

# **A Step-By-Step Guide to Configuring a WebSphere Portal v6.1.0.3/6.1.5 Cluster**

Hunter Tweed  
WebSphere Portal Level 2 support Team Lead  
IBM Raleigh Lab

November, 2009

© Copyright International Business Machines Corporation 2009. All rights reserved.

This guide describes a comprehensive procedure for installing, configuring, and building an IBM® WebSphere® Portal v6.1.0.3/6.1.5 cluster using:

- IBM WebSphere Application Server 7.0.0.5 – 32-bit
- Red Hat Enterprise Linux 4.0 update 8
- DB2 v9.1 fp 5 Server
- IBM Tivoli Directory Server v6.1
- IBM HTTP Server 6.1

## **Table of Contents**

A Step-By-Step Guide to Configuring a WebSphere Portal v6.1.0.3/6.1.5 Cluster.....	1
Table of Contents.....	2
Introduction.....	3
Before you begin.....	5
Install WebSphere Application Server v7 on the future Portal Primary Node.....	6
Install the Primary Portal Node.....	17
Install IBM Support Assistant Lite.....	23
Install the Deployment Manager.....	24
Configure the Deployment Manager.....	36
Configure the Primary Portal node to an external database.....	41
Federate and Cluster the Primary Node.....	46
Install WebSphere Application Server v7 on the future Portal Secondary Node.....	50
Install the Secondary Portal Node.....	61
Install IBM Support Assistant Lite.....	66
Federate and Cluster the Secondary Portal node.....	67
Configure the Portal Cluster for Security.....	71
Configure the Portal Cluster with an external web server.....	74
Appendix A – Create a Deployment Manager profile on the Primary Portal node.....	80
Appendix B – SQL Script to Create DB2 Databases.....	84
Appendix C – Adding a Vertical Cluster member.....	88
Appendix D – Adding a new secondary node to an existing cluster.....	92
Appendix E – Running IBM Support Assistant Lite.....	96
Appendix F – Common Problems.....	98

## **Introduction**

Building and configuring a cluster can be a very complex task. You can build portal clusters in various ways. This article provides a best practice approach for building a cluster environment using WebSphere Portal version 6.1.0.3/6.1.5. This example produces a two-node horizontal cluster, as shown in Figure 1. Your environment might require special considerations, but you should still follow this step-by-step approach as an overall guide.

### **Higher Versions of Portal**

Although this guide is specifically written for 32-bit Portal v6.1.0.3/6.1.5 and WSAS v7.0.0.5, the same approach will apply to any Portal v6.1.0.3 version or higher and any WSAS v7.0.0.x version, 32 or 64-bit.

### **WebSphere Application Server v6.1.0.27 or higher**

You may also use WAS v6.1.0.27 and higher as well, as long as you understand the following:

- You have the option of installing WAS v6.1.0.27 as part of the Portal v6.1.0.3/6.1.5 installation so there is no need to manually install it beforehand (unlike WAS v7). Therefore the Primary Node, Secondary Node, and Deployment Manager installation steps may be slightly different for you.
- Screenshots and paths within the Deployment Manager Administrative console may vary between WAS v6.1 and WAS v7. Screenshots and paths in this guide are from WAS v7.

### **Windows/Unix Differences**

This guide was written using Linux as the base operating system, however the steps/concepts listed in this guide are independent of operating system. That is, you can follow these same steps on any operating system and achieve the same result.

The only significant difference is that for Windows, you must use the batch file commands instead of the UNIX shell commands listed in this guide. For example:

