



IBM Software Group

IBM® WebSphere® Application Server V6

System Management

High Level Architecture



@business on demand.

© 2005 IBM Corporation
Updated March 1, 2005

This presentation will focus on the system management architecture of IBM WebSphere Application Server V6.

Goals

- Describe IBM WebSphere Application Server V6 System Management architecture

- Pre-requisites:
 - ▶ Basic understanding of WebSphere Application Server V6 architecture, packaging, topology and terminology - covered in other presentations



The goals of this presentation are to describe the architecture. Knowledge of the architecture forms the base of knowledge about system management.

Agenda

- System administration tasks
- System Management Architecture
 - ▶ Stand-alone Application Server
 - ▶ Network Deployment Cell
- Web Server definitions in your topology
- Comparison of V5 and V6 System Management tasks and topics



This presentation will cover some typical System Administrator tasks, high level System Management architecture, the new Web Server definitions and a comparison of V5 and V6.

System Administration Tasks

- Install product, understanding the environment and topology
- Make configuration changes
 - ▶ Manage topology (Stand-alone Application Server or Network Deployment cell)
 - Manage/Create WebSphere processes
 - Manage Java™ 2 Enterprise Edition (J2EE) applications
- Make operational changes
 - ▶ Start/stop WebSphere processes, applications, services, etc.
- Problem Determination



System administrators tasks include learning the environment and topology of WebSphere, making configuration changes to the topology, making operational changes, and problem determination.

The different topology options for Stand-alone server environment as well as Network Deployment Cell are covered in topology presentation in the Big Picture and overview section. A good understanding of the topology, its possibilities and high level architecture will be a good foundation to understand V6 System management functions.

System Management Features

- Based on Java™ Management Extensions (JMX) to allow easier integration with 3rd party System Management vendors
 - ▶ Administrative clients use the same JMX interface that is exposed to 3rd party vendors
- V6 System Management model is based on V5
 - ▶ Will reduce the system administration learning curve from V5 to V6
- Common terminology, topology, programming and system management model between Distributed and z/OS™ platforms
 - ▶ While continuing to exploit z/OS Sysplex capabilities



Version 6 system management is built on the Java Management Extensions (JMX) framework that was introduced with Version 5. The use of JMX provides a number of benefits, including management of WebSphere environment in a consistent manner for all topologies, using multiple protocols, like SOAP and RMI/IIOP. Resources within WebSphere managed processes are represented by individual management interfaces implemented as JMX MBeans (referred to as Managed Beans in the JMX specification).

The V6 System administration model is very similar to V5. This will greatly reduce the learning curve for existing V5 users. This was one of the main considerations of the V6 System management design.

All V6 supported platforms use the same System management infrastructure, built around the JMX framework. For System administrators, this benefits them where they have the same management model and client user interfaces, independent of the platform. In addition, where a specific platform has additional capabilities (like Work load management or Security in z/OS), those are reflected back in the System management model and exposed through the administration clients, where needed.

New Features – At a Glance

- Support for J2EE 1.4 specification
 - ▶ JMX 1.2, JSR 077 and JSR 088
- Configuration Archives
 - ▶ Ability to export and import configurations
- Fine-grained Application update
- Improved Administrative Console including integrated of Tivoli™ Performance Viewer and IBM HTTP Server management
- Expanded “wsadmin” command line tasks
- Support for extensible Server types
 - ▶ Web Servers and Generic servers
- Additional Resource Scopes
 - ▶ At Cluster level and application level
- Support for mixed-version cells (V5 and V6)



There are several new features and enhancements in the System Management area listed here. Each of these topics are covered in the details in various System management and other presentations.

Section

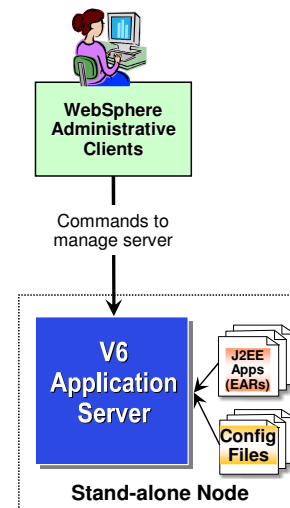
System Management Architecture: Stand-alone Application Server Topology



This section describes a stand-alone Application Server topology.

