

# Best Practices for Configuration Changes in WebSphere Application Server v6.x

**Author:**  
Leigh Williamson

**Co-Author:**  
**Owner:**  
Pravin D Patel  
Ajit Jariwala





## Table of Contents

1	Overview .....	3
1.1	Procedure to backup and restore configurations .....	3
1.2	For more information .....	3
2	Changing Host Name or IP address of WSAS install .....	4
2.1	Procedure to alter the hostName property of ND node (dmgr profile) .....	4
2.2	Procedure to alter the hostName property for a node profile federated in DMGR profile...	6
2.3	Procedure to alter hostname property for a standalone Base NODE profile .....	7
3	Changing the node name of WSAS node profile .....	9
3.1	Procedure to alter the node name property for a standalone Base profile .....	9
3.2	Procedure to alter the node name property for a federated Base node profile .....	10
4	Changing cell name of WSAS install .....	14
4.1	Procedure to alter the cell name property for a standalone Base profile .....	14
4.2	Procedure to alter the cell name property for federated Cell .....	15
5	Changing Server Name of WSAS install .....	17
6	Repository Migration of WSAS install .....	18
6.1	Procedure for Repository Migration of Standalone WSAS profile .....	18
6.2	Repository Migration of federated WSAS dmgr profile .....	19
7	WebSphere Configuration Archive.....	20



# 1 Overview

This white paper describes ways to manually modify certain parts of the WebSphere Application Server version 6.x configuration that are not available through the administrative tooling that comes with the product. Version 6.x configuration, for all editions of the product, is stored in XML files in a profile name subdirectory under the main product installation root directory. Administrative tooling shipped with WebSphere product provides support for modifying most of the configuration settings. This tooling includes the web-based admin console program, the wsadmin scripting tool, command line utilities, and even a public Java API for WebSphere administration.

While the version 6.x tooling supports altering most of the configuration, certain modifications that impact the structure of the configuration directories (the topology of the environment) were not available to be shipped with the version 6.x products. However, these changes can be accomplished without tooling through the manual procedures described in this document.

Each section of this document treats a specific configuration modification separately. Each section begins with a short discussion of the issues related to changing the specific part of the configuration. Then one or more procedures are provided to itemize the steps required to perform the modification manually.

## 1.1 Procedure to backup and restore configurations

**Always backup the existing configuration directories, using the backupConfig utility supplied with WebSphere, before performing any of the procedures described in this document.** If the configuration changes applied by following one of these procedures results in the system no longer working, you can use the restoreConfig utility to return the configuration to its state prior to making the changes.

```
<WAS_DMGR_PROFILE_HOME>/bin/backupConfig (.bat/.sh) -username <userid> -password  
<password>  
<WAS_NODE_PROFILE_HOME>/bin/backupConfig (.bat/.sh) -username <userid> -password  
<password>
```

**Note:** Verify that the jar file created by backupConfig is valid, either use restoreConfig or jar -v <backup\_file>, because we have seen client who have extremely large config directories not be able to run backup and restore config properly.

```
<WAS_DMGR_PROFILE_HOME>/bin/restoreConfig (.bat/.sh) -username <userid> -password  
<password>  
<WAS_NODE_PROFILE_HOME>/bin/restoreConfig (.bat/.sh) -username <userid> -password  
<password>
```

Also backup **setupCmdLine.bat** on Windows systems or **setupCmdLine.sh** on UNIX systems from **<WAS\_PROFILE\_HOME>/bin** or **<WAS\_PROFILE\_HOME>/bin** since backupConfig utility does not backup this file.

## 1.2 For more information

Refer to the WebSphere InfoCenter at following web link.

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>



## 2 Changing Host Name or IP address of WSAS install

**Note:** If the **IP address** is used **instead of Host Name** during WebSphere install, then use this section to alter the IP address; else **IP address of the system can be changed without any modification to WebSphere configuration repository.**

The WSAS 6.x configuration includes a `hostName` property in more than one configuration document. The value of the `hostName` property can be one of the following:

- Fully qualified DNS hostname string
- Short DNS hostname string (the suggested value selected by WSAS 6.x product install)
- Numeric IP address

There are advantages and disadvantages to any of the possible values. Customers will need to use a `hostName` property value that best suites their needs.

The fully qualified DNS hostname has the advantage of being totally unambiguous and also flexible. The actual IP address for the host system can be changed without requiring a WSAS config change. This value for `hostName` is useful if the IP address is anticipated to change frequently, as is the case when DHCP is used to assign IP addresses to host machines when they boot up. This value for `hostName` has the disadvantage of being dependent on DNS. If DNS is not available, then connectivity is compromised.

The short hostname is also dynamically resolvable, and has the advantage of being able to be redefined in the local hosts file so that the system can run WebSphere even when disconnected from the network. If the short hostname is resolved to 127.0.0.1 (local loopback) address in the hosts file for the system, then WSAS can be used when disconnected from the network. This value for `hostName` also has the disadvantage of being dependent on DNS in order to work correctly when connected.