**UNIX:** `./startServer.sh WebSphere_Portal`

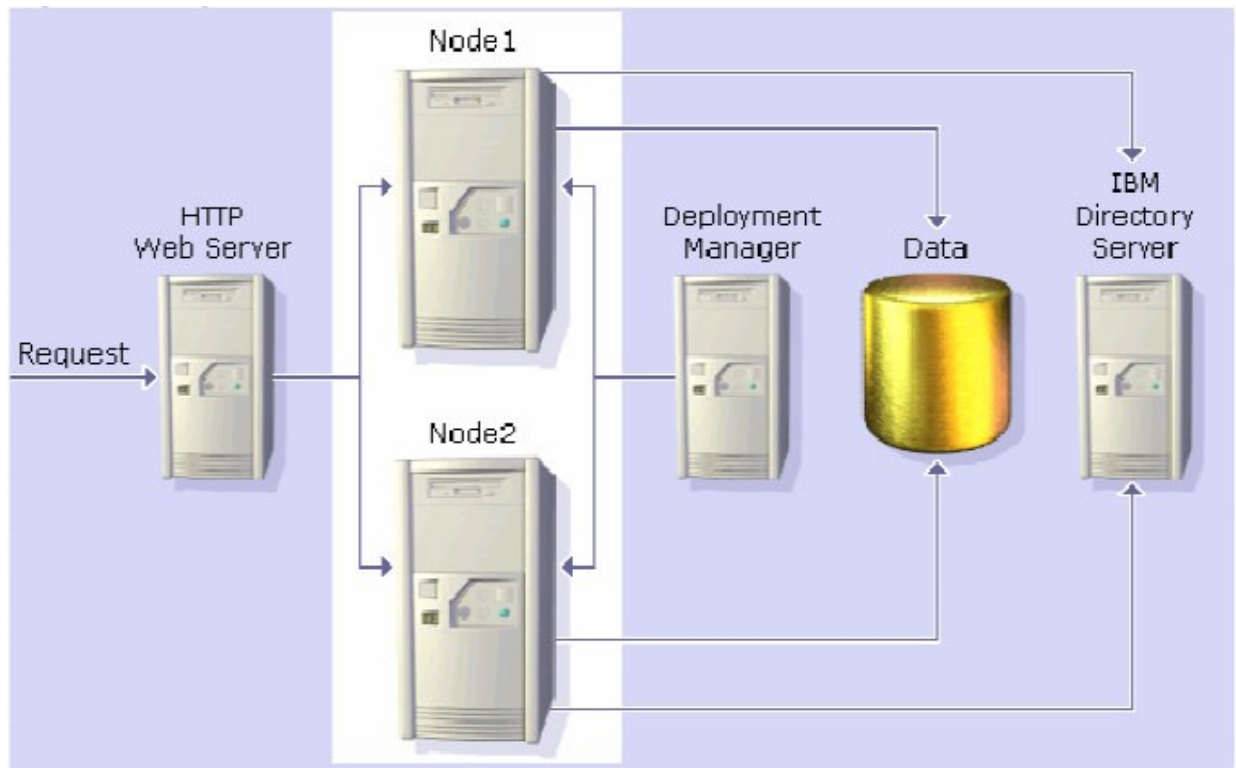
**Windows:** `startServer.bat WebSphere_Portal`

or

**UNIX:** `./ConfigEngine.sh cluster-node-config-cluster-setup`

**Windows:** `ConfigEngine.bat cluster-node-config-cluster-setup`

Figure 1 – Target Portal Cluster



In the instructions for configuring Portal with the database and LDAP, screens shots show valid examples. Use values which are appropriate for your database and LDAP.

## ***Before you begin***

This guide does **NOT** cover the following:

- Installing DB2
- Installing IBM Tivoli Directory Server
- Configuring the cluster with Web Content Management
- Configuring the cluster with WebSphere Process Server
- Configuring a dynamic cluster using WebSphere Application Server XD
- Creating multiple clusters in a single cell

For more information on these and other topics, please visit the IBM WebSphere Portal v6.1.0 Information Center:

[http://publib.boulder.ibm.com/infocenter/wpdoc/v6r1/index.jsp?topic=/com.ibm.wp.ent.doc\\_v615/welcome\\_main.html](http://publib.boulder.ibm.com/infocenter/wpdoc/v6r1/index.jsp?topic=/com.ibm.wp.ent.doc_v615/welcome_main.html)

To perform the tasks described in this document, you need basic WebSphere Portal and WebSphere Application Server knowledge and administration skills. Some steps might require the assistance of another system administrator, such as the database administrator or LDAP administrator.

