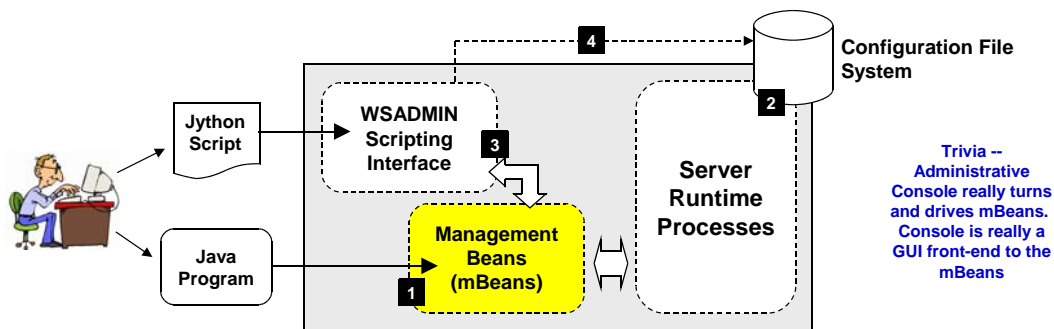
Management interface ...

IBM Advanced Technical Skills

© 2011 IBM CORPORATION

Management Interface to WebSphere Application Server

To facilitate the programmatic management of WebSphere, the product has a management interface. Two layers: WSADMIN and mBeans ...



Key Points:

- Management Beans (mBeans) are just Java classes (programs) provided with the product that allow a programmatic of the environment. You can do things like deploy applications, start/stop servers ... everything the Administrative Console can do mBeans can do as well. mBeans are driven by a Java program you write.
Requires Java programming skills to do this, which is why WSADMIN provided (see #3)
- If the change requested involves an update to the configuration file structure, the mBeans will take care of that
- WSADMIN is a "scripting interface" that makes using mBeans simpler. Java skills not required to use WSADMIN. All you need to do is code up Jython script to drive WSADMIN commands. In most cases those WSADMIN commands then turn and drive the mBeans
- When WSADMIN run in "unconnected mode" (local mode), it is capable of making direct changes to configuration structure. Not recommended when servers up and running.

Example WSADMIN ...

IBM Advanced Technical Skills

© 2011 IBM CORPORATION

Very Simple Example of WSADMIN Script

This will serve to illustrate how easy this can be.

Actual Commands:

```
AdminApp.install(/u/myhome/MyApp.ear,['-node mynodea -server mysr01a'])
AdminConfig.save()
```

Placed in a script file with simple variables making command bit more flexible:

```
# -----
# Set variables
# -----
app_srv      = 'mysr01a'
app_node     = 'mynodea'
app_ear      = '/u/myhome/MyApp.ear'
# -----
# Install application
# -----
print 'Installing application'
options = '[-node ' + app_node + ' -server ' + app_srv + ']'
AdminApp.install(app_ear,options)
AdminConfig.save()
```

Forming up the command and issuing them

Message -- you can do this ... takes a little time to get used to it, but you'll get it

Security ...

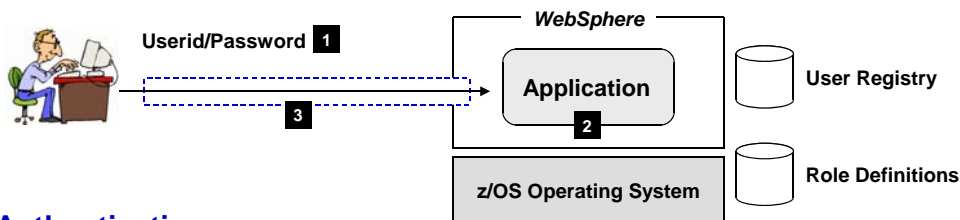
IBM Advanced Technical Skills

© 2011 IBM CORPORATION

Brief Security Overview

Three Essential Elements to WebSphere Security

Lots more detail behind this, but this is the barest essence of it:



1. Authentication

- Issue – is user allowed to access or not? Typical method is with userid/password challenge
- WebSphere must then check if userid is known, and if password supplied correct.
- Choices for user registry: SAF (RACF); LDAP; or custom registry

2. Authority

- Issue – what can user do in application once authentication allowed access?
- Java EE application has option to use “roles” – definition of what access is allowed by user or user group.
- Definition of those roles can be in application bindings or in SAF via EJBROLE profiles.
- Not all applications use roles ... could well be that none are defined and authenticated user allowed full access

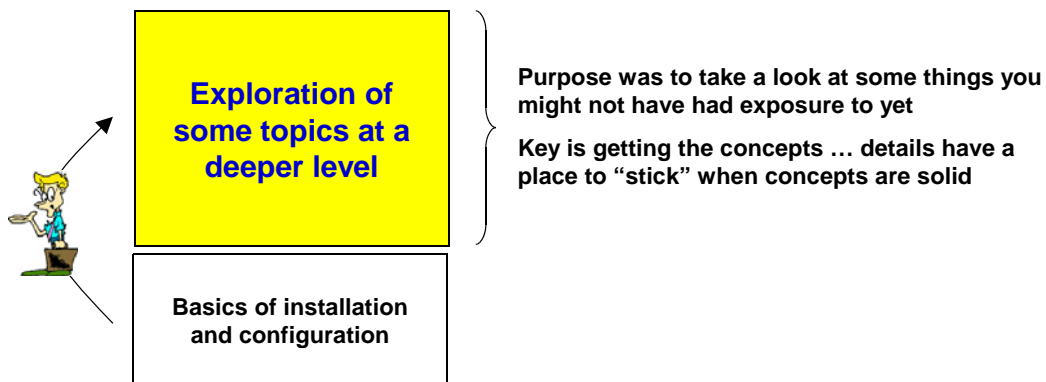
3. Transport Confidentiality

- Issue – can data sent back and forth be protected from being watched, or changed in transit?
- Typical solution is to use Secure Socket Layer (SSL)
- Complex topic involving “certificates” ... key is that it provides data encryption over network so data can’t be seen, and if changed in flight then change is detected.

Several WAS z/OS Security Sessions at this SHARE

Other sessions ...

Summary and Conclusion



Questions?