Stand-alone Server: System Management Model

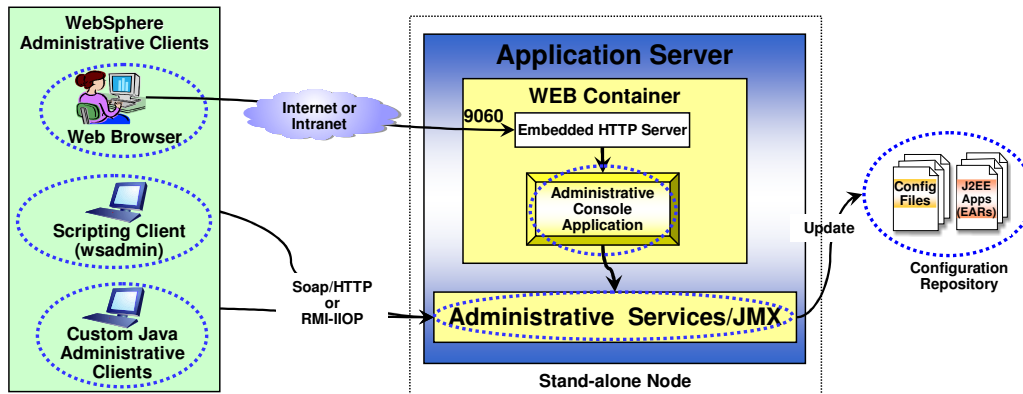
- This topology supports a single Application Server
 - ▶ If you need additional Application Servers, you can create new WebSphere profiles of type “Stand-alone Application Server”
- Application Server has local access to the configuration and application information needed to start the server and run applications
 - ▶ Configuration files are XML files
 - ▶ Application files are J2EE EAR files for applications installed on the server
- System Administrator uses the administrative clients to manage/operate the server



A stand-alone Application Server means only one application server exists in the topology. If other independently managed application servers are needed, they can be created by creating a new profile of the Stand-alone application server profile type. Profiles are covered in another module.

The application server configuration and J2EE application files is stored in XML files in the local file system. The System Administrator uses the administrative clients to interact with the server for any configuration or operational changes. Next foil discusses more on the various types of administrative clients supported for this environment.

Stand-alone Server : System Administration



- Multiple administration tools
 - ▶ Browser Based Administrative Console
 - ▶ Administrative (wsadmin) scripting
 - ▶ Custom Java programming interface for developing administrative programs
- Two type of commands
 - ▶ Configuration change commands – change the Server, application configuration, or both
 - ▶ Operation commands – Start to stop applications, services – no change to the configuration

The Administrative Service pictured here is a wrapper which enables remote communication to a JMX server, if needed. Administrative Clients provide the connection to the Administrative Service.

Besides exposing all the JMX APIs for use with external tools, V6 provides the following administrative clients to manage the environment, from configuration and operation perspective.

(1) Administrative Console: This is a browser based interface to manage the system hat provides many features to guide you through deployment and systems administration tasks. Use it to explore available management options.

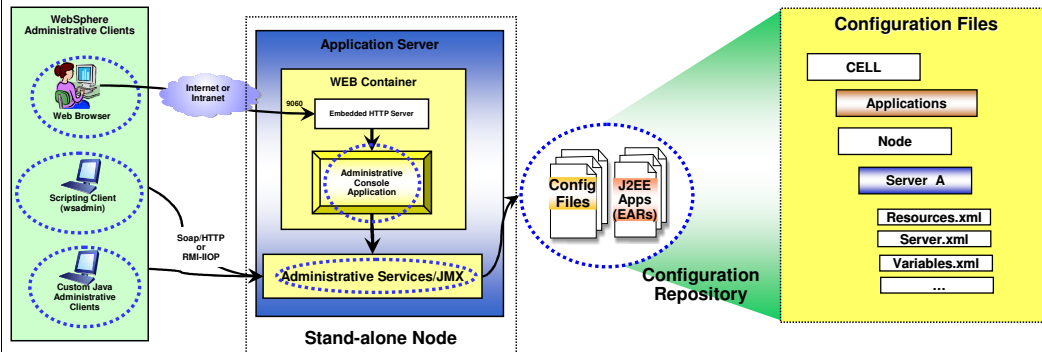
(2) Scripting The WebSphere administrative (wsadmin) scripting program is a powerful, non-graphical command interpreter environment enabling you to execute administrative operations in a scripting language. You can also submit scripting language programs for execution. The wsadmin tool is intended for production environments and unattended operations.

(3) Several Command line tools: These are commands that you run from an operating system command line prompt to perform specific tasks, as opposed to general purpose administration. Using the tools, you can start and stop application servers, check server status, add or remove nodes, and complete similar tasks.

(4) Custom Java Program: The user writes custom Java programs that use the Administrative APIs, based on JMX APIs.

The administration clients connect to the Application Server using SOAP/HTTP or RMI-IIOP, based on the type of the client.