The following references to WebSphere Portal and WebSphere Application Server file paths will be used throughout the guide:

<AppServer root> - The root path of the AppServer directory, for example:  
/opt/WebSphere/AppServer

<PortalServer root> - The root path of the PortalServer directory, for example:  
/opt/WebSphere/PortalServer

<wp\_profile> - The root path of the wp\_profile directory, for example:  
/opt/WebSphere/wp\_profile

<dmgr\_profile> - The root path of the dmgr profile directory, for example:  
/opt/WebSphere/AppServer/profiles/Dmgr01

<plugin root> - The root path of the WebSphere Plugin directory, for example:  
/opt/WebSphere/Plugins

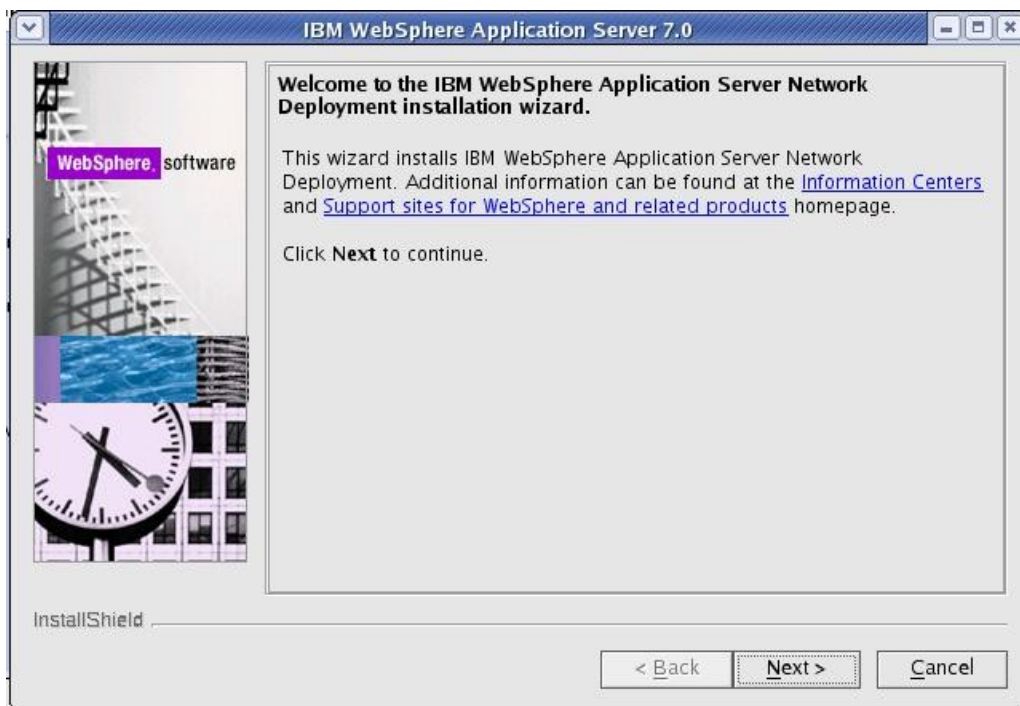
## ***Install WebSphere Application Server v7 on the future Portal Primary Node***

In this section, you will install WebSphere Application Server v7.0.0.0 on the future Portal primary Node, and upgrade it to v7.0.0.5. WebSphere Application Server v7 is NOT provided with the WebSphere Portal v6.1.5 bundle so you must obtain the installation media and license elsewhere.

