WebSphere 5.0 Runtime Architecture

WebSphere 5.0 Skills Transfer

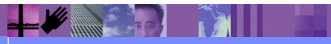
WebSphere. software

e-business software

Updated 04/04/2003

Objectives

- **You will learn to administer WebSphere Application server Network Deployment**
  - add a node to a cell
  - remove an existing node from a cell
  - You will learn to describe file synchronization
- **You will learn to use the Admin clients**
  - Web-based Admin console
  - wsadmin scripting utility
- **You will learn to write ANT scripts**
- **You will learn to use other command line tools**
  - startserver, stopserver, startManager, stopManager....

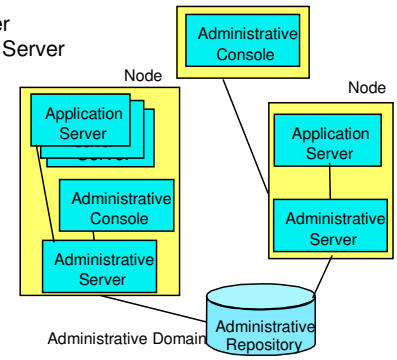


Agenda

- **WebSphere 3.5/4.0 Review**
- **WebSphere 5.0 System Management**
  - Distributed Administration
  - File Based Repository Structure and Handling
  - Distributed Process Discovery
  - Add/Remove Node from a cell
- **Administrative Console**
- **Scripting with wsadmin**
- **Automating build and deployment using ANT**
- **Command line tools (startServer, etc.)**
- **Summary**

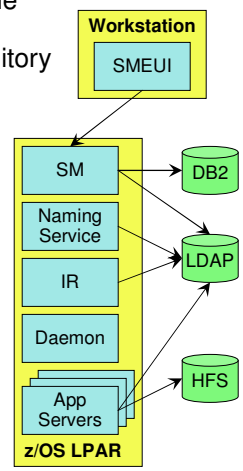
WebSphere 3.5/4.0 Review - Distributed

- **AE - Administrative Model**
  - Administrative Server (Admin Server) installed on all nodes in the Domain
  - Configuration stored in the Administative Repository
    - Relational Database
    - One Respository per domain
  - Admin Server must be up and running
    - Start/Stop application is via Admin Server
    - App Server gets config data from Admin Server
    - Admin Server reads/converts data from repository to pass to App Server
    - Naming Server inside Admin Server
  - Thick client - Java GUI Admin console
  - XMLConfig
  - WebSphere Control Program (WSCP)
- **AEs – Administrative Model**
  - Config data in flat file - server-cfg.xml
  - Thin Client - Browser-based
  - Single process contains everything



WebSphere 4.0 Review – z/OS

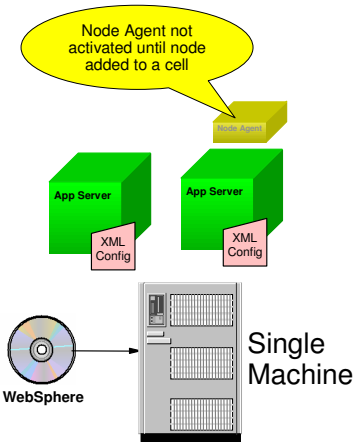
- **Administrative Model**
  - System Management Server installed on multiple systems in one or more Sysplex
  - Configuration stored in the Administative Repository
    - Relational Database (DB2)
    - One Respository per Node
      - Node = AE Domain
  - Servers must be running
    - System Management Server
    - Naming Server
    - Interface Repository
    - Daemon
  - App Servers started using
    - MVS Console
    - SMEUI
  - Thick client – Java SMEUI
  - System Management Scripting API (SMAPI)



WebSphere 5.0 Distributed System Management

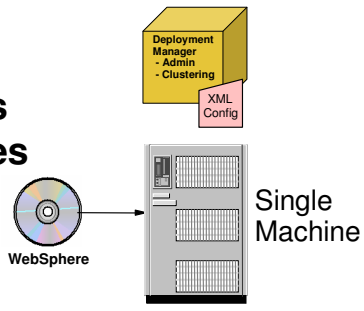
Base Application Server

- **Installed on Single Machine (System/LPAR)**
  - Separate install on Distributed
  - Setup through config on z/OS
- **One or more Servers**
  - J2EE 1.3 Compliant
  - Web Services
- **Configuration files in XML**
- **Admin clients modify XML**
  - Browser Based (AdminConsole)
  - Scripting (wsadmin)
- **Node Agent**
  - Installed but not configured
  - Configured during addNode



