



IBM Software Group

IBM WebSphere Application Server V6.1

Migrating to V6.1



@business on demand.

© 2006 IBM Corporation
Converted to video May 15, 2015

This presentation will focus on migrating to WebSphere Application Server V6.1

Agenda

- Migration overview
- Stand-alone application server migration
- Network Deployment cell migration
- Migrating cell Web server plug-in module
- Script migration
- Mixed-node environment limitations
- Summary and references
- Appendix



This presentation will address the following topics: overview of migration, migration of the stand-alone Application Server, cell migration, Web server plug-in, script migration and mixed node environment limitations of WebSphere Application Server V6.1. This presentation will also discuss what is new in V6.1 migration.

Migration overview

- Network Deployment cell supports mixed version (V5.0, V5.1 and V6) nodes
- Migration step is optional and is the last step
 - ▶ Install V6.1
 - ▶ Create V6.1 profile
 - ▶ Run migration utilities
- Migration does not delete previous levels
 - ▶ V5 /V6.0 deployment manager and federated nodes are deactivated on migrating to V6.1 – can be reactivated using scripts
- V6.1 supports Java™ 2 Enterprise Edition (J2EE) 1.4 and earlier releases

The definition of migration is to upgrade a WebSphere Application Server installation to a newer major release. WebSphere Application Server Version 6.1 migration leverages the existing environment and applications and changes them to be compatible with the WebSphere Application Server Version 6.1 environment. Existing application components and configuration settings are applied to the Version 6.1 environment during the migration process.

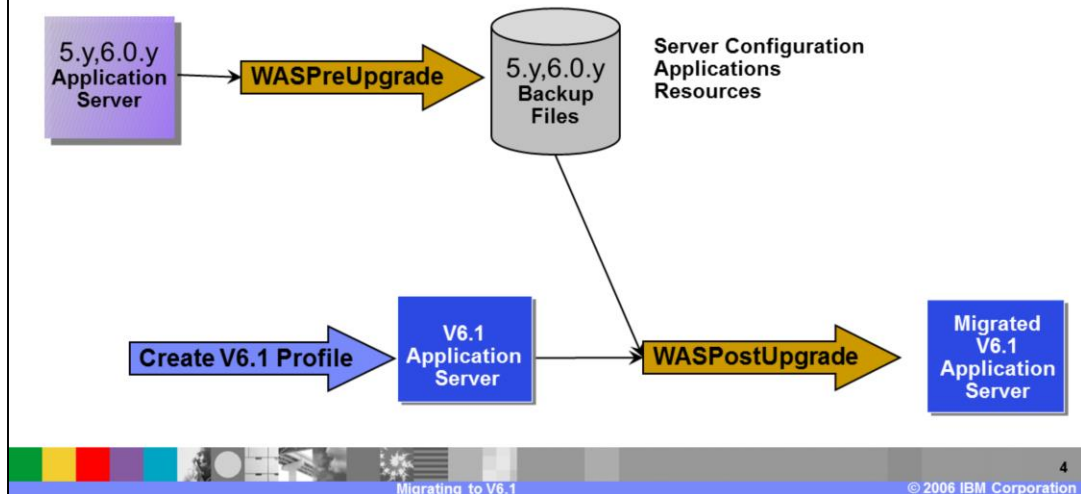
In order to minimize application downtime during migration, WebSphere Application Server V6.1 supports mixed version cells. This means that a cell can support both WebSphere Application Server V5 nodes and V6.1 nodes that operate in a fully functional cell.

Migration involves installing V6.1 and configuring at least one profile. Migration can be done by either running migration utilities, or installing and configuring the application and resources manually.

The previous version will still exist after running migration utilities. However, a V5 deployment manager will be disabled from operation after a migration using migration utilities. The V5 deployment manager can be re-enabled with a provided script.

Migration overview (cont.)

- Run **WASPreUpgrade** command on the 5.Y and 6.0.Y level that needs migration
- Run **WASPostUpgrade** on new V6.1 installation, pointing to the output produced by WASPreUpgrade – result is a migrated V6.1 environment



The migration utilities are a pair of commands.

The **WASPreUpgrade** command is used to save an existing installed WebSphere version's configuration and applications into a migration-specific backup directory.

The **WASPostUpgrade** command is used to process the migration-specific backup directory that was created by WASPreUpgrade and import that configuration and enterprise applications into the current environment.

Note that, you have to create at least one profile in order to perform a migration.

Supported migration scenarios for V6.1

Source product scenario	v6.1.y Express ³	v6.1.y Base ³	v6.1.y ND Standalone and Custom Profiles	v6.1.y ND DMgr Profile	v6.1.y Client ^{3,5}	v6.1.y Embeddable
v5.x.y, v6.0.y Base Standalone	support	support	support			
v5.x.y, v6.0.y Base Federated		support	support			
v5.x.y, v6.0.y ND Standalone	support	support	support			
v5.x.y, v6.0.y ND Federated		support	support			
v5.x.y, v6.0.y ND DMgr				support		
v5.x.y, v6.0.y Client ^{3,5}					support	
v5.x.y, v6.0.y Express ³	support	support	support			
v5.x.y, v6.0.y Bobcat ^{3,4}						support
v5.0.y EEonBase Standalone ^{3,4}	support	support	support			
v5.0.y EEonBase Federated ^{3,4}		support	support			
v5.0.y EEonND Standalone ^{3,4}	support	support	support			
v5.0.y EEonNDFederated ^{3,4}		support	support			
v5.0.y EEonND DMgr ^{3,4}				support		

The following combinations of WebSphere versions and offerings are directly supported for 6.1, that is, those scenarios where an Operating System (OS) upgrade or change is not required. Support is provided for all supported operating systems.