A numeric IP address has the advantage of not depending on DNS in order to function. The disadvantage is that it is a fixed address and must be altered in the WSAS config files if the real system IP address is changed at any point.

The `hostName` property is given its value during product installation. Care should be taken during product install to select the `hostName` value that best suites the situation in the WSAS network environment. There is no feature in the browser-based admin console for WSAS to modify the `hostName` property associated with a particular node. A `wsadmin` script can be written to modify the `hostName` property if this is necessary after the product has been installed. The following procedure can be used to manually alter the `hostName` of a node.

### 2.1 Procedure to alter the `hostName` property of ND node (dmgr profile)

1. Always perform `backupConfig` before making significant configuration changes.
2. Ensure that Deployment Manager, `nodeagent`, `jmsserver` and all Application Server JVM processes have been stopped in the entire Cell. Also verify that no WebSphere java process or `WASService.exe` (on Windows) process is running.

To stop Deployment Manager, use following command from dmgr profiles's bin directory.  
`<WAS_DMGR_PROFILE_HOME>/bin/stopManager (.bat/.sh) -username <name> -password`



<password>

To stop nodeagent, use following command from node profiles's bin directory.

```
<WAS_NODE_PROFILE_HOME>/bin/stopNode (.bat/.sh) -username <name> -password  
<password>
```

To stop Application Server, use following command from node profiles's bin directory.

```
<WAS_NODE_PROFILE_HOME>/bin/stopServer (.bat/.sh) <server> -username <name> -  
password <password>
```

3. In the **serverindex.xml** file for the ND (dmgr profile) being modified, alter the value of the hostName property for the ServerIndex element in the file. Search for string "**hostName=**" to find this property.

Also in the serverindex.xml file for the ND (dmgr profile) being modified, alter the host property for every **EndPoint** in the file. Search all occurrences of "**host=**" string in following serverindex.xml

```
<WAS_DMGR_PROFILE_HOME>/config/cells/<cell>/nodes/<nodeManager>/serverindex.x  
ml
```

4. Edit the **wsadmin.properties** file in the properties subdirectory under the dmgr profile home. Change the value of the **com.ibm.ws.scripting.host** property in that file to the new host address.

```
<WAS_DMGR_PROFILE_HOME>/properties/wsadmin.properties
```

Repeat above step for every Node profile in the Cell; since wsadmin utility on all nodes in the Cell should connect to ND.

```
<WAS_NODE_PROFILE_HOME>/properties/wsadmin.properties
```

5. Delete the following sub-directories entirely.

```
<WAS_DMGR_PROFILE_HOME>/wstemp  
<WAS_DMGR_PROFILE_HOME>/config/temp
```

```
<WAS_NODE_PROFILE_HOME>/wstemp  
<WAS_NODE_PROFILE_HOME>/config/temp
```

6. After all documents for the ND (dmgr profile) and node profiles have been updated, start Deployment Manager using following command.

```
<WAS_DMGR_PROFILE_HOME>/bin/startManager (.bat/.sh)
```

7. Once DMGR process is running, execute the syncNode tool on each node in the cell containing the dmgr profile being modified. SyncNode will replicate the configuration containing the changed hostName property to the local node.

```
<WAS_NODE_PROFILE_HOME>/bin/syncNode (.bat/.sh) <dmgr_host> <dmgr_soap_port> -  
username <name> -password <password>
```

**Note:** syncNode command may take substantial time based on the size of the Cell, so give it



enough time to complete.

8. Start each nodeagent by following command and verify that nodeagent is visible in the ND Administrative Console.

```
<WAS_NODE_PROFILE_HOME>/bin/startNode (.bat/.sh)
```

9. Start servers on each node profile by following command.

```
<WAS_NODE_PROFILE_HOME>/bin/startServer (.bat/.sh) <server>
```

## **2.2 Procedure to alter the *hostName* property for a node profile federated in DMGR profile**

1. Always perform backupConfig before making significant configuration changes.
2. Ensure that nodeagent, jmsserver and all Application Server JVM processes have been stopped in the entire Cell. Also verify that no WebSphere java process or WASService.exe (on Windows) process is running.

To stop nodeagent, use following command from node profiles's bin directory.

```
<WAS_NODE_PROFILE_HOME>/bin/stopNode (.bat/.sh) -username <name> -password <password>
```

To stop Application Server, use following command from node profiles's bin directory.

```
<WAS_NODE_PROFILE_HOME>/bin/stopServer (.bat/.sh) <server> -username <name> -password <password>
```

3. Make all configuration changes through the deployment manager profile in an ND environment.
4. In the **serverindex.xml** file for the node profile being modified, alter the value of the **hostName** property for the **ServerIndex** element in the file. Search for string "**hostName=**" to find this property.