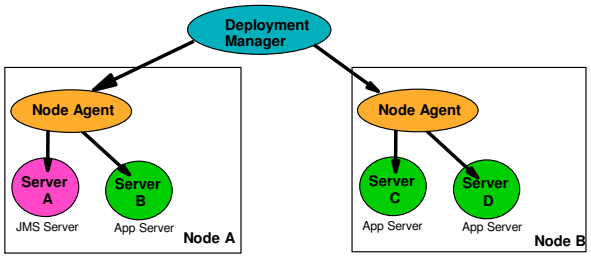
Network Deployment

- **Required to perform Distributed Administration**
- **Installed on Single Machine (System/LPAR)**
  - Separate install on Distributed
  - Setup through config on z/OS
- **Manages nodes**
  - Only after addNode
- **Manages master config files**
- **Manages application binaries**
- **Synchronizes with nodes**



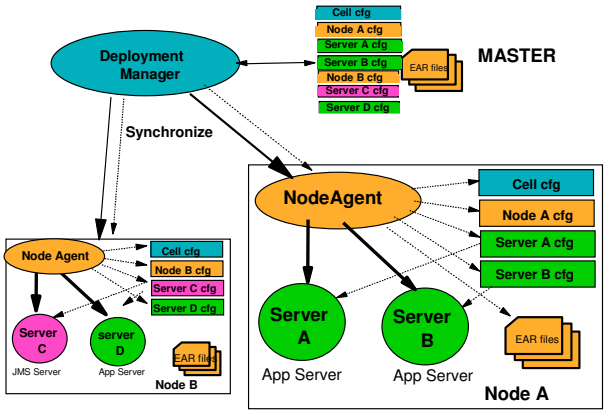
Distributed Administration

- **Basic Topology Illustration**
  - Deployment Manager
  - Two nodes
- **Terminology**
  - Managed Server – managed by Node Agent & Deployment Manager
  - Node - Set of Managed Servers and their managing Node Agent
  - Cell - Aggregation of Nodes managed by a Deployment Manager



Configuration / Application Data

- **Configuration files - Each managed process, Node Agent, Deployment Manager starts with it's own configuration file**
- **Cell Contains the MASTER configuration and application files**
  - Changes made at node agent or server level are temporary
  - Will be overridden by the MASTER configuration at the next synchronization (update)



Configuration/Application Data

- **Administration points within a Cell**
  - Deployment Manager - manage everything under the cell - recommended
  - Node Agent - manage everything under the node, not the cell
  - Managed Server - manage the server process configuration, not the node or the cell
  - Can have the Admin client attach to any of the above process
- **Deployment Manager contains the MASTER copy of the configuration/application files**
  - Admin Client programs are used to modify configuration settings
  - Individual nodes and servers synchronize the files with the master configuration files (repository)
  - Only changes made at the cell level are permanent
    - Config changes made at the Node Agent or Server level are temporary
  - At next data update time, the master data is pushed down to the Nodes
  - Deployment Manager checks in/out the configuration files from the master repository
- **Node Agent receives updates of configuration/application data from Deployment Manager at each file synchronization**

When does File Synchronization Occur?

- **File Synchronization settings can be set in the Admin Console**
- **If Automatic Synchronization flag is ON synchronization occurs:**
  - Periodically. The time interval between the synchronizations can be set by the user.
  - When the Node Agent starts up and the Deployment Manager is already running
  - When the Deployment Manager starts up and the Node agent is running
- **By default, Automatic Synchronization flag is on and the Synchronization interval is 60 seconds**
- **End user can force explicit synchronization**
  - using wsadmin or Admin Console
- **If Startup Synchronization is on**
  - Refers to startup of an Application Server
- **Recommendation: Define your file synch strategy**
  - Production – large interval or explicitly force synchronizations
  - Development/Test – defaults may be OK

What is Synchronized?

- **Deployment Manager maintains a repository of file names**
- **MBeans are instrumented to notify repository of changes**
  - Changes to XML files through WebSphere Admin tools are propagated
  - Changes to XML files made with generic tools (ie, Notepad) are not recognized unless you manually force a full synch, or use the API to inform the repository of changes.
  - MBean specifics in JMX lecture
- **At synchronization:**
  - files listed in the repository are checked against file system CRC's
  - Only changed files are synchronized to the nodes
  - Changes are marked for synchronization
- **Included in the Deployment Manager's Repository are:**
  - Configuration Files
  - Application Binaries