Stand-alone Server: Configuration Repository



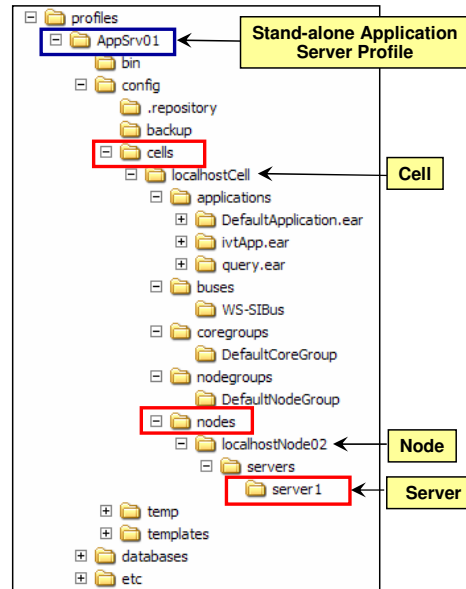
- Configuration repository is a set of XML documents in cascading directories
 - Contained in each <PROFILE_HOME>/config directory
- Contains server configuration and installed J2EE application files

V6 configuration repository are a set of XML files that represent the configuration of the environment (server, resources, and so on) and contain the J2EE application that run in the Application server.

Any configuration changes done through the Administrative client result in updates to the configuration repository XML files.

Configuration Repository Structure

- Repository has a 3-tier directory structure for Stand-alone Application Server, Managed Node or Cell
 - Cell
 - ▶ Contains cell-wide global definitions, resources and applications
 - ▶ All nodes and servers within the cell inherit this information
 - Nodes
 - ▶ Contains a directory for each node in the cell, that has node wide definitions and resources
 - ▶ All servers within the node inherit this information
 - Servers
 - ▶ Contains a directory for each Server in the Node, that has server definitions and resources
 - ▶ Pertains to the specific server only
- For a stand-alone Application Server, it does not matter in which tier the resources are defined
 - ▶ Most commonly, defined at the node or server level
 - ▶ Resources defined at cell level will be lost if you decide to federate the node to a cell later



The configuration repository uses the same structure for a stand-alone Application Server as it does for a full Network Deployment Cell. This makes federation process (adding a stand-alone application server to a Cell) relatively smooth under the covers.

The repository has a 3 tier structure of Cell, Nodes and Servers, as shown on this page.

Resources may be defined at any of the three levels, but on a stand-alone node, it does not matter which level is used. Note that from a best practice point of view, it is advisable to define resources at the Node or the Server level. That will preserve the resources when you need to federate the server to a cell at a later time.

Web Server Definition in Stand-alone Server

- A Web server can be defined in the configuration of a Stand-alone Application Server topology
 - ▶ Only one in stand-alone server topology
- Allows generation of custom plug-in configuration files
- Web Server definitions are created using the script file that is generated during the Plug-in install
 - ▶ Stand-alone server does not support defining Web server via the Administrative Console



New in Version 6 is the ability to define a web server configuration within a node. This allows you to create custom plug in files for specific web servers, taking into account the characteristics of the Web Server. In case of Stand-alone Application server topology, the Web Server definition is done via the generated scripts. There are no option to create a new Web Server definition in the Administrative Console. Once it has been created, you can modify the Web Server definition in the Administrative Console.

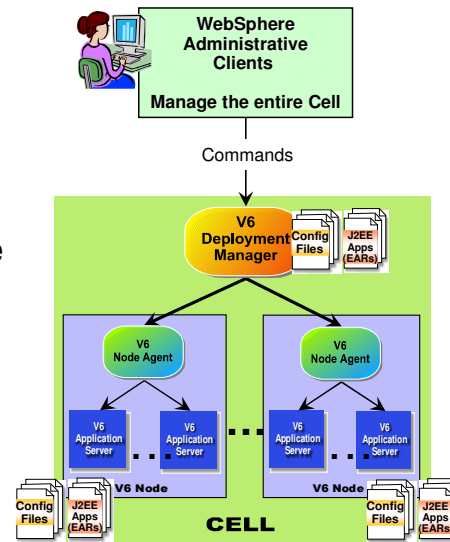
Section

System Management Architecture Network Deployment (ND) Cell

The next section describes the cell architecture.

ND Cell: System Management Model

- Each process in the cell has local access to its configuration and application information
 - Needed to start the process and run the installed applications
- Deployment Manager contains the master copy of configuration and application files and periodically, the files are synchronized to the nodes
- Administrator uses the administrative clients to manage/operate the entire cell



