z/OS Introduction and Workshop

WebSphere Application Server
Unit Objectives

After completing this unit, you should be able to:

• Describe WebSphere Application Server

• Be familiar with the WAS Administration Console
Terminology

- **WebSphere Application Server**
  - Is the name of the product
  - Also used to refer to the actual process that runs the application code

- **Server**
  - This is the component that has the Java Virtual Machine (JVM)
  - This is where the application programs run

- **Cluster**
  - Logical term used to describe a group of servers
Terminology continued….

- **Node**
  - Logical term to describe a single machine that runs one or more servers

- **Cell**
  - Logical term that covers the complete WebSphere configuration

- **Daemon**
  - Separate process required on z/OS
  - Small component, needs little attention
WAS – Dynamic Web Pages – HTTP Server

1. http://www.myzseries.com/cgi-bin/test.cgi

2. httpd.conf

3. Response

CGI – Common Gateway Interface
WAS – Dynamic Web Pages – Interaction with WebSphere

- **Client Browser**
- **URL**: http://www.myzseries.com/my.jsp
- **HTTP Server**
  - `httpd.conf`
  - `was.conf`
- **WAS Plugin**
- **Servlet**
- **Response**
- **CICS Server** or **IMS Server**

WebSphere plug-in, same address space
WAS – Dynamic Web Pages – Interaction with WebSphere

http://www.myzseries.com/my.jsp

HTTP Server

J2EE Server

CICS Server
or
IMS Server

Client Browser

URL

Response

httpd.conf

was.conf

WAS plugin

Servlet

EJB Container

EJB

Web container inside HTTP Server, separate EJB container
WAS – Dynamic Web Pages – Interaction with WebSphere

Separate J2EE server with both Web container and EJB container
WebSphere Application Server Configuration on z/OS

Network Deployment Manager

- Location Service Daemon (BBODMNB)
  - Controller

- Node 1: Deployment manager
  - Deployment manager (BBODMGR)
    - Controller

- Node 2: Application server
  - Node agent (BBON001)
    - Controller
  - JMS server (BB0J001)
    - Controller

- J2EE scalable application server (server1)
  - Controller
  - Servants

- HTTP internal transport

- z/OS functions
  - UNIX System Services
    - TCP/IP
  - FTP
  - RRS
  - Workload Management
  - Language Environment
  - Security Server
  - ARM
  - IMS/TM
  - CICS/TS
  - MQ

- v5 run-time environment
WebSphere Application Server Configuration on z/OS

Cell

Location Service Daemon (BBODMNB)

Controller

Node 1: Deployment manager

Deployment manager (BBODMGR)

Controller

Node 2: Application server

Node agent (BBOND01)

Controller

JMS server (BBJO001)

Controller

J2EE scalable application server (server1)

HTTP internal transport

Controller

Servants

z/OS functions

- UNIX System Services
- TCP/IP
- FTP
- RRS
- Workload Management
- Language Environment
- Security Server
- ARM
- IMS/TFM
- OCS/TS
- MQ

V5 run-time environment

FIG 1: WEBSHARE FOR DISTRIBUTED PLATFORM SERVER ARCHITECTURE

FIG 2: WEBSHARE FOR z/OS SERVER ARCHITECTURE
Work Load Management (WLM)

- A base component of the operating system enables prioritization and balancing of work according to customer selected 'goals' or business policies.

- With workload management, you define performance goals and assign a business importance to each goal.
  - Goals:
    - Response-Time
    - Execution Velocity
    - Discretionary
  - Importance level (1-5)

- Goal is 1 or below (meeting goals)
  - All is well

- Goal is above 1 (failing to meet goals)
  - Revise performance goals or increase capacity
Starting WebSphere Application Server V7

START XSDCR, JOBNAME=XSDMGR, ENV=XSCCELL.XSDMNODE.XSDMGR
START XSACR1, JOBNAME=XSAGNT1, ENV=XSCCELL.XSNODE1.XSAGNT1

url:9505/ibm/console

P XSDEMN  <<< Stop WebSphere Application Server V7

Starting WebSphere Application Server V6.1

START XBMGCR, JOBNAME=XBDMGR, ENV=XBCELL.XBDMNODE.XBDMGR
START XBACR1, JOBNAME=XBAGNT1, ENV=XBCELL.XBNODE1.XBAGNT1

url:8518/ibm/console

P XBDEMN  <<< Stop WebSphere Application Server V6.1

Post Installation Customization

XSCCELL.*  for WAS V7
XBCELL.*  for WAS V6.1

*Note: Above details apply only to class lab system
START XSDCR, JOBNAME=XSDMGR, ENV=XSCCELL.XSDMNODE.XSDMGR