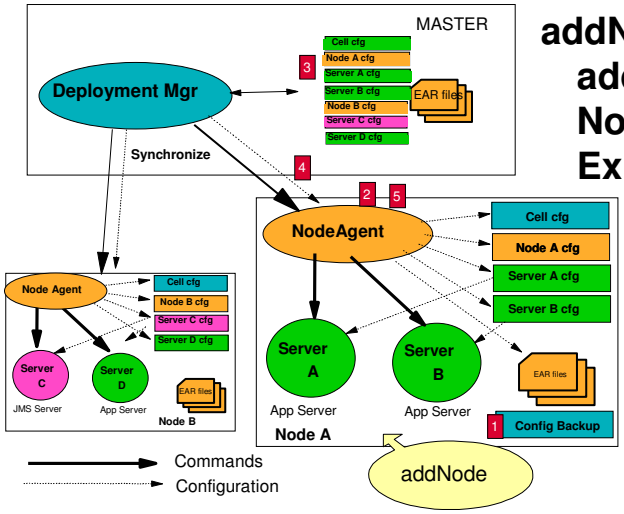
Distributed Process Discovery

- **Deployment Manager, Node Agents & Managed Servers**
  - As they come up try to discover each other
  - Establish communication channels between themselves
- **Each has its own configuration/application data to start**
- **Example:**
  - Node Agent and Deployment Manager not running
  - Deployment Manager comes up, tries to contact Node Agent but no response
  - Node Agent comes up, tries to contact Deployment Manager and establishes a communication channel
  - Depending upon settings, config data may be synchronized

Distributed Process Discovery

- **Order of server startup is not important in most cases**
  - Node agent can be running while deployment manager is not and vice versa
  - Deployment manager can be running while a managed server is not and vice versa
  - However (Distributed Only)
    - An application server can be started only if the Node Agent is running
    - The Node Agent hosts the Location Service Daemon that is needed by the application server.
    - This assures enterprise application versions on multiple app servers are in sync

addNode Utility

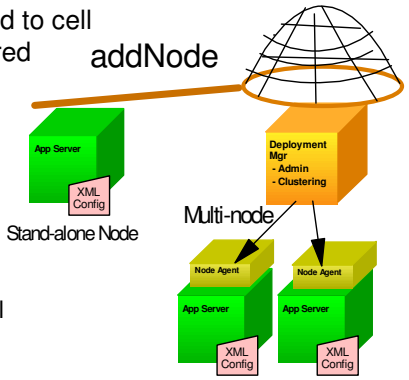


addNode Utility - adds a Standalone Node to an Existing Cell

1. Backs up cfg files
2. Creates NodeAgent
3. Copies cfg files to Cell
4. File Sync for the new node
5. Launches the node agent

addNode Utility

- **addNode <cell host> <cell port> [options]**
  - Mandatory parameters: Deployment Manager host and port
- **Application handling during addNode**
  - By default applications not propagated to cell
  - Therefore they are no longer configured for the newly managed servers
- **-includeApps option**
  - addNode attempts to include existing applications from the new node
  - If application with the same name already exists in the cell
    - a warning is printed
    - application will not be installed in cell

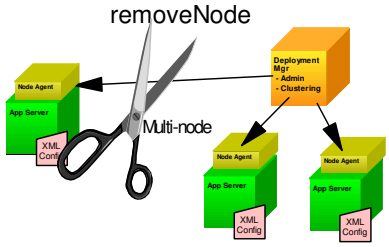


addNode Utility – Security Considerations

- **Newly added node inherits security settings from the cell**
  - Add a secure Node to an unprotected Cell
    - Node will be unprotected
  - Add secure Node to secure Cell
    - Use addNode with -username and -password options
    - Differences in security configuration settings from Cell override settings of Node
  - Add unsecured Node to secure Cell
    - Use addNode with -username and -password options
    - Node will be secure
- **Steps to enable security on ND after nodes added:**
  - Ensure you have your RACF definitions correctly setup (z/OS only)
  - Change the configuration and syncronize config files with nodes
  - Stop the Deployment Manager and restart it (so it starts up in secure mode)
  - Stop and restart each of the Nodes (so they switch to using the new security config)
  - Complete documentation in InfoCenter

removeNode: Detach Node from a Cell

- **Execute command line utility "removeNode" from any node to detach itself from a Cell**
  - Stops all running server processes in the Node
  - Backed-up original base configuration will be restored
  - Lose all configuration changes done after joining a Cell
  - Uninstall will also makes use of the same command
  - Syntax: removeNode [options]
- **Alternative to avoid loss of configuration information:**
  - Don't execute removeNode
  - Set server's standalone property to true
  - AppServer will retain the existing configuration
  - It will no longer participate in the distributed Admin network