This component also provides some tools for upgrading J2EE 1.2 client resources and upgrading the configuration to the latest supported level for those configuration objects that support script compatibility.

Supported scenarios table reflecting product packaging

Notes:

In this table *.y indicates any PTF and Service pack level

In this table V5.x.y indicates V5.0.y **and** V5.1.y

Scenarios are not supported on zSeries because the specified product is not supported

Scenarios are not supported on iSeries® because the specified product is not supported

Client image is not supported with the Migration Wizard on Distributed. The Migration Wizard is also not supported on iSeries® or zSeries.

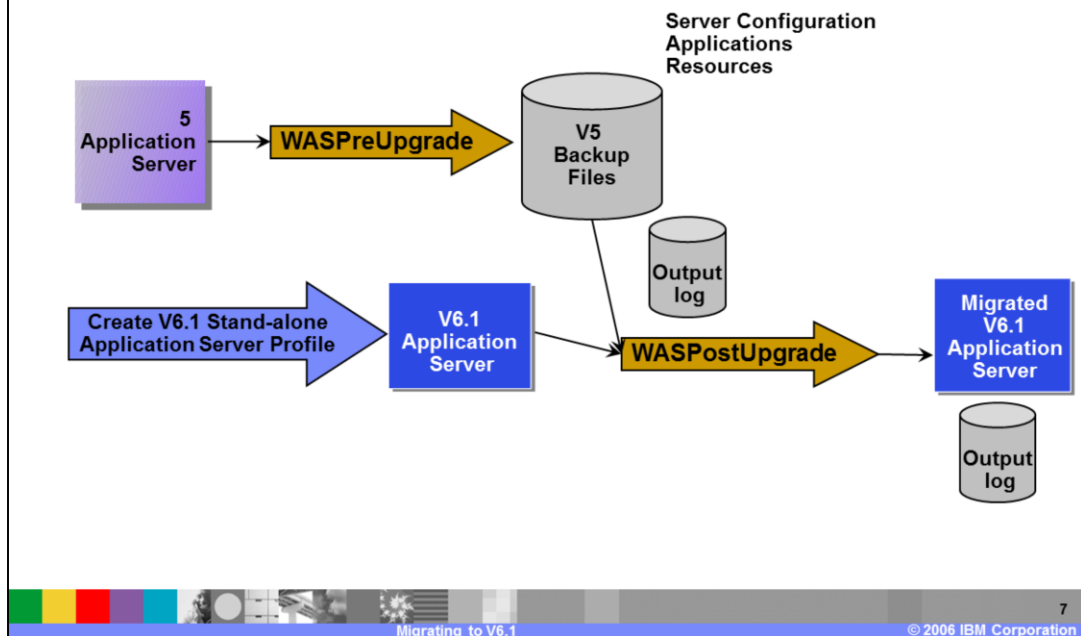
Section

Stand-alone application server migration



The next section discusses migration of a Stand-alone Application Server environment.

Migrating V5 stand-alone server



This example illustrates a more detailed picture of a migration sequence involving WebSphere Application Server V5 as the starting point of the migration.

1. Upgrade web server code and Plug-in
2. Create a V6.1 stand-alone application server profile using profile management tool or using *manageprofiles* command.
- ** Note that before WASPreUpgrade can run, the servers that are part of the source installation must be stopped. You should stop these manually. If there are servers running, WASPreUpgrade attempts to stop any active servers.
3. Run the **WASPreUpgrade** command which quickly copies your configuration to a backup copy
4. Run the **WASPostUpgrade** command which copies and transforms the configuration data from the backup into the new profile
5. Analyze the upgrade logs
6. Start the V6.1 Application Server and begin testing.