$HASP373 XSDMGR STARTED
BBO00001I WEBSPHERE FOR Z/OS CONTROL PROCESS
XSCCELL/XSDMNODE/XSDMGR/XSDMGR IS STARTING.
BBO00238I WEBSPHERE FOR Z/OS CONTROL PROCESS xscell/xsdmnodexdmg IS STARTING.
START XSDEMN, JOBNAME=XSDEMN, ENV=XSCCELL.XSCCELL.S0W1, REUSASID=YES
$HASP373 XSDEMN STARTED
BBO00007I WEBSPHERE FOR Z/OS DAEMON XSCCELL/XSDMNODE/XSCCELL/S0W1 IS STARTING.
BBO00237I WEBSPHERE FOR Z/OS DAEMON xscell/xsdmnodexnode IS STARTING.
BBO00221I: WSVR0001I: Server CONTROL PROCESS dmgr open for e-business
BBO00019I INITIALIZATION COMPLETE FOR WEBSPHERE FOR Z/OS CONTROL PROCESS XSDMGR.

START XSACR1, JOBNAME=XSAGNT1, ENV=XSCCELL.XSNODE1.XSAGNT1

$HASP373 XSAGNT1 STARTED
BBO00001I WEBSPHERE FOR Z/OS CONTROL PROCESS XSCCELL/XSNODE1/XSAGNT1/XSAGNT1 IS STARTING.
BBO00238I WEBSPHERE FOR Z/OS CONTROL PROCESS xscell/xsnodexnodeagent IS STARTING.
BBO00221I: WSVR0001I: Server CONTROL PROCESS nodeagent open for e-business
BBO00019I INITIALIZATION COMPLETE FOR WEBSPHERE FOR Z/OS CONTROL PROCESS XSAGNT1.
ADMS0003I: The configuration synchronization completed successfully.
BBO00222I: ADMS0003I: The configuration synchronization completed successfully.
## Address Spaces

**SDSF** STATUS DISPLAY ALL CLASSES  
PREFIX=X*  DEST=(ALL)  OWNER=*  

<table>
<thead>
<tr>
<th>NP</th>
<th>JOBNAME</th>
<th>JobID</th>
<th>Owner</th>
<th>Queue</th>
</tr>
</thead>
<tbody>
<tr>
<td>XSAGNT1</td>
<td>STC01603</td>
<td>XSACRU</td>
<td>EXECUTION</td>
<td></td>
</tr>
<tr>
<td>XSDMGRS</td>
<td>STC01602</td>
<td>XSASRU</td>
<td>EXECUTION</td>
<td></td>
</tr>
<tr>
<td>XSDEMN</td>
<td>STC01601</td>
<td>XSACRU</td>
<td>EXECUTION</td>
<td></td>
</tr>
<tr>
<td>XSDMGR</td>
<td>STC01599</td>
<td>XSACRU</td>
<td>EXECUTION</td>
<td></td>
</tr>
</tbody>
</table>
WAS Configuration on z/OS – Network Deployment Manager

Network Deployment Manager

Diagram showing the configuration of WAS on z/OS, including components like Location Service Daemon, Node agent, JMS server, HTTP internal transport, and various controllers and servants.
z/OS and Distributed – which Admin GUI is which?

In other words – same look and feel
Administration Console

Log in to the console.

User ID:

Login

Note: After some period of inactivity, the system will log you out automatically and ask you to log in again.
Administration Console

Integrated Solutions Console provides a common administrative console for multiple products. The table lists the product suites that can be administered through this installation. Select a product suite to view more information.
Administration Console
START XSACR1,AMODE=64,JOBNAME=XSSR011,ENV=XSCCELL.XSNODE1.XSSR011,XSSR011,REUSASID=YES,PARMS='-Dwas:status.socket=1082'

$HASP373 XSSR011  STARTED

BBOO0001I WEBSPHERE FOR Z/OS CONTROL PROCESS XSCCELL/XSNODE1/XSSR01/XSSR011 IS STARTING.

BBOO0238I WEBSPHERE FOR Z/OS CONTROL PROCESS xscell/xsnode1/xssr011 IS STARTING.

BBOO0222I: WSVR0001I: Server CONTROL PROCESS xssr011 open for e-business

BBOO0019I: INITIALIZATION COMPLETE FOR WEBSPHERE FOR Z/OS CONTROL PROCESS XSSR011.