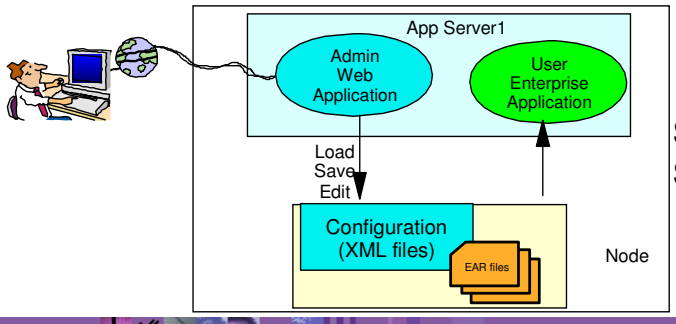
Administrative Console (Admin Console)

WebSphere 5.0 Admin Console Overview

- **Browser based Admin Console was introduced in the WebSphere 4.0 AEs**
- **In WebSphere 5.0, Browser based Admin Console is expanded to manage entire Cell**
- **Admin Console is standard J2EE 1.3 Web application**
  - The Admin Web Application loads, edits and updates the configuration (XML) files
- **Supported Browsers:**
  - Microsoft Internet Explorer 5.5, 6.0 (or later)
  - Netscape Navigator 4.7.x (varies by platform)
    - Netscape 6.1 Browser is not supported, but is expected to work

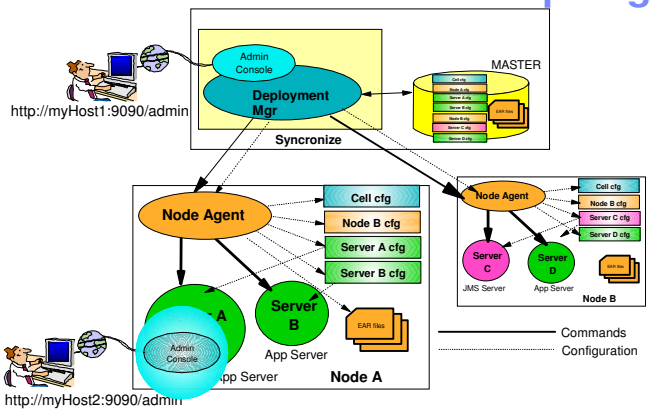
How does it work?

- **The Admin Console Web Application**
  - Runs within an Application Server instance
  - Performs configuration and operational changes
- **Admin Console Web Application access**
  - http://localhost:9090/admin
  - http://myHost.myCompany.com:9043/admin



Standalone Single Server

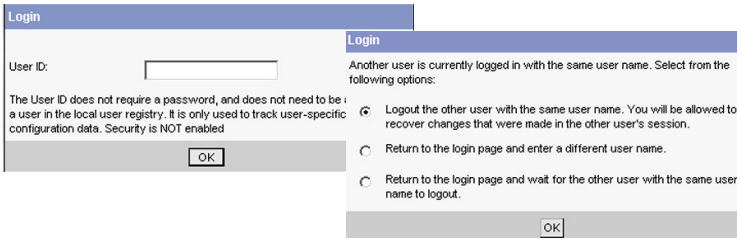
Admin Console in a Multi-node Topology



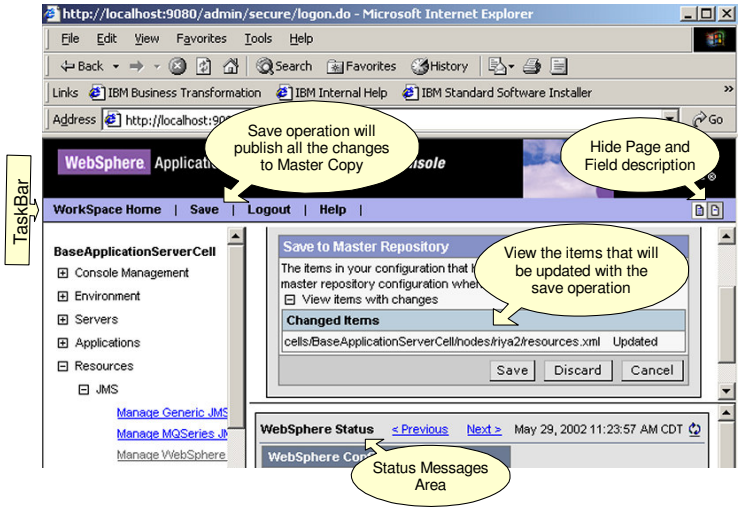
- **Admin Console always configured on Deployment Manager**
- **Optionally can be configured on an Application Server**
  - If configured as stand alone, local changes remain in effect
  - If not configured stand alone, local changes overridden on file sync