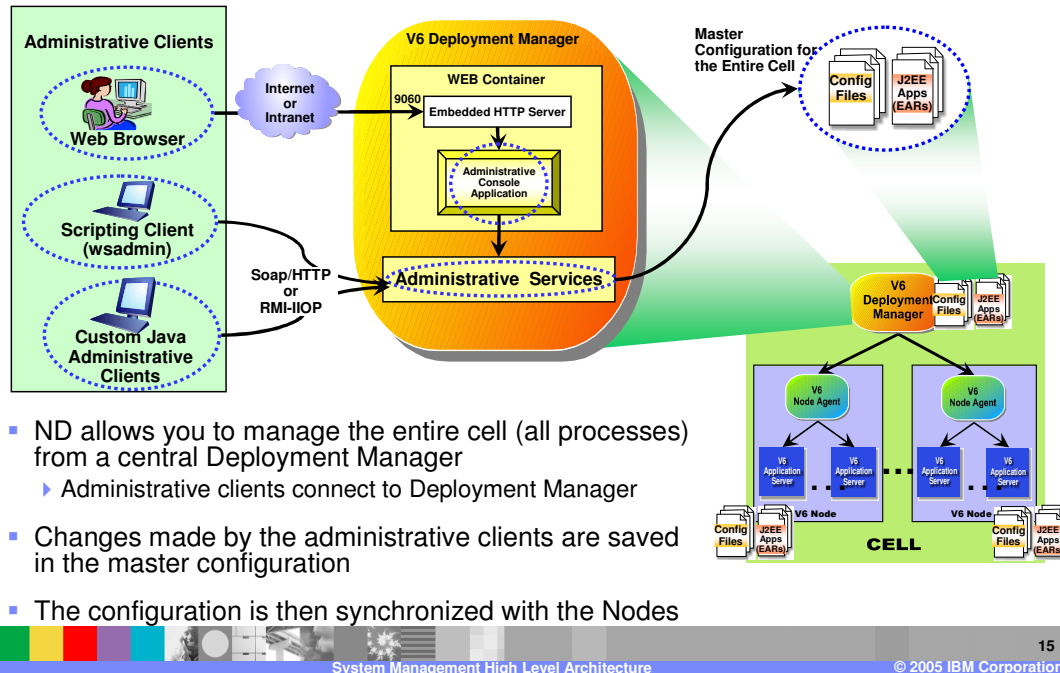
Network Deployment Cell System Management model has one Deployment Manager and zero or more Nodes. Each Node has one Node Agent and zero or more Application Servers, as shown within the Cell.

The Deployment Manager contains the master copy of the configuration repository.

Each node has the local copy of the configuration repository needed for all the processes on that node (the node agent and all the application servers). The processes on a node do not need to get the files from the Deployment manager, when they start.

When configuration files are modified (using the administrative clients) in the master repository, the changes are synchronized to the nodes, periodically, under System Administration control. The synchronization process is described in details later.

ND Cell: System Management



In a ND Cell environment, the System Administrator has the ability to manage the entire cell and all the processes in the cell. The Deployment manager (DMgr) and the nodes could be physically on separate machines or different distributed Operating systems.

The administrative clients connect to the DMgr, and from the DMgr, they can manage the entire cell. Configuration changes are saved in the master repository and then, those changes are synchronized to the nodes, under Administrator control.

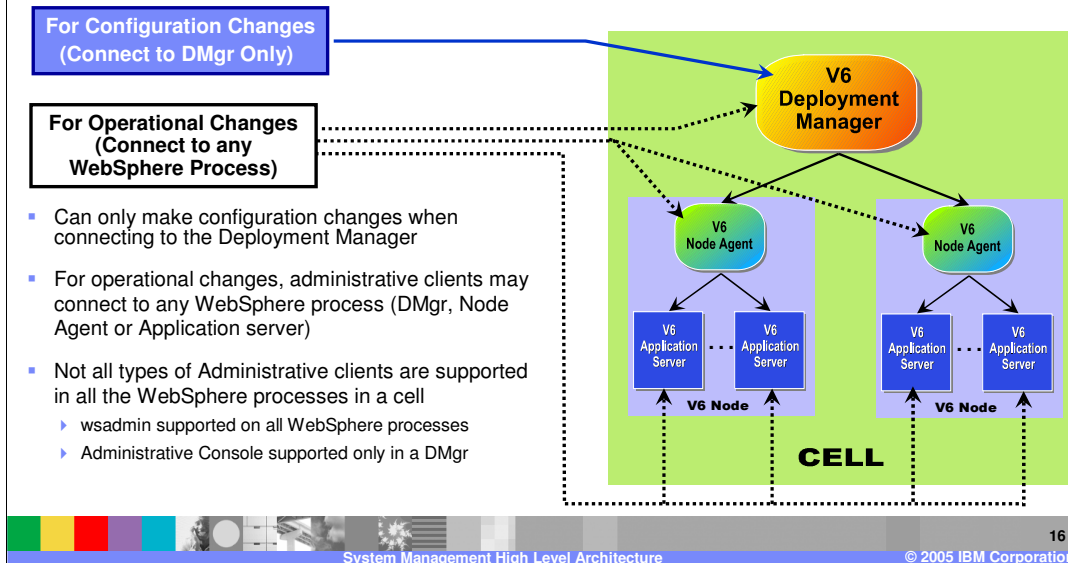
The administrative services run within the DMgr through which the master configuration changes are made.

The picture shows the Administrative clients connecting to the DMgr. The Administration console runs within the DMgr. The clients connect to the DMgr using the default port of 9060. The browser based client uses the Administration Console. The wsadmin and Custom Java programs use the JMX APIs.