Also in the **serverindex.xml** file for the node profile being modified, alter the **host** property for every **EndPoint** in the file. Search all occurrences of "**host=**" string in following **serverindex.xml**

```
<WAS_DMGR_PROFILE_HOME>/config/cells/<cell>/nodes/<node>/serverindex.xml
```

5. Wsadmin utility in every node of the Cell should connect to ND; make sure ND host is specified for **com.ibm.ws.scripting.host** in the **wsadmin.properties** file.

```
<WAS_NODE_PROFILE_HOME>/properties/wsadmin.properties
```

6. Delete the following sub-directories entirely.

```
<WAS_NODE_PROFILE_HOME>/wstemp
```



<WAS\_NODE\_PROFILE\_HOME>/config/temp

7. After all documents for the node have been updated in dmgr profile configuration, execute the syncNode tool on each node in the cell containing the node being modified. SyncNode will replicate the configuration containing the changed hostName property to the local node profile. Once syncNode has been executed, the node should be functional again within the cell, with a new hostName property.

```
<WAS_NODE_PROFILE_HOME>/bin/syncNode (.bat/.sh) <dmgr_host> <dmgr_soap_port>  
-username <name> -password <password>
```

**Note:** syncNode command may take substantial time based on the size of the Cell, so give it enough time to complete.

10. Start nodeagent by following command and verify that nodeagent is visible in the ND Administrative Console.

```
<WAS_NODE_PROFILE_HOME>/bin/startNode (.bat/.sh)
```

11. Start each server on node profile by following command.

```
<WAS_NODE_PROFILE_HOME>/bin/startServer (.bat/.sh) <server>
```

### **2.3 Procedure to alter hostname property for a standalone Base NODE profile**

1. Always perform backupConfig before making significant configuration changes.
2. Ensure that all server processes have been stopped on this standalone node profile. Also verify that no WebSphere java process or WASService.exe process is running.

```
<WAS_NODE_PROFILE_HOME>/bin/stopServer (.bat/.sh) <server>
```

3. In the **serverindex.xml** file for the node being modified, alter the value of the hostName property for the ServerIndex element in the file. Search for "**hostName=**" to find this property.

Also in the serverindex.xml file for the node being modified, alter the host property for every EndPoint in the file. Search all occurrences of "**host=**" string to find these properties.

```
<WAS_NODE_PROFILE_HOME>/config/cells/<cell>/nodes/<node>/serverindex.xml
```

4. Edit the **wsadmin.properties** file in the properties subdirectory under the installation root. Change the value of the **com.ibm.ws.scripting.host** property in that file to the new host address.

```
<WAS_NODE_PROFILE_HOME>/properties/wsadmin.properties
```

5. Delete the following sub-directories entirely.

```
<WAS_NODE_PROFILE_HOME>/wstemp  
<WAS_NODE_PROFILE_HOME>/config/temp
```



6. Start each server on node profile by following command.

```
<WAS_NODE_PROFILE_HOME>/bin/startServer (.bat/.sh) <server>
```

**Note:** See section “**WebSphere Configuration Archive**” for alternative wsadmin scripting option to **alter hostname property for a standalone Base NODE profile.**



## 3 Changing the node name of WSAS node profile

Node names are also the name of the node configuration context (directory) for that node in WSAS 6.x. This means that the node name is a property value in configuration files and also the name of a directory within the configuration repository. Nodes exist in the configuration for both Base and ND WSAS profiles.

The node name is selected during profile creation process, and it is best to define the desired node name at that time, since altering the node name after product install is an involved procedure. It is especially important to be settled on the name of the node before federating that node into a cell, using the addNode utility. Once part of a cell, the node name is stored in config files in the master cell repository, so altering the name of the node becomes even more involved.

There is no feature in the browser-based admin console for WSAS to modify the name property associated with a particular node. A wsadmin script can be written to modify the node name property if this is necessary after the product has been installed. The following procedure can be used to manually alter the name of a node.

### 3.1 Procedure to alter the node name property for a standalone Base profile

1. Always perform backupConfig before making significant configuration changes.
2. Ensure that all server processes has been stopped on this node. Also verify that no WebSphere java process or WASService.exe process is running.

```
<WAS_NODE_PROFILE_HOME>/bin/stopServer (.bat/.sh) <server> -username <name> -password <password>
```

3. Execute the appropriate operating system command to change the name of the node directory (the subdirectory under config/cells/<cellname>/nodes) to the desired new node name. For example changing oldNode to newNode:

```
<WAS_NODE_PROFILE_HOME>/config/cells/myCell/nodes>/oldNode  
<WAS_NODE_PROFILE_HOME>/config/cells/myCell/nodes>/newNode
```

4. Edit and alter the **node.xml** file in the node directory. Change the name property to the new node name value. Search for string “**name=**” to find the property.

```
<WAS_NODE_PROFILE_HOME>/config/cells/<cell>/nodes/<node>/node.xml
```

5. Edit and alter the <WAS\_NODE\_PROFILE\_HOME>/bin/**setupCmdLine** (.bat/.sh) file. Change the WAS\_NODE variable value to the new node name.
6. Edit the <WAS\_NODE\_PROFILE\_HOME>/config/cells/<cell>/**security.xml** file in the cell directory. Change all occurrences of the old node name to the new name in that file. The node name should appear in the security.xml file as part of the sslConfig aliases defined in that file. Search for all occurrences of string “**sslConfig=**” to find these properties.
7. Edit the <WAS\_NODE\_PROFILE\_HOME>/config/cells/<cell>/nodes/<node>/**resources.xml** file in the node directory. Change all occurrences of the old node name to the new one in that



file. The node name should appear in the resources.xml file as part of the node aliases defined in that file. Search for all occurrences of string “**node=**” to find these properties.