BBOO0222I: ADMS0003I: The configuration synchronization completed successfully.
Administration Console
## Address Spaces

**SDSF** STATUS DISPLAY ALL CLASSES  
PREFIX=X*  DEST=(ALL)  OWNER=*  

<table>
<thead>
<tr>
<th>NP</th>
<th>JOBNAME</th>
<th>JobID</th>
<th>Owner</th>
<th>Queue</th>
</tr>
</thead>
<tbody>
<tr>
<td>XSSR011S</td>
<td>STC01612</td>
<td>XSASRU</td>
<td>EXECUTION</td>
<td></td>
</tr>
<tr>
<td>XSSR011</td>
<td>STC01608</td>
<td>XSACRU</td>
<td>EXECUTION</td>
<td></td>
</tr>
<tr>
<td>XSAGNT1</td>
<td>STC01603</td>
<td>XSACRU</td>
<td>EXECUTION</td>
<td></td>
</tr>
<tr>
<td>XSDMGRS</td>
<td>STC01602</td>
<td>XSASRU</td>
<td>EXECUTION</td>
<td></td>
</tr>
<tr>
<td>XSDEMN</td>
<td>STC01601</td>
<td>XSACRU</td>
<td>EXECUTION</td>
<td></td>
</tr>
<tr>
<td>XSDMGR</td>
<td>STC01599</td>
<td>XSACRU</td>
<td>EXECUTION</td>
<td></td>
</tr>
</tbody>
</table>
Administration Console

New Application
This page provides links to create new applications of different types.

- New Enterprise Applications
- New Business Level Application
- New Asset
Administration Console
Administration Console
Documentation & Professional Manuals

- WAS Education Assistant
- WAS IBM Redbooks
- WAS Manuals
- WAS General Information
Unit summary

Having completed this unit, you should be able to:

• Describe WebSphere Application Server
• Be familiar with the WAS Administration Console
WebSphere Application Server
Distributed vs. z/OS
Additional material
Similarities

• **Code base**
  - Since V6.0, code base for WebSphere on z/OS same as used on distributed
    - Which is since Mar 2005
  - Includes
    - Portal
    - Process Server
    - etc
  - Has extra code to take advantage of z/OS

• **Things that are the same:**
  - J2EE Specification support
  - Terminology
  - Product and maintenance release dates
  - Administration
J2EE Specification support

- J2EE Applications
  - Written to the specification
  - Will run unchanged in WAS on z/OS
    - No need to recompile
    - IBM Techdoc: Moving Applications to WebSphere on z/OS
Maintenance levels

• **WebSphere on z/OS**
  - Uses same maintenance nomenclature

• **From log on z/OS**
  - BBOM0007I CURRENT CB SERVICE LEVEL IS build level 6.1.0.12 (cf120738.13) release WAS61.ZNATV date 09/25/07 00:03:32.

• **From log on Windows**
  - WebSphere Platform 6.1 [BASE 6.1.0.19 cf190836.04]
z/OS and Distributed – which Admin GUI is which?

In other words – same look and feel.
wsadmin on z/OS and distributed

WZADMIN @ SC55:/WebSphereEd/wzcell/dmgr/DeploymentManager/profiles/default/bin>./wsadmin.sh
–port 7010 –user wzadmin –password xyz –lang jython

WASX7209I: Connected to process "dmgr" on node wzdmnode using SOAP connector; The type of process is: DeploymentManager

WASX7031I: For help, enter: "print Help.help()"

wsadmin>AdminControl.completeObjectName("type=DeploymentManager,*")

'WebSphere:name=DeploymentManager,process=dmgr,platform=common,node=wzdmnode,diagnosticProvider=true,version=6.1.0.12,type=DeploymentManager,mbeanIdentifier=DeploymentManager,cell=wzcell,spec=1.0'

C:\zProducts\was61\AppServer\profiles\Dmgr01\bin>wsadmin -lang jython

WASX7209I: Connected to process "dmgr" on node Dmgr01 using SOAP connector; The type of process is: DeploymentManager

WASX7031I: For help, enter: "print Help.help()"

wsadmin>AdminControl.completeObjectName("type=DeploymentManager,*")

'WebSphere:name=DeploymentManager,process=dmgr,platform=common,node=Dmgr01,diagnosticProvider=true,version=6.1.0.9,type=DeploymentManager,mbeanIdentifier=DeploymentManager,cell=Dmgr01,spec=c=1.0'
Tracing via Admin GUI – z/OS and Distributed
Trace via wsadmin on z/OS and distributed

WASX7209I: Connected to process "dmgr" on node wzdmnode using SOAP connector;  The type of process is: DeploymentManager

WASX7031I: For help, enter: "print Help.help()"

wsadmin>ts = AdminControl.completeObjectName('type=TraceService,process=wzsr01a,*')
wsadmin>AdminControl.setAttribute(ts, 'traceSpecification', 'com.ibm.*=all=enabled')