It is important to understand the relationship between the files in the master repository and those out at the nodes. Using wsadmin or the Administrative Console or your favorite text editor, you can update files at the node level. However, those changes will only be temporary and will be overridden the next time there is a file synchronization between the deployment manager and node agent for that node. So generally, this is not something you would want to do, but there are times when this might make sense. More on this in later slides.

ND Cell: Administration Points

Shows the different WebSphere processes where the Administrative clients can be attached in a cell, for configuration and operational changes



JMX server runs within every V6 process. Hence, administrative clients can attach to any process.

However, since the Deployment Manager contains the master configuration repository, configuration changes are only permitted when connecting to Deployment Manager.

Operational changes can be done by connecting to any process.

When connecting to Deployment Manager, operational changes can be made to the entire cell.

When connecting to Node Agent in a cell, operational changes can be made to that node only.

When connecting to an Application Server in a cell, operational changes can be made to that server only.

In a ND environment, the Administration console is only available in the Deployment Manager.

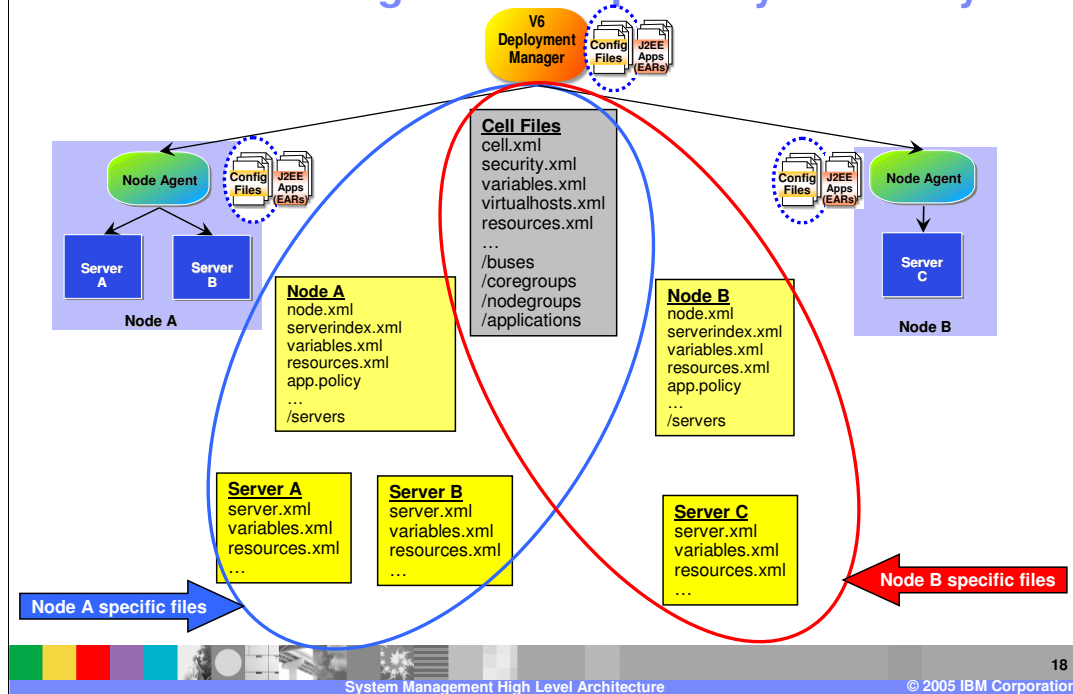
Command line wsadmin is available on all WebSphere processes.

Configuration Repository Overview

- Configuration repository contains the configuration files (XML files) and J2EE application files for all processes within a Cell
- Configuration files are stored in the profile/config/ directory
 - ▶ Default: <WAS_INSTALL>/profiles/<profile_name>/config/
- The master configuration repository is in the Deployment Manager's /config directory

The master configuration repository resides with the Deployment Manager. Nodes work from local copies stored in the individual profiles /config directories.

ND Cell: V6 Configuration Repository Hierarchy



The repository consists of several XML files, and is hierarchical in nature. The cell spans multiple nodes, and the nodes contain multiple servers. Each server has associated files with the same names, but they are stored in separate directories to avoid conflict. Similarly, Nodes each have their own files, stored in distinct directories. But the master copy of each of these files is stored at the cell level.