8. Edit every **server.xml** file in server directories under this node. Change all occurrences of the old node name to the new node name. The node name in the server.xml file is usually part of sslConfig alias (defined in the security.xml file previously edited). Search for string “**sslConfig=**” to find this property.

```
<WAS_NODE_PROFILE_HOME>/config/cells/<cell>/nodes/<node>/servers/<server>/server.xml
```

9. Edit the **deployment.xml** file under each installed application directory. Alter the nodeName property of the deploymentTargets element in that file to change the old node to the new node name. Search for string “**nodeName=**” to find this property.

```
<WAS_NODE_PROFILE_HOME>/config/cells/<cell>/applications/<app.ear>/deployments/<app>/deployment.xml
<WAS_NODE_PROFILE_HOME>/config/cells/<cell>/applications/DefaultApplication.ear/deployments/DefaultApplication/deployment.xml
<WAS_NODE_PROFILE_HOME>/config/cells/<cell>/applications/ivtApp.ear/deployments/ivtApp/deployment.xml
<WAS_NODE_PROFILE_HOME>/config/cells/<cell>/applications/SamplesGallery.ear/deployments/SamplesGallery/deployment.xml
<WAS_NODE_PROFILE_HOME>/config/cells/<cell>/applications/TechnologySamples.ear/deployments/TechnologySamples/deployment.xml
<WAS_NODE_PROFILE_HOME>/config/cells/<cell>/applications/PlantsByWebSphere.ear/deployments/PlantsByWebSphere/deployment.xml
<WAS_NODE_PROFILE_HOME>/config/cells/<cell>/applications/petstore.ear/deployments/petstore/deployment.xml
```

10. Search all the template files under the config/templates directory for any that contain the old node name value. Change all occurrences of the old node name in the config templates to the new name.

```
<WAS_NODE_PROFILE_HOME>/config/templates/
```

11. Delete the following sub-directories entirely.

```
<WAS_NODE_PROFILE_HOME>/wstemp
<WAS_NODE_PROFILE_HOME>/config/temp
```

12. Start each server on node by following command.

```
<WAS_NODE_PROFILE_HOME>/bin/startServer (.bat/.sh) <server>
```

**Note:** See section “**WebSphere Configuration Archive**” for alternative wsadmin scripting option to **alter nodename property for a standalone Base NODE profile**.

### ***3.2 Procedure to alter the node name property for a federated Base node profile***

Changing the name of a node that has already been federated into a cell involves making config file changes to the master cell repository and node’s repository and then performing a syncNode on the local node to replicate the changes onto the local node config files.



1. Always perform backupConfig (on the master cell configuration repository in ND) before making significant configuration changes.
2. Ensure that Deployment Manager, nodeagent, jmsserver and all server processes have been stopped in the entire Cell. Also verify that no WebSphere java process or WASService.exe process is running.

To stop Deployment Manager, use following command from dmgr profiles's bin directory.  
<WAS\_DMGR\_PROFILE\_HOME>/bin/stopManager (.bat/.sh) -username <name> -password <password>

To stop nodeagent, use following command from node profiles's bin directory.  
<WAS\_NODE\_PROFILE\_HOME>/bin/stopNode (.bat/.sh) -username <name> -password <password>

To stop Application Server, use following command from node profiles's bin directory.  
<WAS\_NODE\_PROFILE\_HOME>/bin/stopServer (.bat/.sh) <server> -username <name> -password <password>

3. In the master cell repository, execute the appropriate operating system command to change the name of the node directory (the subdirectory under config/cells/<cellname>/nodes) to the desired new node name. For example changing oldNode to newNode.

```
<WAS_DMGR_PROFILE_HOME>/config/cells/myCell/nodes>/oldNode  
<WAS_DMGR_PROFILE_HOME>/config/cells/myCell/nodes>/newNode
```

Repeat the step on Node also.

```
<WAS_NODE_PROFILE_HOME>/config/cells/myCell/nodes>/oldNode  
<WAS_NODE_PROFILE_HOME>/config/cells/myCell/nodes>/newNode
```

4. In the master cell repository, edit and alter the **node.xml** file in the node directory. Change the name property to the new node name value. Search for string "**name=**" to find the property.

```
<WAS_DMGR_PROFILE_HOME>/config/cells/<cell>/nodes/<node>/node.xml
```