Starting the Admin Console

- **Admin Console gets installed**
  - During WebSphere 5.0 installation (Distributed)
  - During Configuration (z/OS)
- **Start the server where Admin Console is installed**
  - Base Application server, issue → **startserver server1**
  - Deployment Manager, issue → **startmanager**
- **Access via web Browser "http://<hostname>:9090/admin/" (9043 for SSL)**
- **No Password required if Global security is not enabled**
- **UserName is used to track and save user-specific configuration data**
  - Will be able to recover from unsaved previous session changes
  - user Workspace: <was50-root>/wstemp/USER/workspace



WebSphere 5.0 Administrative Console



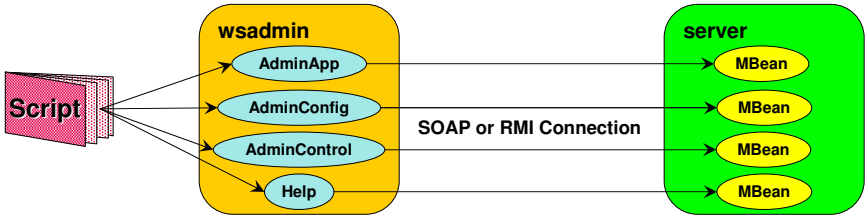
Scripting with wsadmin

wsadmin - Overview

- **New scripting interface called “wsadmin"**
- **wsadmin offers a number of Advantages**
  - Same scripting tool for all versions of WebSphere v5.0
  - Based on the Bean Scripting Framework (BSF)
    - Current supported languages for wsadmin - more to come
      - jacl - Java Command Language based on Tcl scripting
  - Robust scripting features and programming model very similar to Java
  - Utilize existing programming and scripting skills and technology
- **BSF Overview**
  - Architecture for easily incorporating scripting in Java applications & applets
  - Scripting language are commonly used to augment an application's function or to script together a set of application components to form an application
  - Applications independent and not bound to a single scripting language
  - Different language scripts can access Java objects (wsadmin)
  - Java programs can evaluate and access results from scripts in other languages
  - BSF supports interoperability between Java and BML, JScript, Netscape Rhino (JavaScript), NetRexx, PerlScript, VBScript, Jacl, JPython, and LotusXSL

wsadmin - How does it work?

- **wsadmin provides interface to Java objects for access by scripts**
  - Objects in wsadmin communicate with MBeans (JMX management objects)
  - MBeans run in servers and are accessed using SOAP or RMI connectors
- **Objects perform different operations**
  - AdminConfig - create or change the WebSphere configuration
  - AdminApp - install, modify, or administer applications
  - AdminControl - work with live running objects (e.g. query state, set trace on)
  - Help - display general help information and details about running MBeans
- **Separation between Configuration and Control**



Working with wsadmin

- **Running wsadmin.bat or wsadmin.sh**
  - Located in <WAS_ROOT>\bin
- **wsadmin [-c command | -f scriptfile]**
  - [-p propertiesfile]
  - [-profile scriptfile]
  - [-h(elp)]
  - [-?]
  - [-lang language]
  - [wsadmin_classpath classpath
  - [-conntype SOAP [-host host_name] [-port port_number] [-user userid] [-password password] |RMI [-host host_name] [-port port_number] [-user userid] [-password password] | NONE]
  - [script parameters]
- **Specify no parameters for interactive prompt**
  - Script language for prompt based on value in property file
- **Profile can customize scripting environment**
  - with predefined or helper commands
  - available for use by main script

wsadmin - Operations & Functions – Examples

- **AdminConfig Commands**
  - createObject
  - create
  - remove
  - list
  - show
  - modify
  - getid
  - contents
  - parents
  - attributes
  - types
  - save
  - reset
  - hasChanges
  - queryChanges
  - setSaveMode
- **AdminApp Commands**
  - install
  - install interactive
  - options
  - taskinfo
  - list
  - move
  - uninstall
  - modify
  - show
  - listmodules
  - export

wsadmin - Operations & Functions – Examples

- **AdminControl Commands**
  - getHost
  - getPort
  - getType
  - reconnect
  - queryNames
  - getMBeanCount
  - getDomainName
  - makeObjectName
  - getDefaultDomain
  - getMBeanInfo
  - isInstanceOf
  - isRegistered
  - getAttribute
  - setAttribute
  - invoke
  - isAlive
  - trace
- **Help commands**
  - attributes
  - operations
  - constructors
  - description
  - notifications
  - classname
  - all
  - help
  - AdminControl
  - AdminConfig