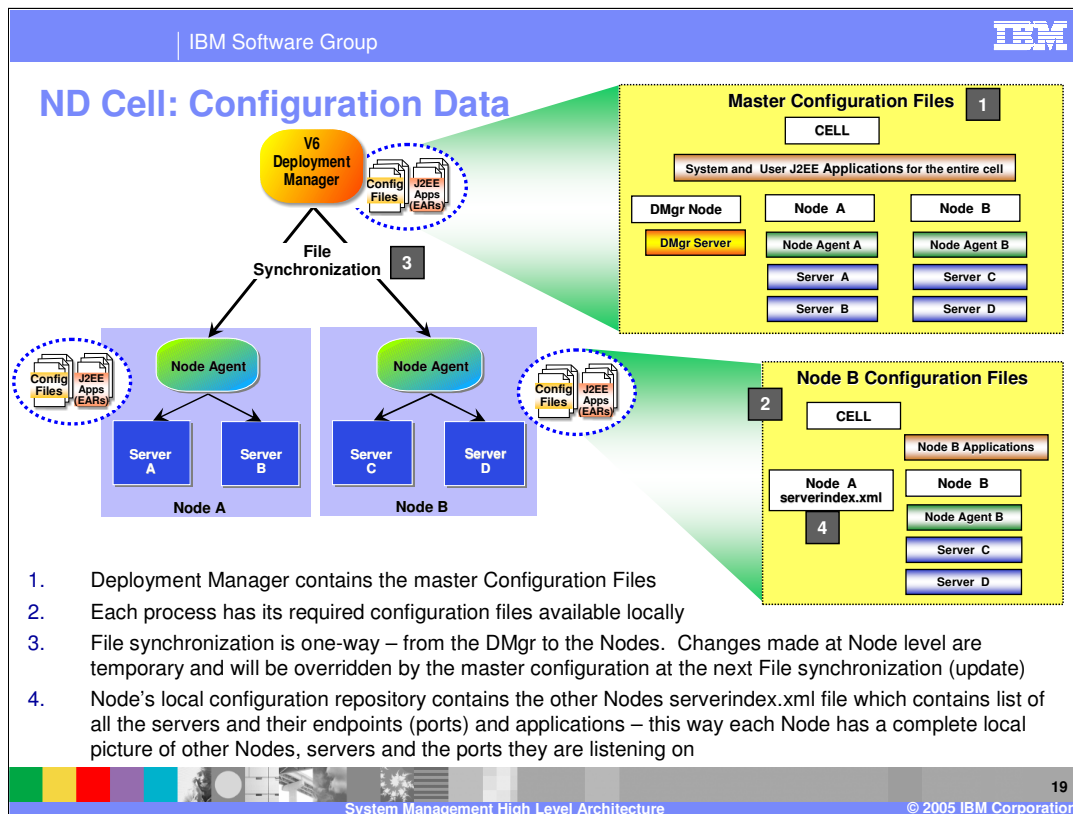
This picture shows the configuration repository of a cell with 2 nodes, node A and B. Node A has 2 servers, server A and B. Node B has 1 server, server C.

The master configuration contains all the files, whereas, each node will contain common cell files and files for that node only. One exception is that a node contains the serverindex.xml file for all the other nodes. The node serverindex.xml files contains the host and port end points of all the processes in that node.

So, in this picture, Node A will only contain the serverindex.xml file for Node B.

Similarly, Node B will only contain the serverindex.xml file for Node A.

This allows processes on Node A to know how and where to contact processes on Node B, and vice-versa. Hence, each process knows the entire cell topology and all the endpoints for all the processes within a cell.



In this slide, notice the configuration and application binary data in the Network Deployment environment. Each server has access to its own configuration files and application binary files needed for the server to start. The configuration data is in XML files, and the application data in J2EE EARs. The servers read these files in during server startup which allows the servers to initialize themselves with the appropriate configuration.

The deployment manager maintains a master configuration repository at the cell level, as shown by (1). This master repository contains all of the configuration files and EAR files for the entire cell. If you look at Node B, you will see that it only contains the subset of XML files from the master repository that are needed for the servers running on that node, having the Server C and Server D configuration files, but not having the Node A, Server A and Server B configuration files. Although not specifically illustrated, that is also true of the EAR files, with Node B only having the EAR files needed for Servers C and D.

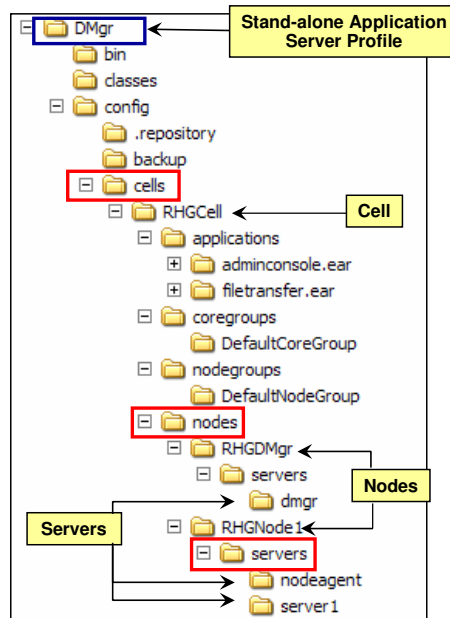
Additionally, node B will have the serverindex.xml file for Node A, as shown by (4). This lets all the processes in node B, know the end points of processes in node A.

Though not shown in this picture, similar set of files exist on Node A.