Repeat the step on Node also.

```
<WAS_NODE_PROFILE_HOME>/config/cells/<cell>/nodes/<node>/node.xml
```

5. In the master cell repository, edit the **security.xml** file in the cell directory. Change all occurrences of the old node name to the new one in that file. The node name should appear in the security.xml file as part of the sslConfig aliases defined in that file. Search for string "**sslConfig=**" to find this property.

```
<WAS_DMGR_PROFILE_HOME>/config/cells/<cell>/security.xml
```

6. In the master cell repository, edit the **resources.xml** file in the node directory. Change all occurrences of the old node name to the new one in that file. The node name should appear in the resources.xml file as part of the node aliases defined in that file. Search for all occurrences of string "**node=**" to find these properties.

```
<WAS_DMGR_PROFILE_HOME>/config/cells/<cell>/nodes/<node>/resources.xml
```

7. In the master cell repository, edit the **deployment.xml** file under each installed application directory. Alter the nodeName property of the deploymentTargets element in that file to



change the old node to the new node name.  
Search for string "nodeName=" to find this property.

```
<WAS_DMGR_PROFILE_HOME>/config/cells/<cell>/applications/<app.ear>/deployments/<app>/deployment.xml
<WAS_DMGR_PROFILE_HOME>/config/cells/<cell>/applications/DefaultApplication.ear/deployments/DefaultApplication/deployment.xml
<WAS_DMGR_PROFILE_HOME>/config/cells/<cell>/applications/ivtApp.ear/deployments/ivtApp/deployment.xml
<WAS_DMGR_PROFILE_HOME>/config/cells/<cell>/applications/SamplesGallery.ear/deployments/SamplesGallery/deployment.xml
<WAS_DMGR_PROFILE_HOME>/config/cells/<cell>/applications/TechnologySamples.ear/deployments/TechnologySamples/deployment.xml
<WAS_DMGR_PROFILE_HOME>/config/cells/<cell>/applications/PlantsByWebSphere.ear/deployments/PlantsByWebSphere/deployment.xml
<WAS_DMGR_PROFILE_HOME>/config/cells/<cell>/applications/petstore.ear/deployments/petstore/deployment.xml
```

**Note:** Check for the node name needs to be changed in the deployment.xml file. Change the node name, only if the application is installed on the node.

8. In the master cell repository, edit every **server.xml** file in server directories under this node. Change all occurrences of the old node name to the new node name. The node name in the server.xml file is usually part of a sslConfig alias (defined in the security.xml file previously edited). Search for string "**sslConfig=**" to find this property.

```
<WAS_DMGR_PROFILE_HOME>/config/cells/<cell>/nodes/<node>/servers/<server>/server.xml
```

9. In the master cell repository, edit the **cluster.xml** file for each cluster under the /clusters directory. Change all occurrences of the old node name to the new node name. Search for string "**nodeName=**" to find this property.

```
<WAS_DMGR_PROFILE_HOME>/config/cells/<cell>/clusters/<cluster>/cluster.xml
```

10. In the master cell repository, search all the template files under the config/templates directory for any that contain the old node name value. Change all occurrences of the old node name in the config templates to the new name.

```
<WAS_DMGR_PROFILE_HOME>/config/templates
```

11. On the local node, edit and alter the **setupCmdLine** (.bat/.sh) file. Change the WAS\_NODE variable value to the new node name.

```
<WAS_NODE_PROFILE_HOME>/bin/setupCmdLine.bat(.sh)
```

12. Delete the following sub-directories entirely.

```
<WAS_NODE_PROFILE_HOME>/wstemp
<WAS_NODE_PROFILE_HOME>/config/temp
```

13. Start the deployment manager by following command from dmgr profiles's bin directory.

```
<WAS_DMGR_PROFILE_HOME>/bin/startManager (.bat/.sh)
```

14. Execute the syncNode utility on the local node to download and sync the master cell repository with the (renamed) node.



```
<WAS_NODE_PROFILE_HOME>/bin/syncNode (.bat/.sh) <dmgr_host> <dmgr_soap_port>  
-username <name> -password <password>
```

**Note:** syncNode command may take substantial time based on the size of the Cell, so give it enough time to complete.

15. Start nodeagent by following command and verify that nodeagent is visible in the ND Administrative Console.

```
<WAS_NODE_PROFILE_HOME>/bin/startNode (.bat/.sh)
```

16. Start servers including jmsserver on each node by following command.

```
<WAS_NODE_PROFILE_HOME>/bin/startServer (.bat/.sh) <server>
```



## 4 Changing cell name of WSAS install

Cell names are also the name of the cell configuration context (directory) for that cell in WSAS 6.x. This means that the cell name is a property value in configuration files and also the name of a directory within the configuration repository. Cell name exist in the configuration for both Base and ND WSAS

The cell name is selected during profile creation, and it is best to define the desired cell name at that time, since altering the cell name after product install is an involved procedure.