wsadmin - Comparison with 4.0 Scripting

- **wscp 4.0 (Distributed)**
  - wscp> EnterpriseApp install /Node:mynode/ c:\temp\myapp.ear -defappserver /Node:mynode/ApplicationServer:Default_Server/
- **smapi 4.0 (z/OS)**
  - CB390CFG("-action 'processearfile' -xmlinput 'inputprocessearfile.xml' -input 'tempin' -output 'processearfile'")
  - The input xml file contains the real description of what to do
- **wsadmin 5.0**
  - wsadmin>$AdminApp Install c:/temp/myapp.ear -conntype NONE
- **Multiple ways to execute the command**
  - wsadmin -c "$AdminApp Install c:/temp/myapp.ear -conntype NONE"
  - wsadmin -lang jacl -f mycommand.jacl

wsadmin - Extra Info

- **Commands are case-sensetive**
- **Configuration changes not persisted until "save" call**
  - Upon "save" call, validation procedure verifies updates
  - Validation errors throw exceptions (e.g. two servers with same name)
- **Multiple client updates (wsadmin and/or Admin Console)**
  - Overlapping updates to same file will be detected
  - Exception thrown on Save call
  - No changes will be persisted (override with force option)
- **Performance of wsadmin modes**
  - wsadmin –c “command” – very slow - use only for single commands
  - wsadmin (prompt mode) - if interactively entering multiple comands
  - wsadmin –f “scriptfile” - best performance, use for repeated tasks
    - Call "save" in file periodically to persist configurations updates
    - Avoids no changes being persisted should an exception occur
- **Migration is documented in the InfoCenter**
  - WSCP and XMLConfig → wsadmin
  - smapi → wsadmin

Configuration Repository Overview

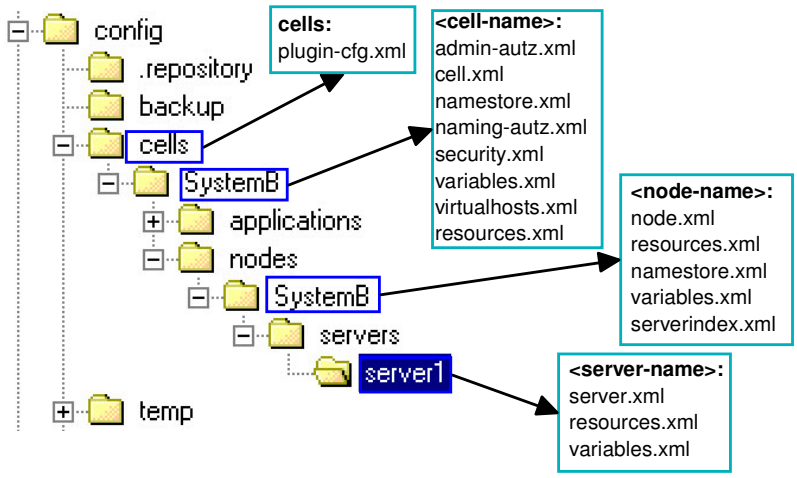
Configuration Repository – Overview

- **Configuration data contained in XML documents**
  - Set of hierarchical directories under <was_root>/config
- **Directories are in three tiers: cell, node, server**
  - Each contains documents specific to that scope
  - Examples:
    - cell level – contains clusters.xml defining all the clusters in the cell
    - server level – contains server.xml defining the attributes of the server
- **Admin clients used to modify configuration**
  - Web Based Admin (Admin Console)
  - WebSphere Scripting (wsadmin)

Configuration Repository – Same Name Files

- **Different levels of the hierarchy contain some documents with the same name**
- **Handling of these documents differ**
  - Some are logically combined
  - Some are independent
- **When logically combined:**
  - The “most specific” value takes precedence – i.e. what is defined at server level overrides node and node overrides cell
  - Result of final combination applied at the server level
  - Examples files:
    - variables.xml
    - resources.xml
- **When not combined:**
  - File contents apply to the level where found (server, node, cell)
  - Example: namestore.xml

Configuration Repository - Structure



Automating the Build Process using ANT

Why automate build and deployment?