1. From the WAS v7 installation CD or image, launch the installer located in the WAS directory:

`./install`

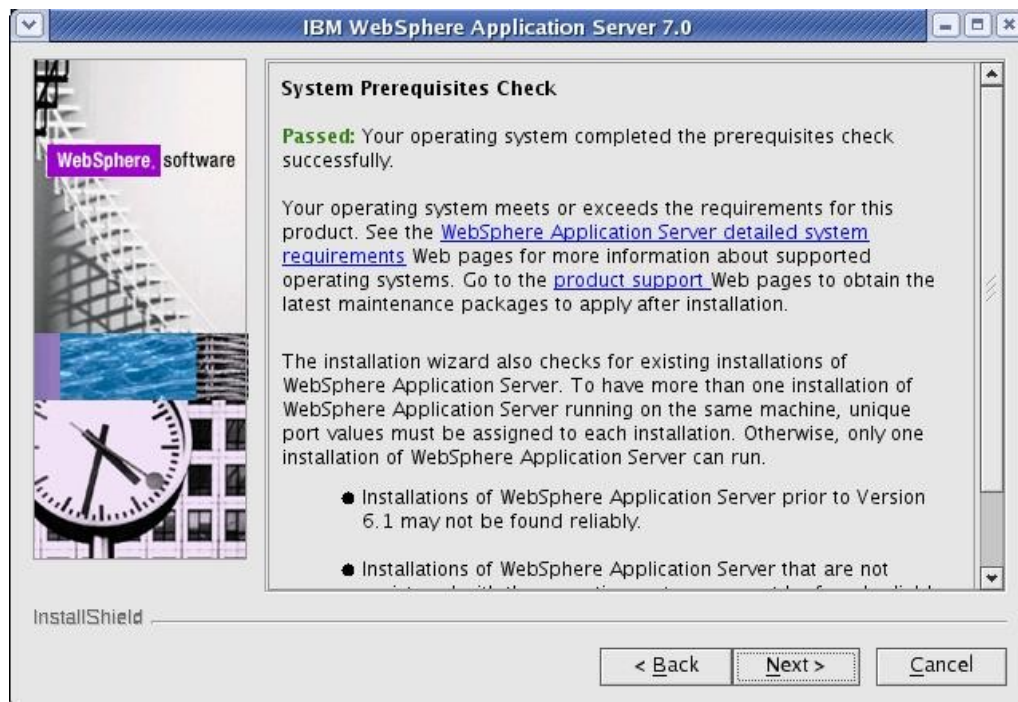
2. Click 'Next' on the Welcome Screen:



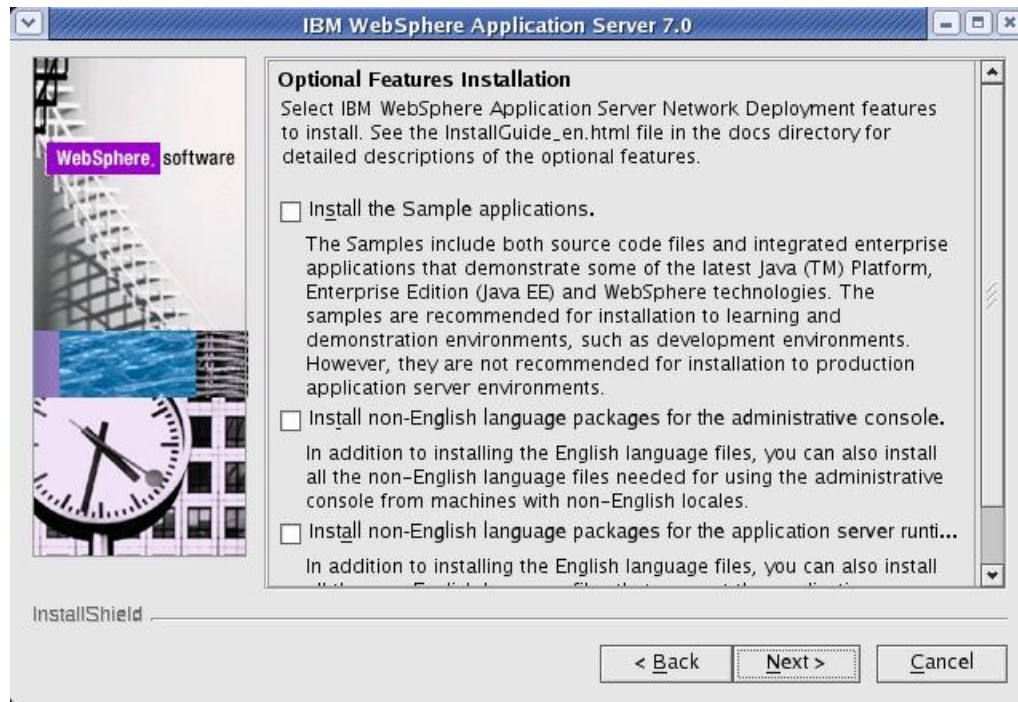
3. Accept the license and click 'Next':