There is no feature in the browser-based admin console for WSAS to modify the name property associated with a particular cell. The following procedure can be used to manually alter the name of a cell.

### 4.1 Procedure to alter the cell name property for a standalone Base profile

1. Always perform backupConfig before making significant configuration changes.
2. Ensure that all server processes has been stopped on this node. Also verify that no WebSphere java process or WASService.exe process is running.

```
<WAS_NODE_PROFILE_HOME>/bin/stopServer (.bat/.sh) <server> -username <name> -password <password>
```

3. Execute the appropriate operating system command to change the name of the cell directory (the subdirectory under config/cells) to the desired new cell name.

```
<WAS_NODE_PROFILE_HOME>/config/cells/oldCell  
<WAS_NODE_PROFILE_HOME>/config/cells/newCell  
  
<WAS_NODE_PROFILE_HOME>/installedApps/oldCell  
<WAS_NODE_PROFILE_HOME>/installedApps/newCell
```

4. Edit and alter the **cell.xml** file in the cell directory. Change the name property to the new cell name value. For example oldCell to newCell. Search for string "**name=**" to find this property.

```
<WAS_NODE_PROFILE_HOME>/config/cells/<cell>/cell.xml
```

5. Edit and alter the **setupCmdLine** (.bat/.sh) file. Change the WAS\_CELL variable value to the new cell name.

```
<WAS_NODE_PROFILE_HOME>/bin/setupCmdLine (.bat/.sh)
```

6. Edit the **deployment.xml** file under each installed application directory. Alter the binariesURL property in the deployment.xml file to change the old cell name portion of the binariesURL path to the new cell name. Search for string "**binariesURL=**" to find this property.

```
<WAS_NODE_PROFILE_HOME>/config/cells/<cell>/applications/<app.ear>/deployments/<app>/deployment.xml  
<WAS_NODE_PROFILE_HOME>/config/cells/<cell>/applications/DefaultApplication.ear/deployments/DefaultApplica
```



```
tion/deployment.xml
<WAS_NODE_PROFILE_HOME>/config/cells/<cell>/applications/ivtApp.ear/deployments/ivtApp/deployment.xml
<WAS_NODE_PROFILE_HOME>/config/cells/<cell>/applications/SamplesGallery.ear/deployments/SamplesGallery/deployment.xml
<WAS_NODE_PROFILE_HOME>/config/cells/<cell>/applications/TechnologySamples.ear/deployments/TechnologySamples/deployment.xml
<WAS_NODE_PROFILE_HOME>/config/cells/<cell>/applications/PlantsByWebSphere.ear/deployments/PlantsByWebSphere/deployment.xml
<WAS_NODE_PROFILE_HOME>/config/cells/<cell>/applications/petstore.ear/deployments/petstore/deployment.xml
```

For example:

```
binariesURL="$(APP_INSTALL_ROOT)/oldCell/DefaultApplication.ear"
binariesURL="$(APP_INSTALL_ROOT)/newCell/DefaultApplication.ear"
```

7. Delete the following sub-directories entirely.

```
<WAS_NODE_PROFILE_HOME>/wstemp
<WAS_NODE_PROFILE_HOME>/config/temp
```

8. Start each server on node by following command.

```
<WAS_NODE_PROFILE_HOME>/bin/startServer (.bat/.sh) <server>
```

**Note:** See section “**WebSphere Configuration Archive**” for alternative wsadmin scripting option to **alter hostname property for a standalone Base NODE profile.**

## ***4.2 Procedure to alter the cell name property for federated Cell***

The procedure for altering the cell name in an ND environment is similar to the procedure for Base WAS, with the following exceptions.

1. Always perform backupConfig (on the master cell configuration repository in ND) before making significant configuration changes.
2. Ensure that Deployment Manager, nodeagent, jmsserver and all server processes has been stopped in the entire Cell. Also verify that no WebSphere java process or WASService.exe process is running.

To stop Deployment Manager, use following command from dmgr profiles’s bin directory.  
<WAS\_DMGR\_PROFILE\_HOME>/bin/stopManager (.bat/.sh) -username <name> -password <password>

To stop nodeagent, use following command from node profiles’s bin directory.  
<WAS\_NODE\_PROFILE\_HOME>/bin/stopNode (.bat/.sh) -username <name> -password <password>

To stop Application Server, use following command from node profiles’s bin directory.  
<WAS\_NODE\_PROFILE\_HOME>/bin/stopServer (.bat/.sh) <server> -username <name> -password <password>



3. In the master cell repository, execute the appropriate operating system command to change the name of the cell directory (the subdirectory under config/cells/<cellname>) to the desired new node name.