- **Speed the development process**
  - Tools speed up the process and reduce the risk of errors, leading to a repeatable, reliable process that is essential through multiple development cycle iterations.
- **Minimize risk of error caused by manual process**
- **Test the production deployment process**
- **A build process alone will speed the migration of software from one environment to another**
- **Minimize need for scarce resources to move code changes into test environments**

Using ANT

- **What is ANT**
  - "Another Neat Tool"
  - ANT is a modernized make facility
  - ANT uses XML scripts instead of makefiles
  - ANT is extensible
- **Use ANT to:**
  - compile code
  - construct EJB jars, wars, ears etc.
  - launch EJBDeploy
  - install applications
  - run JUnit tests
- **<was_root>\bin\ws_ant.bat can be used to run ANT scripts**

ANT tasks

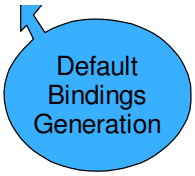
- **A task is a piece of code that can be executed**
- **ANT provides a large number of built-in tasks**
  - jar - jars a set of files
  - Exec - executes a system command
  - javac - Compiles java source
  - mkdir - makes a directory
- **WebSphere provides additional ANT tasks.**
  - **wsInstallApp -** enables you to install a new application into a WebSphere Server or Cell
  - **wsUninstallApp -** enables you to uninstall an existing application from a WebSphere Server or Cell
  - **wsListApps -** lists all the applications installed on a WebSphere Server or Cell
  - **wsStartServer -** enables you to start a standalone server instance
  - **wsadmin -** executes the WebSphere command-line administration tool with the specified arguments
  - **wsejbdeploy -** executes the WebSphere EJB Deploy tool on the specified Jar file with the specified options

ANT Examples - Compiling with ANT

```
<project name = "mycompile" default="compile" basedir=".">
<property name="SRC" value="src"/>
<property name="IMAGE" value="classes"/>
<target name="init">
<!-- Create the time stamp -->
<tstamp/>
<mkdir dir="${IMAGE}"/>
</target>
<target name="compile" depends="init">
<javac srcdir="${SRC}"
destdir="${IMAGE}"
extdirs="C:/WebSphere/AppServer/lib"/>
</target>
</project>
```

ANT Examples - Installing an Application

```
<project name = "myinstall" default="install" basedir=".">
<taskdef name="wsInstallApp"
        classname="com.ibm.websphere.ant.tasks.InstallApplication"/>
 <property name="APPLICATION.EAR" value="MyBankCMRQL.ear"/>
 <target name="install" description="Installs the app">
   <wsInstallApp
  ear="C:/LabFiles50/ANTlab/bld/${APPLICATION.EAR}" options="-
  deployejb -usedefaultbindings -defaultbinding.force -
  defaultbinding.ejbjndi.prefix ejb/MyBank
 -defaultbinding.cf.jndi eis/jdbc/MyBank_CMP
 -defaultbinding.cf.resauth PerConnFact"/>
  </target>
</project>
```



Starting & Stopping Servers and Command Line Tools

Starting and Stopping Servers - Overview

- **Base Application Server**
  - Start/Stop using scripts
- **Network Deployment**
  - Start/Stop from Admin Console
    - Application Servers
    - JMS Server
  - Start/Stop using scripts
    - Deployment manager
    - Node Agent
    - Optionally Application Servers and JMS Server
- **z/OS**
  - MVS start and cancel task commands normally used
  - Scripts & Admin Console also work

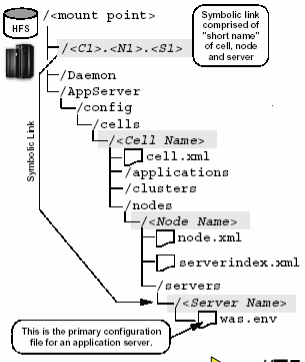
Starting and Stopping Servers - Base

- **startServer.sh <server> options**
  - <server> is the name server to start
    - Server name is used to find configuration directory for the server
  - example: startServer.sh server1
    - reports process ID to command prompt.
    - logs in file <WAS_ROOT>\logs\server1\startServer.log
  - Creates new JVM, reads configuration and for server and launches the server
- **stopServer.sh <server> options**
  - example: stopServer.sh server1
    - logs in file <WAS_ROOT>\logs\server1\startServer.log
  - Creates new JVM to read configuration and send message to server to shutdown
  - By default, the stopServer utility does not return control to the command line until the server completes shutting down

Starting and Stopping Servers - ND

- **Starting and Stopping the Deployment Manager**
  - <WASND_Root>\bin\startManager.bat
    - logs in file <WASND_Root>\logs\dmgr\startServer.log
  - <WASND_Root>\bin\stopManager.bat
    - logs in file <WASND_Root>\logs\dmgr\stopServer.log
- **Starting and Stopping the Node Agent**
  - <WAS_Root>\bin\startNode.bat
    - logs in file <WAS_Root>\logs\nodeagent\startServer.log
  - <WAS_Root>\bin\stopNode.bat
    - logs in file <WAS_Root>\logs\nodeagent\stopServer.log
- **Starting and Stopping application servers**
  - <WAS_Root>\bin\startServer <server>
    - logs in file <WAS_Root>\logs\<server>\startServer.log
  - <WAS_Root>\bin\stopServer <server>
    - logs in file <WAS_Root>\logs\<server>\stopServer.log
- **startServer.log and stopServer.log report the results of the script**
  - For failure cases, normally need to check SystemOut.log in same directory
  - SYSPRINT for z/OS