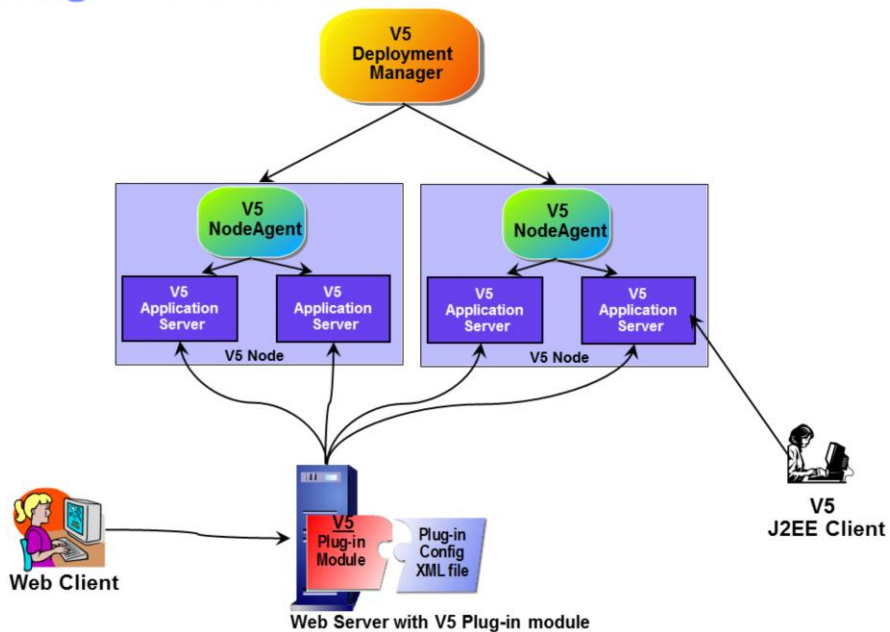
Section

Network Deployment cell migration



The next section discusses migration of Network Deployment Cells.

Starting with a V5 cell



Migrating to V6.1

© 2006 IBM Corporation

9

The next several pages will show you a step-by-step process of migrating a full WebSphere Application Server V5 cell to a full WebSphere Application Server V6.1 cell, with intermediate Mixed V5 and V6.1 Nodes.

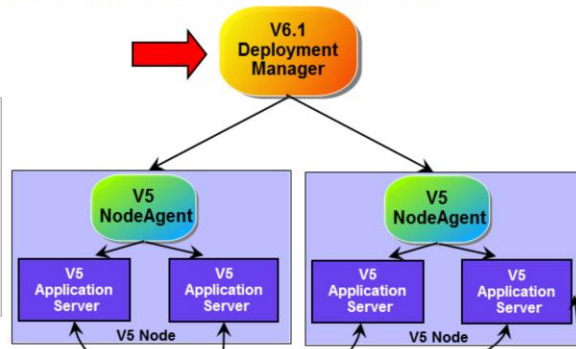
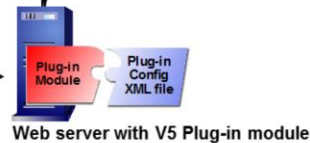
This scenario starts with a full V5 cell and V5 Web Server Plug-in as shown here.

Upgrading to a V6.1 cell environment

Upgrade Steps

1. Upgrade DMgr to V6.1
2. Upgrade Web Server Plug-in
3. Upgrade 1st Node
4. Upgrade 2nd Node
5. Upgrade Client

→ 1 Mixed node environment



¹ mixed node environment limitations listed later



The first step is to migrate the V5 Deployment Manager to a V6.1 Deployment Manager profile with the same Cell name. Now you are in a Mixed Node environment.

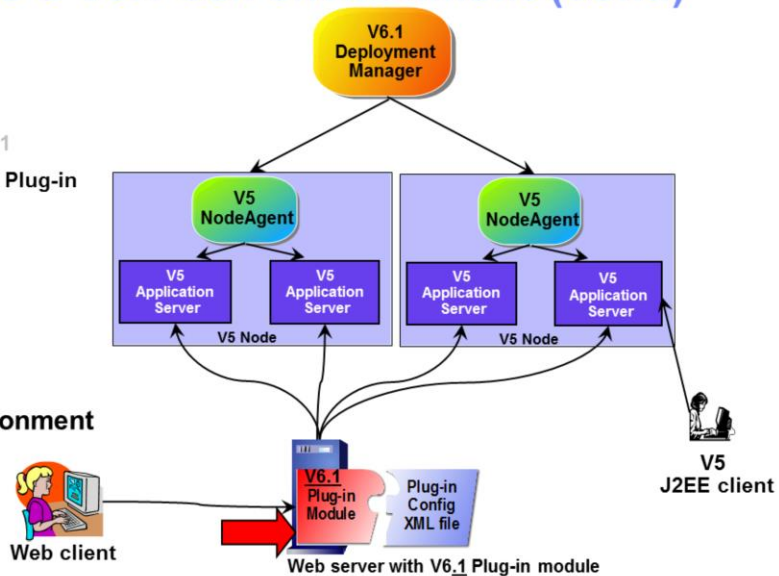
The result is that the V5 Deployment Manager is disabled. The V6.1 Deployment Manager now manages two V5 nodes. These nodes are fully operational.

Upgrading to a V6.1 cell environment (cont.)