For example changing oldCell to newCell:

```
<WAS_DMGR_PROFILE_HOME>/config/cells/oldCell  
<WAS_DMGR_PROFILE_HOME>/config/cells/newCell
```

Repeat the step on Node also.

```
<WAS_NODE_PROFILE_HOME>/config/cells/oldCell  
<WAS_NODE_PROFILE_HOME>/config/cells/newCell
```

4. Edit and alter the **cell.xml** file in the cell directory. Change the name property to the new cell name value. For example oldCell to newCell. Search for string “**name=**” to find this property.

```
<WAS_DMGR_PROFILE_HOME>/config/cells/<cell>/cell.xml
```

5. Delete the following sub-directories entirely.

```
<WAS_DMGR_PROFILE_HOME>/wstemp  
<WAS_DMGR_PROFILE_HOME>/config/temp
```

6. On each of the nodes in the cell, edit the **setupCmdLine** file to change the WAS\_CELL variable to the new cell name.

```
<WAS_DMGR_PROFILE_HOME>/bin/setupCmdLine (.bat/.sh)  
<WAS_NODE_PROFILE_HOME>/bin/setupCmdLine (.bat/.sh)
```

7. Start the deployment manager.

```
<WAS_DMGR_PROFILE_HOME>/bin/startManager (.bat/.sh)
```

8. Execute the syncNode utility on the local node to download and sync the master cell repository with the (renamed) node.

```
<WAS_NODE_PROFILE_HOME>/bin/syncNode (.bat/.sh) <dmgr_host> <dmgr_soap_port>  
-username <name> -password <password>
```

**Note:** syncNode command may take substantial time based on the size of the Cell, so give it enough time to complete.

9. Start nodeagent by following command and verify that nodeagent is visible in the ND Administrative Console.

**Note:** On Windows platform, if Windows Services is used to stop nodeagent, use Windows Services panel to start it.

```
<WAS_NODE_PROFILE_HOME>/bin/startNode (.bat/.sh)
```

10. Start servers including jmsserver on each node by following command.

**Note:** On Windows platform, if Windows Services is used to stop servers, use Windows Services panel to start them.

```
<WAS_NODE_PROFILE_HOME>/bin/startServer (.bat/.sh) <server>
```



## 5 Changing Server Name of WSAS install

Individual server names are also the name of the server configuration context (directory) for that server in WSAS 6.x. Because the server name is a directory name, in addition to being a property value within configuration documents, the procedure for altering a server name is complex.

The suggested approach for modifying the name of a server is to use the admin console (or wsadmin) to create a second server definition based on the server to be modified. Keep everything the same for the new server except for its name. Specify the desired new server name. Once a server with the desired name has been defined, edit the deployment.xml for each application deployed on the original server and change the server name of the deploymentTargets property from old to new server. Once you have verified that the applications run on the newly named server, you can delete the definition of the original server.

**Note:** See section “**WebSphere Configuration Archive**” for alternative wsadmin scripting option to **alter hostname property for a standalone Base NODE profile.**



## 6 Repository Migration of WSAS install

Repository migration is the process of taking an entire configuration repository from one cell and modifying it for use by a different cell. This can apply for both Base WSAS and ND.

When considering repository migration, it is important to keep the operating system platforms in mind. The migration procedure described below assumes identical source and target node operating systems. If the operating systems for any of the nodes in the new cell are different than the equivalent nodes in the original cell, then additional modifications must be made to the configuration. Specifically, the values of the \$WAS\_INSTALL\_ROOT and \$JAVA\_HOME variables must be altered to match those applicable to the operating system on the target node. These variables are defined in the **variables.xml** file in the directory containing the node configuration data.

```
<WAS_DMGR_PROFILE_HOME>/config/cells/<cell>/nodes/<node>/variables.xml  
<WAS_NODE_PROFILE_HOME>/config/cells/<cell>/nodes/<node>/variables.xml
```

Another repository migration consideration is the fixpack level of the source and target nodes. The migration procedure described below assumes identical fixpack levels for both source and target nodes. Repository migration may work between nodes with different fixpack levels, but the user should consult the release notes for both levels of the product to be sure. Certain fixpack levels are known to include changes that require different configuration files or configuration file contents (for instance 6.x.1 and 6.x.2). Repository migration may not be possible between cells (or portions of cells) at different fixpack levels.

### 6.1 Procedure for Repository Migration of Standalone WSAS profile

In the Base WSAS profile scenario, make a total copy of the Base configuration repository (everything under the config directory). Then just follow the procedure for renaming the cell in the Base environment for the new copy of the config tree (procedure outlined above). If the migration is to a different host system, then the hostname modification procedure may need to be applied as well.

This procedure assumes migration from HostA, nodeA to HostB, NodeB.

**Note:** If the target Operating System is different than source Operating System, then some Operating System specific changes might be required on target Operating System, for example, migration from Windows to UNIX or UNIX to Windows.

1. Install standalone WSAS on **HostB**.
2. On **HostB**, run backupConfig (.bat/sh) using following command.  
<WAS\_NODE\_PROFILE\_HOME>/bin/backupConfig (.bat/sh) HostB.zip  
Move the backup configuration file HostB.zip to a safe place.
3. On **HostB**, remove entire <WAS\_NODE\_PROFILE\_HOME>/config/cell directory structure using appropriate Operating System command.
4. On **HostA**, run backupConfig (.bat/sh) using following command.  
<WAS\_NODE\_PROFILE\_HOME>/bin/backupConfig (.bat/sh) HostA.zip



Copy the backup configuration file HostA.zip to HostB under <WAS\_NODE\_PROFILE\_HOME>/bin directory.