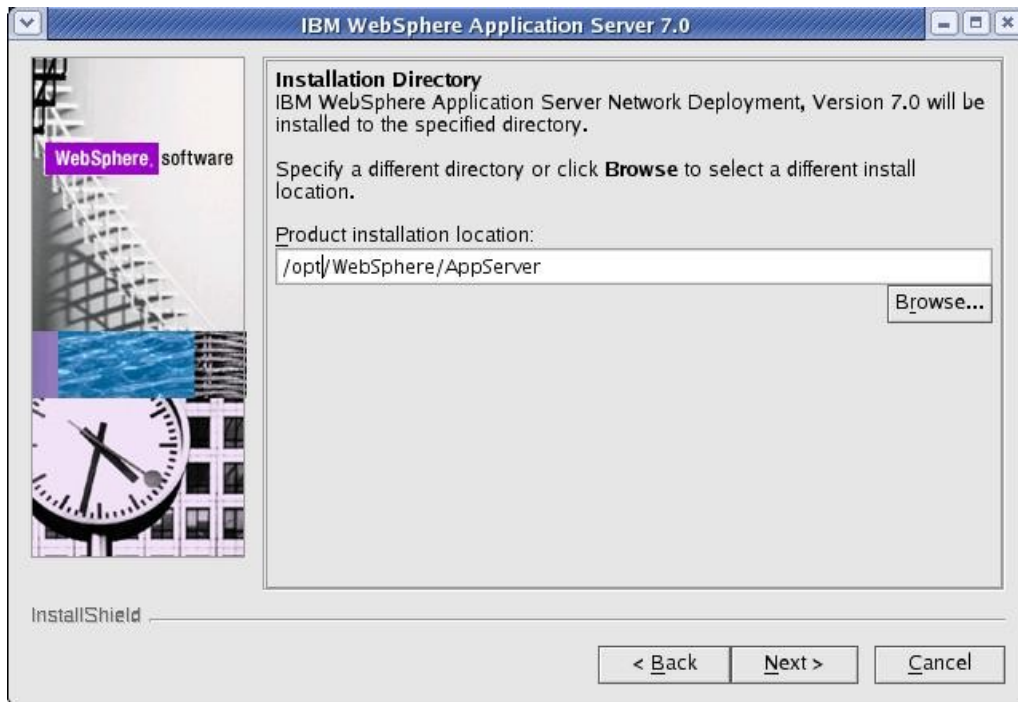
4. Click 'Next' on the Systems Prerequisite Check screen:



5. Do not select any options, click 'Next'.



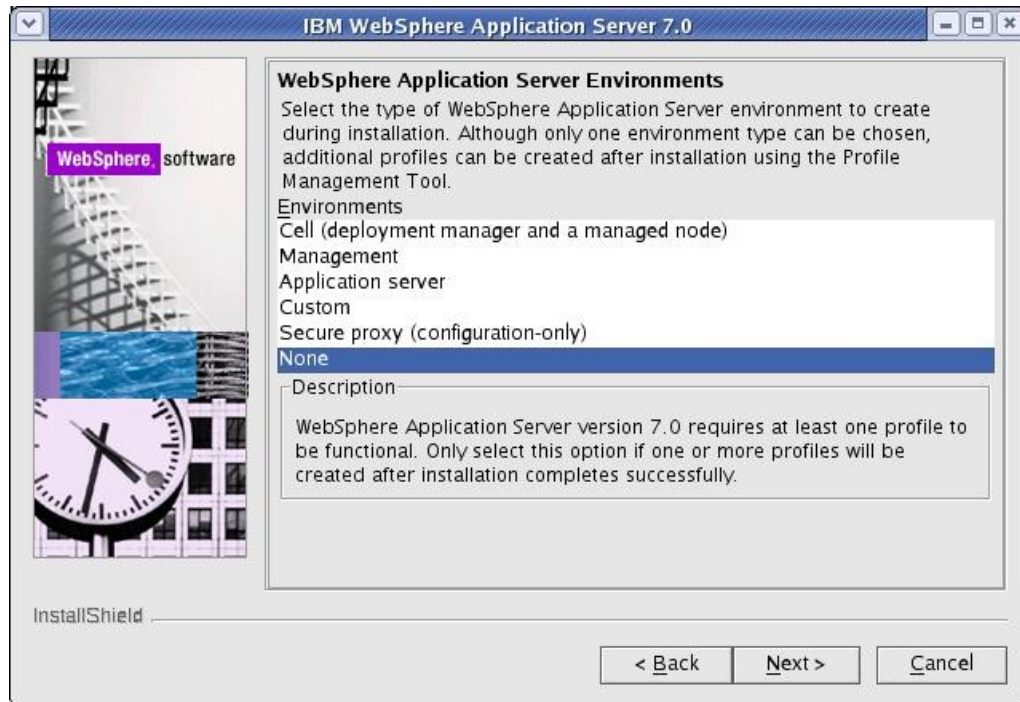
6. Select your installation directory and click 'Next':





7. **Do not** select to create a profile.