Any changes to the master configuration is synchronized to the nodes, as shown in (3).

Configuration Repository: Structure

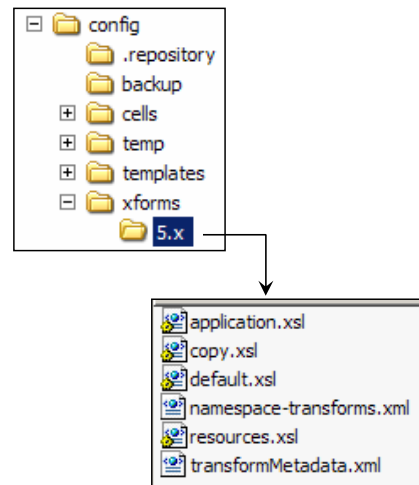
- Repository has a 3-tier directory structure for Cell, Managed Node or Stand-alone node
- Cell
 - Contains cell wide global definitions, resources and applications
 - All nodes and servers within the cell inherit this information
- Nodes
 - Contains a directory for each node in the cell, that has node wide definitions and resources
 - All servers within the node inherit this information
- Servers
 - Contains a directory for each Server in the Node, that has server definitions and resources
 - Relates to the specific server only



The Repository uses the same structure for a stand-alone node as it does for a full cell. This structure is more intuitively obvious for the Deployment Manager configuration. Illustrated here is the directory structure of the master repository.

ND Cell: Support of Mixed V5.x and V6 Nodes

- The master repository is always at the most recent version
 - ▶ Filtered to level appropriate for each node
 - ▶ Transformation happens during synchronization
- V6 -> V5 conversions implemented with XSLT
 - ▶ ...config/xforms/5.x directory



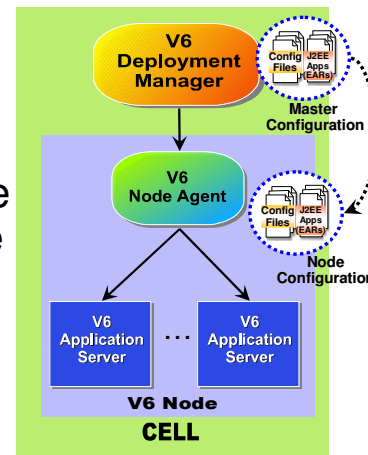
When migrating from V5, the environment may contain both V5 and V6 nodes.

In a V6 Cell that has mixed V5 and V6 nodes, the master configuration is always saved with V6 formatted data.

During the process of synchronization, the data is transformed so that V5 nodes get repository data in the V5 format, and V6 nodes see V6 formatted data. The transformation occurs using the XSLT supplied by the product.

ND Cell: File Synchronization

- Configuration updates made at the DMgr are synchronized with the node via the Node Agent
- DMgr and Node Agent use the file transfer application to transfer the file over HTTP
 - ▶ Updates are pulled by the Node Agent during file synchronization
 - ▶ Only one-way synchronization
 - Changes saved at the DMgr level are propagated down
 - ▶ Uses HTTPS if security is turned on

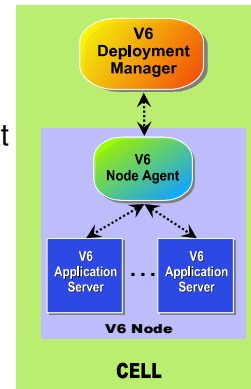


Repository synchronization uses the file transfer application, and is initiated by the node agent. Updates are never pushed from the node level up to the cell level.

The Filetransfer system application transfers the configuration files using the HTTP protocol. When you enable security in the WebSphere Application Server product, the file transfer service uses certificate-based mutual authentication. You can use the default key files in a test environment. Ensure that you change the default key file to secure your system

ND Cell: Distributed Process Discovery

- Deployment Manager, Node Agents & Application Server can start in any order
 - ▶ Exception: Node Agent must start before any application server on that node
 - ▶ Node Agent hosts the Location Service Daemon that is needed by the application server and application clients
- Communication channels established during process discovery
- Each node has its own copy of the configuration and application data



When servers start, they each have access to their own configuration data. Other servers that they normally communicate with may or may not be running, so which ones are running needs to be determined. They use a discovery protocol to attempt to establish communication with each other. Specifically, the deployment manager and the node agents need to establish communication and the node agents and managed servers on their node need to establish communication.

In general, servers do not have dependencies on other servers and therefore the order in which they are started is not important. However, on the distributed platform, there is one exception.

A node agent must be running in order for a managed server to run. This is because the Location Service Daemon runs in the Node Agent process, and a managed server cannot be started unless it is able to register itself with the location service daemon. One positive side effect of this limitation is that it guarantees that the node agent has a chance to synchronize before starting application servers and therefore the application servers are started at the current configuration level.