5. On **HostB**, run restoreConfig (.bat/sh) using following command.  
<WAS\_NODE\_PROFILE\_HOME>/bin/restoreConfig (.bat/sh) HostA.zip
6. If you going to use the node name and cell name **NodeB** installed as, then use one of the procedure documented in this paper to change the node name and host name of repository.
7. On **NodeB**, if you want to use **NodeA** node name and cell name, then edit <WAS\_NODE\_PROFILE\_HOME>/bin/**setupCmdLine** (.bat/.sh) file for WAS\_NODE and WAS\_CELL variable to reflect NodeA node name and cell name.

**Note:** See section “**WebSphere Configuration Archive**” for alternative wsadmin scripting option to **alter hostname property for a standalone Base NODE profile**.

## ***6.2 Repository Migration of federated WSAS dmgr profile***

The ND environment poses greater challenges. Not only will the cell name be different, but the nodes and hosts for those nodes will potentially be different. The exact ND procedure to follow depends a great deal on the topology of the cell to be migrated (the source repository) and the topology of the target cell to which it is to be migrated.

If possible, keep the target cell the same topology as the old cell. It is possible (and recommended) that the names of the nodes in the new cell NOT be changed from the old cell. There is no problem if nodes in one cell are the same name as nodes in another cell. It is quite possible that the hostName value will have to be changed in the target environment. Follow the ND procedure for this documented above.

Install and setup the target cell structure before copying the source configuration over the top of it. Install and federate all nodes before the source configuration is laid over the target cell. Always perform backupConfig before copying the source repository over the target. Once the source repository has been copied over the top of the target cell, perform the renaming procedures as appropriate (ideally just the cell and hostName procedures). Be sure to perform syncNode on each of the nodes after the configuration updates.



## 7 WebSphere Configuration Archive

**Note:-** This only supported for standalone base only profile

The WebSphere Configuration Archive is new features in WebSphere Application Server v6.0. This feature allows a set of complete or subset of WebSphere configuration archive. WebSphere Application Server V6 provides a mechanism that allows you to export certain profiles, or server objects from a profile, to an archive. The archive can be distributed and imported to other installations. An exported archive is a zip file of the config directory with host-specific information removed. The recommended extension of the zip file is .car. The exported archive can be the complete configuration or a subset. Importing the archive creates the configurations defined in the archive. The target configuration of an archive export / import can be a specific server or an entire profile.

To use an archive you would:

1. Export a WebSphere configuration. This creates a zip file with the configuration.
2. Unzip the files for browsing or update for use on other systems. For example, you might need to update resource references.
3. Send the configuration to the new system. An import can work with the zip file or with the expanded format.
4. Import the archive. The import process requires that you identify the object in the configuration you want to import and the target object in the existing configuration. The target can be the same object type as the archive or its parent:
  - If you import a server archive to a server configuration the configurations are merged.
  - If you import a server archive to a node, the server is added to the node.

A tutorial on creating and using archives can be found in the Information Center. See

[ftp://ftp.software.ibm.com/software/eod/was/6.0/SystemManagement/WASv6\\_SM\\_Configuration\\_Archives/playershell.swf](ftp://ftp.software.ibm.com/software/eod/was/6.0/SystemManagement/WASv6_SM_Configuration_Archives/playershell.swf)

### Profile archives

The following command can be used to create an archive of a profile:

```
$AdminTask exportWasprofile {-archive <archive_location>}
```

You can only create an archive of an unfederated profile (standalone application server).

```
$AdminTask importWasprofile {-archive <archive_location>}
```

#### **exportWasprofile:**

Use the **exportWasprofile** command to export the entire cell configuration to a configuration archive. (myArchive.car)

- Using Jacl:



- `$AdminTask exportWasprofile {-archive c:\myCell.ear}`
- Using Jython string:  
`AdminTask.exportWasprofile(['-archive c:\myCell.ear'])`

#### Interactive mode example usage:

- Using Jacl:  
`$AdminTask exportWasprofile {-interactive}`
- Using Jython string:  
`AdminTask.exportWasprofile (['-interactive'])`

#### importWasprofile:

Use the **importWasprofile** command to import a cell configuration in the configuration archive to the system. Only a base single server configuration is supported for this command.

- Using Jacl:  
`$AdminTask importWasprofile {-archive c:\myCell.ear}`
- Using Jython string:  
`AdminTask.importWasprofile(['-archive c:\myCell.ear'])`

#### Interactive mode example usage:

- Using Jacl:  
`$AdminTask importWasprofile {-interactive}`
- Using Jython string:  
`AdminTask.importWasprofile (['-interactive'])`