WASX7031I: For help, enter: "print Help.help()"

wsadmin>ts = AdminControl.completeObjectName('type=TraceService,process=server-1,*)
wsadmin>AdminControl.setAttribute(ts, 'traceSpecification', 'com.ibm.*=all=enabled')

BossLog: { 0096} 2008/09/25 07:10:48.934 01 SYSTEM=SC55 SERVER=WZSR01A PID=0X02010237
./bborjtr.cpp+440412145 ... BBOO0222I: TRAS0018I: The trace state has changed. The new trace state is *=info:com.ibm.*=all.

TRAS0018I: The trace state has changed. The new trace state is *=info:com.ibm.*=all.

[25/09/08 17:17:05:099 EST] 0000002c ManagerAdmin I

TRAS0018I: The trace state has changed. The new trace state is *=info:com.ibm.*=all.
In short…

• From an administration point of view
  ➢ Lots of similarities
    ▪ Admin Gui – same
    ▪ wsadmin – same
    ▪ Terminology - same

• If you have skills as a administrator for WebSphere on Windows or Unix
  ➢ Then those same skills transfer seamlessly to WebSphere on z/OS
So what is different?

• Working on z/OS
• Information about running WAS processes
• Logs
• Threads in the JVM
• Installation
• Configuration
Working on z/OS

- **Windows**
  - Has its user interface – the GUI we’ve all grown to love ;-)  

- **Unix**
  - Has its user interface  
  - Either command line or GUI

- **z/OS**
  - Has its own user interface as well  
  - Several in fact
    - TSO, ISPF
    - Telnet
    - Rational Application Developer for z

- **Bottom line**
  - The WebSphere administrator for WebSphere on z/OS
    - needs to know some TSO, ISPF basics  
    - But does not need to be a z/OS guru
Telnet into z/OS

C:\>telnet wtsc55oe.itso.ibm.com
What WebSphere processes are running?

Windows
Task Manager

Unix
top

Q: Which java process is the DMGR and which is the node agent?

Q: Can you monitor WebSphere processes on other machines?
**On z/OS - the view from SDSF**

<table>
<thead>
<tr>
<th>JOBNAME</th>
<th>StepName</th>
<th>ProcStep</th>
<th>JobID</th>
<th>Owner</th>
<th>Real</th>
</tr>
</thead>
<tbody>
<tr>
<td>WZDEMN</td>
<td>WZDEMN</td>
<td>BBODAEMN</td>
<td>STC07908</td>
<td>WZDMN</td>
<td>6785</td>
</tr>
<tr>
<td>WZDMGR</td>
<td>WZDMGR</td>
<td>BBOCTL</td>
<td>STC07906</td>
<td>WZDMCR1</td>
<td>70T</td>
</tr>
<tr>
<td>WZDMGRS</td>
<td>WZDMGRS</td>
<td>BBOSR</td>
<td>STC07910</td>
<td>WZDMSR1</td>
<td>101T</td>
</tr>
<tr>
<td>WZNODE5</td>
<td>WZNODE5</td>
<td>BBOCTL</td>
<td>STC07909</td>
<td>WZASCR1</td>
<td>43T</td>
</tr>
<tr>
<td>WZSR01A</td>
<td>WZSR01A</td>
<td>BBOCTL</td>
<td>STC07923</td>
<td>WZASCR1</td>
<td>67T</td>
</tr>
<tr>
<td>WZSR01AS</td>
<td>WZSR01AS</td>
<td>BBOSR</td>
<td>STC08092</td>
<td>WZASSR1</td>
<td>86T</td>
</tr>
<tr>
<td>WZSR01AS</td>
<td>WZSR01AS</td>
<td>BBOSR</td>
<td>STC08065</td>
<td>WZSRSR1</td>
<td>87T</td>
</tr>
<tr>
<td>WZSR01AA</td>
<td>WZSR01AS</td>
<td>BBOSR</td>
<td>STC08066</td>
<td>WZSRSR1</td>
<td>85T</td>
</tr>
<tr>
<td>WZSR64A</td>
<td>WZSR64A</td>
<td>BBGCTL</td>
<td>STC07928</td>
<td>WZASCR1</td>
<td>88T</td>
</tr>
<tr>
<td>WZSR64AS</td>
<td>WZSR64AS</td>
<td>BBGSR</td>
<td>STC08091</td>
<td>WZASSR1</td>
<td>255T</td>
</tr>
</tbody>
</table>