In some non-common cases, it is possible in version 5.1 and version 6.0 that the application server can be started before node agent is running. Refer to the Information Center for more details.

Section

Web Server Definitions

The next section will compare V5 and V6 administrative tasks.

Web Server Definitions

- Web servers can now be defined in your topology
- Allows association of the application to one or more defined Web servers
 - ▶ This allows generation of custom Plug-in configuration files for specific Web servers, allowing Web server to target specific applications running in the environment
- Web servers can be a Managed or Unmanaged Node

New in Version 6 is the ability to define a web server configuration within a node. In the case of the IBM HTTP Server, the WebSphere administrative console can stop and start the web server, and transparently update the plug-in configuration file on a remote machine where the web server is installed. Also, WebSphere can selectively target applications to specific web servers, so that only specific web servers can route requests to the application.

Section

Comparison of V5 and V6 System Management Tasks and Topics

The next section will compare Version 5 and Version 6 administrative tasks.

V5 to V6 Comparison: Stand-alone Server

Task or Topic for Standalone Application Server	V5 → V6
Topology	New WebSphere Profiles for Standalone Node, Managed Node and DMgr, sharing the same product install
System Management model	Same administration model using JMX MBeans
Configuration Repository	Similar Repository structure with some new V6 files and directories
Administrative Clients	Similar Administrative Clients with some new enhancements in Browser Administrative Console and new wsadmin AdminTask
Start, Stop WebSphere processes	Similar to V5, except "-profileName" option to start a server for a specific profile
Monitor WebSphere process	Similar to V5
Admin Security	Similar to V5
Commands	Most of the commands similar to V5, with "-profile" option added to the command New V6 commands: wasProfile (replaces wsinstance)

Most of this chart illustrates the similarity of V5 and V6.

Some of the differences is listed below:

V6 runtime environments are created using WebSphere profiles. In a V6 Network Deployment package, from the same install, the Deployment Manager and Stand-alone Application server environment are created. In V5, these are from different install images.

Many V6 commands have "-profileName" option to run the command for specific profile environment. If not specified, default profile is used.

V5 to V6 Comparison: Stand-alone Server (cont.)

Task or Topic for Standalone Application Server	V5 → V6
Application Administration	New in V6: <ul style="list-style-type: none"> ▪ Fine Grain Update new in V6 ▪ System Applications (example: adminconsole, filetransfer) ▪ Enhanced Application Ripple update for Cluster
Resources Administration	New in V6: <ul style="list-style-type: none"> ▪ Management of Service Integration Technology resources – Service Integration Bus resources and JMS resources using the bus Similar to V5: <ul style="list-style-type: none"> ▪ JDBC, Java Mail, J2C, Etc.

This chart highlights some more differences between V5 and V6.

New in V6 is the fine grain application update, where delta artifacts of an application can be modified, added, deleted without affecting the other parts of the application.

Also new in V6 is the notion of system applications that come bundled with the product and cannot be externally managed by the Administrator using the console. This allows system applications to be updated using with product update, if needed. More important is that the System Applications cannot be unknowingly deleted or stopped, having unintentional consequences.

Another new V6 feature in Application administration is the enhanced Application Ripple update for an application in a cluster. This allows the application update to occur one cluster member at a time, while they continue to run on other cluster members. This provides availability of the application even during updates.

From a resource view point, new in V6 is the Service Integration Technology resources to handle JMS messaging and provide the infrastructure for Service Oriented Applications (SOA) and Enterprise Service Bus.

V5 to V6 Comparison: Network Deployment Cell

Task or Topic for Network Deployment Cell	V5 → V6
Building Cell – Add/remove Nodes	Similar to V5
Configuration Repository File Synchronization from DMgr to the Managed Node	Similar to V5
Support for Mixed V5.x and V6 Nodes in a V6 Cell	New in V6
Support for Web Server definitions in the Cell topology	New in V6
Node Groups	New in V6
Core Groups	New in V6

The cell environment is similar to V5.

New in V6 is the support for mixed V5 and V6 nodes in a V6 cell, support for Web Server definitions, new architectural groupings called Node Groups and Core Groups.

Summary

- Discussed the following System Management architecture
 - ▶ System Management Models
 - ▶ Configuration Repository
 - ▶ Web Server definitions
 - ▶ Network Deployment specific topics on Mixed Nodes support, File Synchronization
- Provided comparison of V5 and V6 System Management Tasks



In summary, this presentation has illustrated some of the System Management architecture of the IBM WebSphere Application Server V6.

Trademarks, Copyrights, and Disclaimers

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

IBM	CICS	IMS	MQSeries	Tivoli
IBM (logo)	Cloudscape	Informix	OS/390	WebSphere
eIogo business	DB2	Series	OS/400	xSeries
AIX	DB2 Universal Database	Lotus	pSeries	zSeries

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

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

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

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

Linux is a registered trademark of Linus Torvalds.

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

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprocessing in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2004. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.