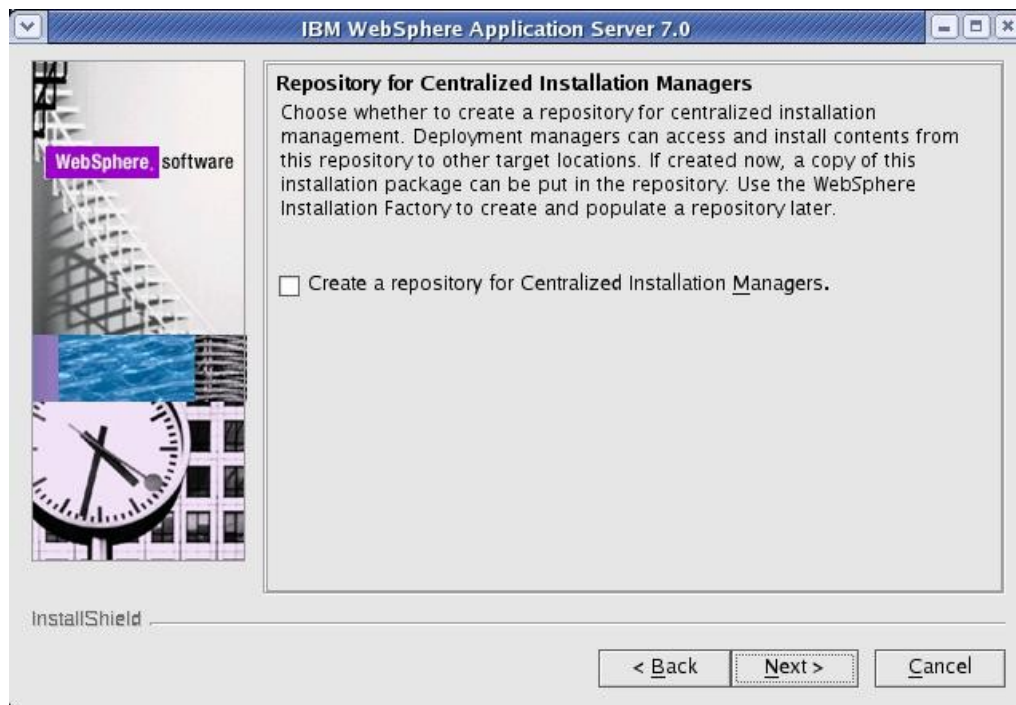
**Note:** The WebSphere Portal installer will create its own WAS profile so there is no need to create a profile here. If you do create a profile, WebSphere Portal will not use it.