Starting and Stopping Servers – z/OS



- **HFS Structure**
  - was.env file contains server definition
    - –This is needed when starting the server
    - –Located in server's directory in config tree
  - Symbolic link
    - –Points to server's directory in config tree
    - –Composed of shortnames for cell, node, server
- **JCL Start Procedure**
  - Pass symbolic link on ENV parameter
  - Allows one JCL proc for all servers under the same mount point

JCL Procedure

```
//T5ACR   PROC  ENV=<cell>.<node>.<server>, ……
//     SET ROOT='/<mount point>
       :
//BBOENV  DD  PATH='&ROOT/&ENV/was.env'
```

MVS Start Cmd

```
S T5ACR,JOBNAME=S1,ENV=C1.N1.S1
```

Starting and Stopping Servers – z/OS

- **Starting and Stopping the Deployment Manager**

  s t5acr,jobname=t5srvndc,env=T5CELLND.T5NODEND.T5SRVND
  c t5srvndc

- **Starting and Stopping the Node Agent**

  s t5acr,jobname=bbon001,env=T5CELLND.T5NODEA.BBON001
  c bbon001

- **Starting and Stopping application servers**

  s t5acr,jobname=t5srv1c,env=T5CELLND.T5NODEA.T5SRV1
  c t5srv1

Command Line Tool

- **dumpNameSpace**
  - Displays JNDI namespace and entries
- **ivt (Installation Verification Tool)**
  - Syntax: ivt <hostname> <port#> (ivt myhost 9080)
- **JspBatchCompiler**
  - Pre-compile JSPs in a web module
- **mb2mdb (Distributed only)**
  - Convert a WebSphere 4.0 EE message bean into a message driven bean
- **tperfviewer**
  - Tivoli Performance Viewer (previously known as Resource Analyzer)
- **syncNode**
  - Forces synchronization between the node and the Deployment Manager
- **versionInfo**
  - Provides IBM WebSphere Application Server Version Report. Syntax: versioninfo.bat
- **serverStatus**
  - Retrieves server status. Syntax: serverStatus <server name>
- **backupConfig**
  - utility to back-up configuration of your node to a file
- **restoreConfig**
  - utility to restore the configuration of your node after backing it up using the backupconfig command

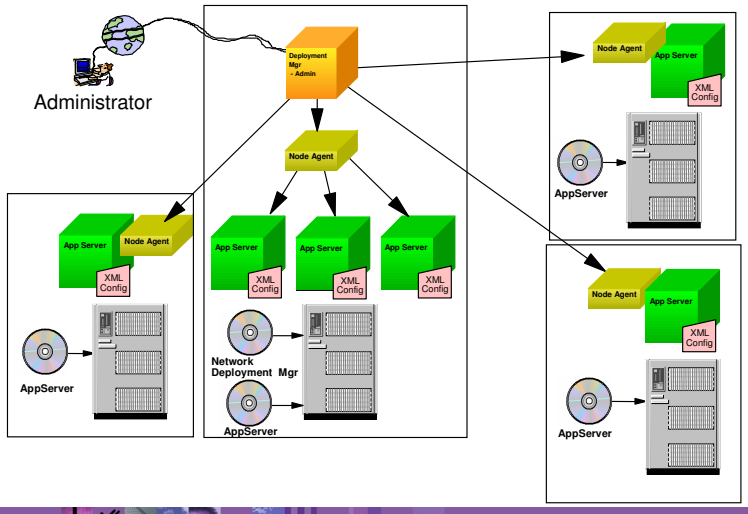
Problem Determination

Problem Determination

- **Node Agent and Deployment Manager must be able to resolve each other's IP addresses by host name**
- **Not Synchronizing?**
  - Check the logs.
  - Trace com.ibm.ws.management.sync.*=all=enabled
  - Stop and re-start node agent
- **Common issues:**
  - Forgot to start nodeagent or deployment manager
  - IP / DNS / gateway / network settings

Summary

Network Deployment Topology



Summary of New Systems Management

- Convergence of System Management Model across the WebSphere Family of Application Servers
- Distributed administration model
- File based repository - servers no longer dependent on central repository to run
- Browser based Administrative Console
- Scripting with wsadmin
- Automating build and deployment using ANT