Upgrade Steps

1. Upgrade DMgr to V6.1
2. Upgrade Web Server Plug-in
3. Upgrade 1st Node
4. Upgrade 2nd Node
5. Upgrade Client


→ mixed node environment



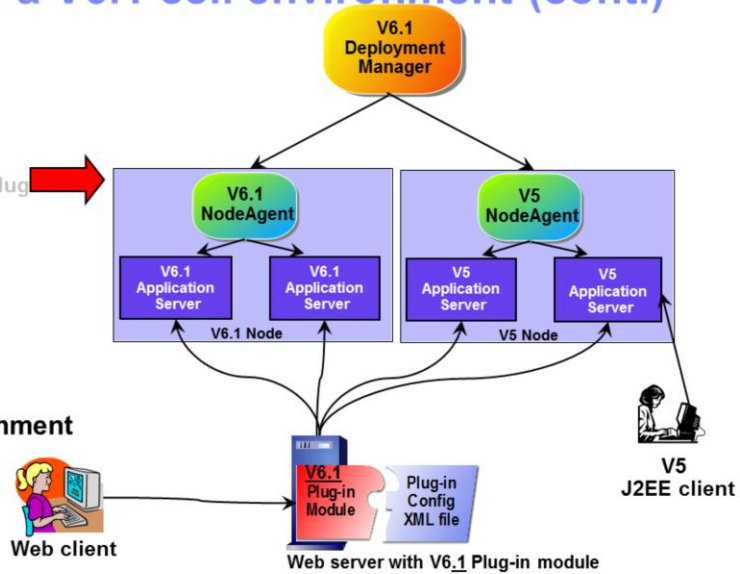
The next step is to upgrade the Web Server installation and WebSphere Plug-in with a manual installation. Note that the V6.1 Web Server plug-in is compatible with earlier versions in that it can send requests to V5 Application Servers and V6 Application Servers.

Upgrading to a V6.1 cell environment (cont.)

Upgrade Steps

1. Upgrade DMgr to V6.1
2. Upgrade Web Server Plug-in 
3. Upgrade 1st Node
4. Upgrade 2nd Node
5. Upgrade Client

→ mixed node environment



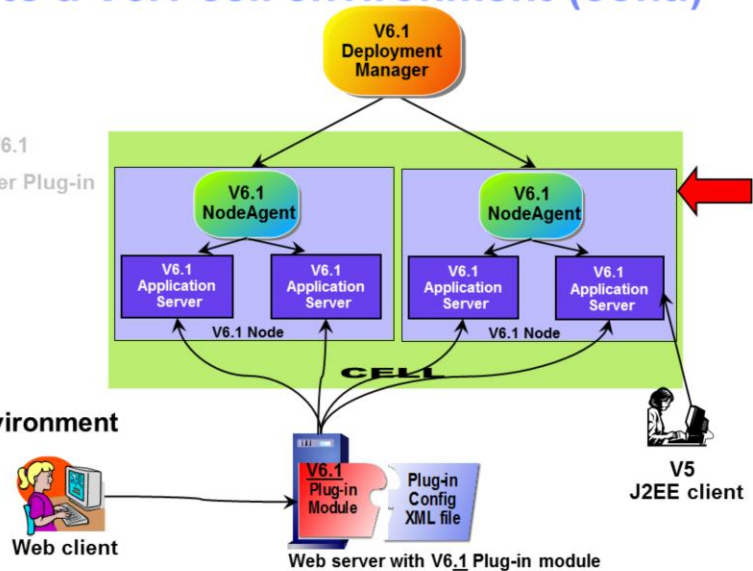
The next step is to migrate the first node using the migration utilities. Now the cell consists of a V5 node, and a V6.1 node, which are fully operational.

Upgrading to a V6.1 cell environment (cont.)

Upgrade Steps

1. Upgrade DMgr to V6.1
2. Upgrade Web Server Plug-in
3. Upgrade 1st Node
4. **Upgrade 2nd Node**
5. Upgrade Client

→ full V6.1 cell environment and V5 client



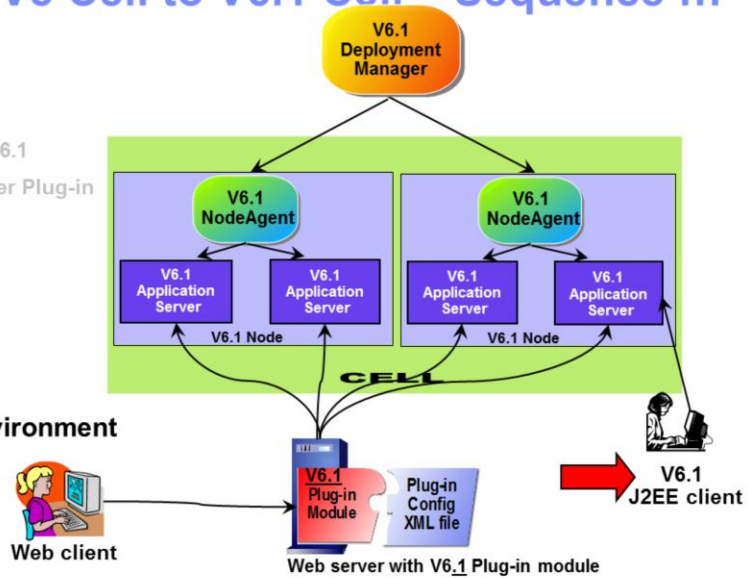
The next step is to migrate the second node. Both nodes are at V6.1 and are fully operational. Now the cell is a full V6.1 Cell. You still may have V5 Application Clients in the environment that are calling V6.1 Application Servers.

Upgrading V5 Cell to V6.1 Cell – Sequence ...