- **Deployment Manager**
- **Node Agent**
- **Server**
  - **Control**
  - **Servants**
  - **Adjunct**
- **Daemon**

**Via SDSF**

- with a good naming convention, able to determine which STC is DMGR etc
- Able to view WebSphere STCs running on any LPAR in the Sysplex
WebSphere logs

• **On distributed**
  - SystemOut.log
  - SystemErr.log
  - native_stderr.log
    - Verbose Garbage collection

• **On z/OS**
  - What is normally written to these files is written to z/OS spool
WebSphere logs on z/OS

- This is default setup
  - Custom properties can be used to write old log data from spool to a file
    - ras_stderr_ff_interval, ras_stdout_ff_interval
- Can change JCL so that SystemOut and SystemErr
  - Are written to files
  - But no rolling capability
  - See:
    - http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD101087
Threads in the JVM

• On Distributed
  ➢ Can set number of threads to any value

• On z/OS
  ➢ Number of threads determined by workload profile selected

• Workload Profile
  ➢ Set via wsadmin or adminconsole:
    ▪ Servers >> Application Servers >> server_name >> ORB Service>> Advanced Settings >> "Workload Profile“
  ➢ Can be set to:
    ▪ ISOLATE (1 thread)
    ▪ NORMAL (3 threads)
    ▪ CPUBOUND (# of CPs-1, minimum of 3)
    ▪ IOBOUND (Number of CPs*3, Min=5, Max=30)
    ▪ LONGWAIT (40)
  ➢ V7 – provides property to set custom value

• WebSphere for z/OS doesn't need threads as placeholders for work
  ➢ WLM queues are used for that
Installation

• On Windows and Unix
  ➢ Typically run WebSphere supplied install GUI or install script
    ▪ Which installs the software into some specified location
• On z/OS
  ➢ All z/OS software installed via z/OS mechanism called SMP/E
    ▪ SMP/E has been in use for over 20 years for software install
    ▪ Typically done by your friendly neighbourhood z/OS System programmer
  ➢ Can have different versions of WebSphere installed at the same time
    ▪ In fact can have different maintenance levels of a WebSphere version installed at the same time
    ❖ And in use
Configuration on Distributed

- **On Windows and Unix**
  - Can use GUI
  - Or run WebSphere supplied command:
    - manageprofiles
    - To create profiles for nodes etc

- **On z/OS**
  - Different process
  - If you’re going to get anywhere with this then you...
    - Need to accept it is a different process
    - Be willing to learn

- **If you are new to z/OS**
  - You need someone with z/OS experience to assist you
Configuration on z/OS

- **Process to build a cell:**
  - Use TSO/ISPF or Windows Eclipse based tool
  - In which supply various values such as:
    - Started task names
    - Security related userid’s
    - TCPIP port numbers
    - Cell, node and server names
  - Generates small number of batch jobs
  - Run batch jobs to create cell

- **Keep sense of perspective**
  - Generally you are not building WebSphere cells everyday
  - Most work in WebSphere around day to day administration
    - Installing applications
    - Defining resources
    - Helping application developers to solve their problems etc etc
Separation of product and configuration data

• On distributed typically
  – Configuration data that defines a cell
    ▪ Stored under config sub-directory
    ▪ Which is located under root directory where product code installed

• On z/OS
  – Product code stored in one file
  – Configuration data stored in different file

• Advantages
  – Easy to manage multiple versions of WebSphere
    ▪ And even multiple versions at multiple maintenance levels
  – Easy to change a cell to run on a new maintenance level
    ▪ And to fall back to previous maintenance level
And now for the big difference...

- **A WebSphere server on Windows and Unix**
  - Is one JVM

- **On z/OS**
  - A WebSphere server split into two components
  - Control Region
    - A JVM
    - Handles receiving requests and sending the response
  - One or more Servant Regions
    - A JVM
    - Where the application code runs
Why is server split asunder?

• **Control region**
  - Runs authorised code
    - Has access to restricted z/OS functionality
  - Handles HTTP/S communications

• **Servant region**
  - Does not run authorised code
    - Just application code
  - Means application code cannot get access to authorised z/OS services
  - Prevents application code being used to attack the system
The Control Region is watching

- **Control region**
  - For each request
    - Records time dispatched
  - If no reply within specified timeout period
  - Kills the servant region
  - WAS V7
    - Introduces more advanced options
- **Why does request not complete within timeout period?**
  - Typically some backend system not responding
  - Could also be that application is looping
- **On distributed – what would happen in such a case?**
  - Nothing until someone intervenes
- **On z/OS**
  - Servant cancels results in automatic restart of new one
  - If second servant already running, then it take new requests immediately