Upgrade Steps

1. Upgrade DMgr to V6.1
2. Upgrade Web Server Plug-in
3. Upgrade 1st Node
4. Upgrade 2nd Node
5. Upgrade Client

→ full V6.1 cell environment and V6.1 client



The last step is to migrate the J2EE client using the clientUpgrade command.

Migrating a V5 deployment manager

1. WASPreUpgrade copies configuration to backup copy
2. Create V6.1 profile for the DMgr
3. WASPostUpgrade copies and transforms configuration data from backup into new profile
4. Analyze migration log
5. Start V6.1 DMgr



To migrate a V5 Deployment manager, you should first stop it. The WASPreUpgrade will attempt to stop the Deployment Manager if it is running.

Next, create a V6.1 profile for the Deployment Manager. Note that you should make sure the cell name matches the cell name for the V5 cell.

Next, run the WASPostUpgrade command. This modifies the V6.1 Deployment Manager profile. Consult the output log for any errors. The log resides in the profile logs directory.

The V5 Deployment Manager is disabled from starting. If circumstances dictate that you must start the V5 Deployment Manager, you can re-enable the operation by running a wsadmin script migrationDisablementReversal.jacl from the previous deployment manager's bin directory.

Migrating V5 node in a cell

1. V6.1 DMgr must be running
2. Stop servers of node
3. Stop nodeagent
4. Run WASPreUpgrade copies configuration to backup directory
5. Create V6.1 Application Server profile with same node name as V5 node name
6. Run WASPostUpgrade - copies and transforms configuration data from backup into new profile
7. Analyze upgrade log
8. Start V6.1 nodeagent and test



To migrate a V5 node in a cell, the V6.1 Deployment Manager must be running, while the node agent and servers must be stopped.

Run WASPreUpgrade command which copies the configuration to the backup directory. Next, create an Application Server profile and ensure that the node name is the same as the V5 node name.

Next, run WASPostUpgrade, which first transforms the information in the backup directory into the V6.1 node profile. The V6.1 Deployment Manager is contacted and some of this information is transferred there. Analyze the output log in the node profile logs directory for any errors or warnings.

Start the nodeagent using the startNode command from the V6.1 node profile. The nodeagent will synchronize with the Deployment Manager and the node migration is complete. You can then start the application servers and test the applications.

Migration requirement: naming rules

- Migrating V6 deployment manager to V6.1
 - ▶ V6.1 deployment manager cell name must match V6 cell name
 - ▶ Though not required, it is recommended to have the same node names
- Migrating V6/V5 federated node to V6.1
 - ▶ V6.1 federated node name must match one of the V5 node names in the V5 repository, the same rules apply to V6.0 as V5 for federated
- Migrating V6 or V5 stand-alone application server to V6.1
 - ▶ No naming rule requirement for this migration



The only naming restrictions involve migrating nodes and deployment managers.

When migrating the deployment manager, the cell name in the V6.1 profile must match the cell name of the V5 deployment manager. When migrating a V6/V5 node in a cell, the node name in the V6.1 profile must match the node name of the V6/V5 node.

There are no naming restrictions when migrating stand-alone nodes. There are no restrictions on how you may assign port numbers.

What is migrated to the new profile?

- Applications
 - ▶ Copied unless turned off by includeApps = false option
- Server settings
 - ▶ Copied identically
- Resources
 - ▶ Non-JMS resources copied identically
 - ▶ V5 embedded JMS resource copied to V5 JMS compatibility resources
 - ▶ Generic and WebSphere MQ resources copied



- Applications are copied to the new profile unless `-includeApps=false` is specified on the `WASPostUpgrade` command. Standard sample applications that are installed as WebSphere samples are always excluded from copying.

- All server settings are copied.

- All resources are copied. Note that some resources are considered deprecated. V5 JMS resources are copied, but you must navigate to the V5 JMS resources, which are distinctly identified from V6.1 JMS resources.

Section

Migrating Web server plug-ins



The next section discusses migrating Web server plug-ins.

Web server plug-in migration sequence

- All Web Server Plug-in migration issues are manual
 - ▶ Not handled by migration utilities
- Must upgrade Web server plug-in before upgrading first node of V6.1 cell
 - ▶ V6.1 Plug-in modules can send requests to V5 and V6.1 nodes
- Regenerate plugin-cfg.xml after upgrade of DMgr and each node
 - ▶ For remote Web server, make sure it gets the new copy of the plugin-cfg.xml copy (manually or automatically)



If a node configuration relies on a Web server, that Web server must be upgraded to a supported level. That upgrade is a manual installation of the new Web server and installation of the V6.1 Web server Plug-in files. The upgrade must happen before any WebSphere Application Server V6.1 node is tested. The alternative is that the HTTP request will not be routed to the V6.1 application server.

After migrating the Web server and Plug-in files, the Plug-in configuration file , plugin-cfg.xml, must be regenerated and copied after each node migration.

Section

Script migration



The next section discusses script migration.

Script compatibility

- There are some changes that support scripting when migrating from WebSphere Application Server V5 to V6
- The following changes can be made with script compatibility support:
 - ▶ HTTP transports: When compatibility mode is chosen, the old HTTPTransport objects are migrated and mapped onto the channel architecture. Existing scripts can modify these objects and will run unchanged.
 - ▶ Process definition: The name of this object is changed from processDef to processDefs. You can mitigate this change by using the compatibility mode mapping provided by the migration tools.



There are a few changes to be aware of that are required for your existing scripts when moving to WebSphere Application Server Version 6. These changes are assisted by the compatibility mode provided by WASPostUpgrade command. During migration, the default is to migrate using compatibility mode. If this option is taken, then the old object types are migrated into the new configuration; all existing scripts will run unchanged.

The following changes can be made with script compatibility support.

HTTP transports: the new architecture for V6 uses the new channel framework. HTTP definitions are mapped on top of this support. When compatibility mode is chosen, the old HTTPTransport objects are migrated and mapped onto the channel architecture. Existing scripts can modify these objects and will run unchanged.

Process definition: The name of this object is changed from processDef to processDefs. You can mitigate this change by using the compatibility mode mapping provided by the migration tools. The change to scripts to use the new location is as follows:

Old example:

```
Using Jacl:
set processDef [$AdminConfig list JavaProcessDef $server1]
set processDef [$AdminConfig showAttribute $server1 processDefinition]
Using Jython:
processDef = AdminConfig.list('JavaProcessDef', server1)
print processDef
```

New example. Identify the process definition belonging to this server and assign it to the processDefs variable:

```
Using Jacl:
set processDefs [$AdminConfig list JavaProcessDef $server1]
set processDefs [$AdminConfig showAttribute $server1 processDefinitions]
Using Jython:
processDefs = AdminConfig.list('JavaProcessDef', server1)
print processDefs
```

Section

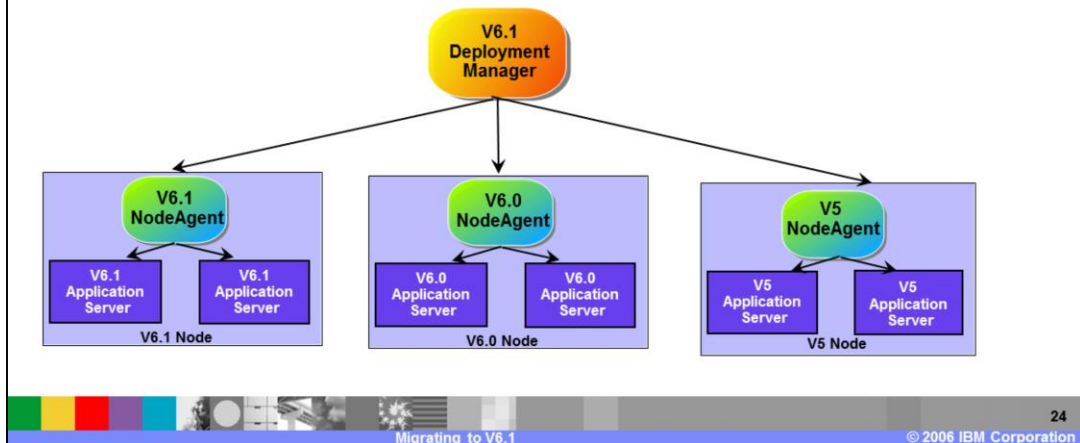
Mixed version cells



This section discusses mixed version cell environment limitations.

Mixed version environment

- A mixed version cell is a topology where different nodes from different releases exist together.
- V6.1 Enables customers to add a new V5 or V6.0 Application Server to a mixed (v5 and V6) release cell



A mixed version cell is a topology where different nodes from different releases exist together.

V6.1 Enables customers to add a new V5 or V6 Application Server to a mixed V5 and V6 release cell

Support will be provided for a Deployment Manager (Dmgr) profile that has mixed Releases of nodes in the cell (5.0, 5.1, 6.0 or 6.1).

V6.0.0 and V6.0.1 will not be supported in a Mixed node environment, so your V6.0 nodes should be at the V6.0.2 level. Migration support will still migrate these nodes during DMgr migration but will issue a warning message that the nodes cannot be managed by the V6.1 DMgr.

Mixed version environment (cont.)

- Some restrictions were removed in V6.1

	What is to be added to the Mixed Release Cell	Supported in v6	Supported in v6.02	Supported in v6.1
Adding New Nodes	V6.0.2 Node	Yes	Yes	Yes
	V5 Node	No	No	No
Adding New Servers	V6.0.2 Servers in V6.0.2 Node	Yes	Yes	Yes
	V5 Servers in V5 Node	No	Yes	Yes ¹
Adding New Cluster Members	V5 Server in V5 only Cluster	No	Yes	Yes ¹
	V5 Server in V6 only Cluster	No	No	No
	V6 Server in V5 only Cluster	No	No	No
	V5 Server in mixed (V5 and V6) Cluster	No	Yes	Yes ^{1,2}
	V6 Server in mixed (V5 and V6) Cluster	Yes	Yes	Yes ^{1,2}

- Minimum V5.0 level required
- Minimum V6.0 level is required



Some of the restrictions for a mixed release environment were relaxed in V6.0.2 and V6.1, and you can now add a new V5 cluster member to a pre-existing mixed cluster, or cluster containing only V5 members.

However, the ability to add V5 nodes into a V6.1 cell is still not supported. A node must be at version 6.0.2 or later to be federated into a V6.1 cell.

In Summary:

- V5 nodes can only be added to a V6 cell through migration.
- Creation of a V5 server in a V6-only cluster is not supported.
- Similarly, creation of a V6 server in a V5-only cluster is not supported.

JMX compatibility

- Version 6.1 supports JMX 1.2
- This version offers JMX interoperability between 6.1 and back level nodes at Version 5 or at Version 6.0.2 and later
 - ▶ Interoperability with Version 6.0.0 or 6.0.1 nodes is not supported
- Version 6.0 supports JMX 1.2
- Version 5 supports JMX 1.1
 - ▶ Interoperability between Version 6.1 or 6.0.2 nodes with 5 nodes is supported only with the SOAP connector



WebSphere Application Server version 6.1 supports JMX 1.2. It offers JMX interoperability between 6.1 and back level nodes at Version 5 or at Version 6.0.2 and later

Interoperability with Version 6.0.0 or 6.0.1 nodes is not supported

Version 6.0 supports JMX 1.2 and WebSphere Application Server version 5 supports JMX 1.1

Interoperability between Version 6.1 or 6.0.2 nodes with 5 nodes is supported only with the SOAP connector

Note: The JMX V1.2 specification is compatible with the earlier JMX V1.0 specification. However, you might need to migrate custom MBeans that are supplied by products other than the Application Server from Version 5 to Version 6. The primary concern for these custom MBeans is related to the values that are used in key properties of the JMX ObjectName class for the MBean. The open source mx4j implementation more stringently enforces property validation according to the JMX 1.2 specification. Test the custom MBeans that you deployed in Version 5 in Version 6, to ensure compatibility. Full details of the JMX V1.2 specification changes from the JMX V1.0 specification are available in the JMX 1.2 specification.

Section

Summary and references

The next section summarizes the presentation.

Summary

- Large degree of application compatibility
 - ▶ Application levels forward compatible
 - ▶ Many IBM programming model extensions (Enterprise) supported at lowest product level
- Large degree of upgrade flexibility
 - ▶ Migration tools provide automated migration of configuration and applications
 - ▶ WebSphere Application Server V5 and V6.1 coexist in Network Deployment cell



This presentation has shown you that the WebSphere Application Server V6.1 product offers a great deal of application compatibility so that most applications will migrate to V6.1 with very little effort. Applications can be migrated using automated migration utilities and thus allow rapid migration with a minimum of effort. The migration utilities allow a staged migration of Network Deployment cells such that nodes can be migrated one at a time as time permits allowing a more controlled migration with minimum downtime. Additionally mixed node and coexistence of V5/V6 are supported for 6.1.

This presentation has shown you how WebSphere Application Server Version 6.1 migration leverages the existing environment and applications and changes them to be compatible with the WebSphere Application Server Version 6.1 environment. Existing application components and configuration settings are applied to the Version 6.1 environment during the migration process.

References

- WebSphere Application Server V6.1 Information Center
- IBM Redbooks – www.redbooks.ibm.com
 - ▶ IBM WebSphere Application Server V5.0 System Management and Configuration SG24-6195
 - ▶ Migrating to WebSphere V5 SG24-6910
 - ▶ Check for new Redbook on migration



Additional references include the V6.1 Information Center and various IBM Redbooks covering WebSphere Application Server and migration topics.

References

- **DeveloperWorks articles**
 - ▶ **Migrating from V4 to V6.0**
 - http://www-106.ibm.com/developerworks/websphere/library/techarticles/0405_beaton1/0405_beaton1.html
 - ▶ **Migrating from Weblogic to WebSphere**
 - http://www-128.ibm.com/developerworks/websphere/library/techarticles/0405_beaton2/0405_beaton2.html
 - ▶ **Migrating Applications from WebLogic, JBoss and Tomcat to WebSphere V6**
 - <http://www.redbooks.ibm.com/abstracts/SG246690.html?Open>

Section

Appendix

Appendix

- Derby Migration
- Deprecated features
- Migration commands

Derby migration

- Migration tools will perform two kinds of migration:
 - ▶ WebSphere JDBC provider migration
 - ▶ Cloudscape database migration to Derby
- Cloudscape JDBC providers no longer exist in the WebSphere V6.1 release, and are replaced by Derby JDBC providers
- Changes that the migration tools will perform on Cloudscape providers:
 - ▶ classpath
 - ▶ implementationClassName
 - ▶ datasourceHelperClassname
 - ▶ databasename.



WebSphere V6.1 ships and supports Derby v10.1 which is the open source follow on release of Cloudscape V5.1.

Cloudscape migration from one release to the next used to be (prior to Derby) as simple as appending "upgrade=true" to the database (or add the "upgrade=true" as a connection attribute). This Cloudscape migration story has changed, however, and a manual/explicit migration is needed to move database instances from Cloudscape V5.1 to Derbyv10.1

WebSphere plans to ship a batch/shell file that users and migration tool can call to do the migration. The migration script will be called migratedb.bat/sh and will be located in WebSphere_HOME/derby/migration/bin. The migration script will take two parameters <new database name> and <old database name>.

Deprecated functions

- The `clientUpgrade` command is deprecated in this release. Its purpose is solely to upgrade J2EE 1.2 client resources in client jars to the latest client resource level. Since version 4.0 (the introduction of J2EE 1.2 support) is no longer supported this deprecation is made
- The **`wasprofile`** command is deprecated and has been replaced by the **`manageprofiles`** command.
- The following **`WASPostUpgrade`** command parameters are deprecated:
 - ▶ `-import xmi_data_file`
 - ▶ `-substitute "key1=value1[key2=value2;[...]]"`



If a function is deprecated, IBM intends to remove it in a later release of the product. Generally, IBM will not remove a function/feature until at least two major releases or three full years (whichever is longer) after the release in which it is deprecated. In rare cases, it might become necessary to remove functions/features sooner.

Migration commands

- WASPreUpgrade
- WASPostUpgrade
 - ▶ Examples
- migrationDisablementReversal.jacl script



The next screens show migration commands and some examples.

WASPreUpgrade command syntax

```
WASPreUpgrade <backupDirectoryName> <currentWebSphereDirectory>
[-traceString <trace_spec> [-traceFile <file_name>]] <options>
```

Required Parameters

-backupDirectoryName	output backup directory
-currentWebSphereDirectory	Current installation

Optional Parameters

-traceString <trace_spec> [-traceFile <file_name>]	Used to gather trace information
--	----------------------------------

Value of traceString is "*"=all=enabled" with quotation marks

Example: `WASPreUpgrade C:\WASV5_backup C:\Websphere\Appserver`



The first two parameters are required and positional; the others are optional. The example given would also be the same for V6.

```
WASPreUpgrade backupDirectoryName currentWebSphereDirectory
[-traceString <trace_spec> [-traceFile
<file_name>]]
```

The first two parameters are required and positional. The others are optional.

backupDirectoryName - The name of the directory in which to store the saved configuration and files. The directory is created if it does not already exist. This is a required parameter.

currentWebSphereDirectory - The name of the root directory for the current WebSphere Application Server instance. On Distributed this is the install Root, on iSeries it is the userRoot. This can be any of the [version-offering combinations](#) listed in the first column of the table. This is a required parameter.

[-traceString <trace_spec> [-traceFile <file_name>]] - These optional parameters are used to gather trace information for use by IBM service personnel. The value of the traceString parameter is "*"=all=enabled" and must be specified with quotation marks to be processed correctly.

Example:

```
WASPreUpgrade c:\WAS6Backup c:\WebSphere\WAS6.0
```

For more information, see the WebSphere V6.1 Information Center for more options and details regarding the WASPreUpgrade command.

WASPostUpgrade command syntax

WASPostUpgrade backupDirectoryName(-oldProfile<profile name>)[.....]

Required Parameters

backupDirectoryName	output of WASPreUpgrade
---------------------	-------------------------

Optional Parameters

-oldProfile <profile name>	A profile in V6.0 and an instance name in V5
-profileName <profile name>	V6.1 profile name
-portBlock <startingPortNumber>	starting seed to create new ports
-backupConfig <true false>	true/false decision to backup the new install config
-replacePorts <false true>	true/false decision to add or replace virtual host ports
-includeApps <true false>	true/false decision to migrate applications
-scriptsCompatibility <true false>	true/false decision to migrate definitions in configuration
-applInstallParameters - <"wsadmin parameters">	Parameters passed to wsadmin during installation of the application during migration
-traceString <trace_spec> [-traceFile <file_name>]	Used to gather trace information



WASPostUpgrade backupDirectoryName [-oldProfile <profile name>][.....]

Example:

```
WASPostUpgrade c:\WAS6Backup -oldProfile Stand-aloneProfile -profileName Stand-aloneProfile
```

For more information, see the WebSphere V6.1 Information Center for more options and detail regarding the WASPostUpgrade command.

WASPostUpgrade examples

- **Default migration**
 - ▶ WASPostUpgrade **C:/WASV6.0_backup**
 -profileName V6.1_Newnode

- **No copying applications**
 - ▶ WASPostUpgrade **C:/WASV5.0_backup**
 -profileName V6.1_Newnode
 -includeApps false



The first example shows the most simple case using the defaults. If you omit the `-profileName` option, the default profile is used.

The second example omits copying applications. This means just server configurations and resources are copied.

migrationDisablementReversal.jacl script

- If you chose to disable the previous deployment manager when you migrated to the WebSphere Application Server Version 6.1 deployment manager, use the **wsadmin** command to run this script from the Version 5 or 6.0 *profile_root/bin* directory of the deployment manager that you need to roll back from Version 6.1.



Use the following parameters:`app_server_root/bin/wsadmin -instance instance -conntype NONE -f profile_root/bin/migrationDisablementReversal.jacl`

To restore the Version 5.1 default deployment manager, for example, you might use the following command:`/QIBM/ProdData/WebAS51/ND/bin/wsadmin -instance default -conntype NONE -f /QIBM/UserData/WebAS51/ND/default/bin/migrationDisablementReversal.jacl`