8. Click 'Yes' on the warning that pops up when you select no profile:



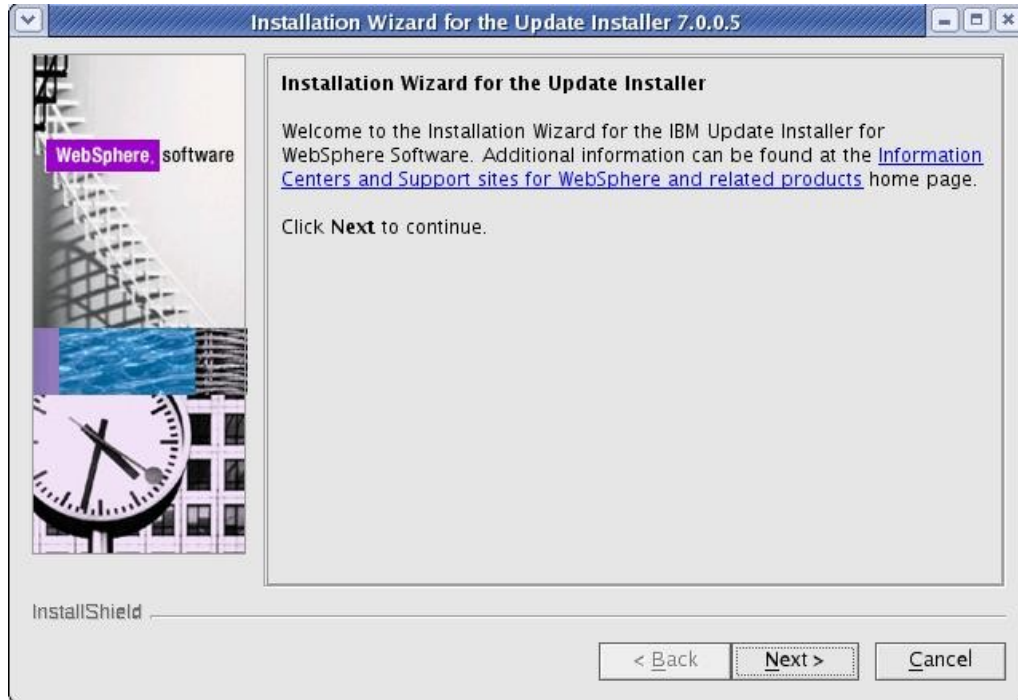
9. Check the option to create a repository for Centralized Installation Managers if you'd like and click Next. In this guide, the option is not checked:



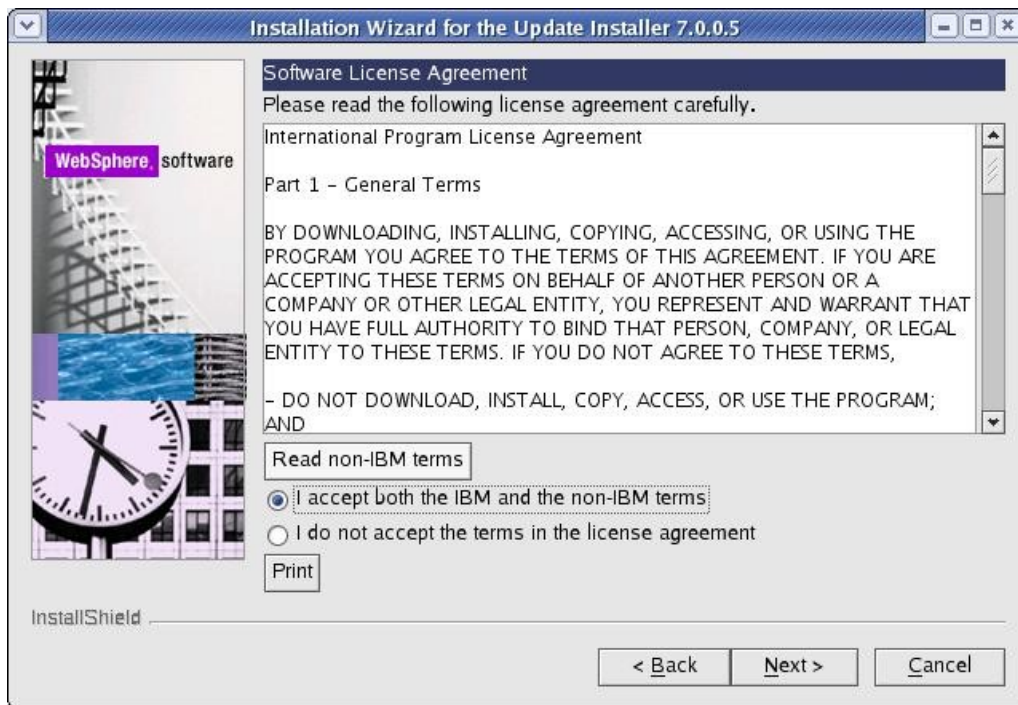
10. Review the information on the summary screen and click 'Next' to begin the installation.
11. After the installation completes, uncheck the option to create a new profile and click 'Finish' to exit the installation program.
12. Download the WebSphere Application Server v7 Update Installer:  
<http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg24020446>
13. Extract the download into a temporary directory and launch the installer located in the <temp location>/UpdateInstaller directory:

```
./install
```

14. Click 'Next' on the Welcome Screen:

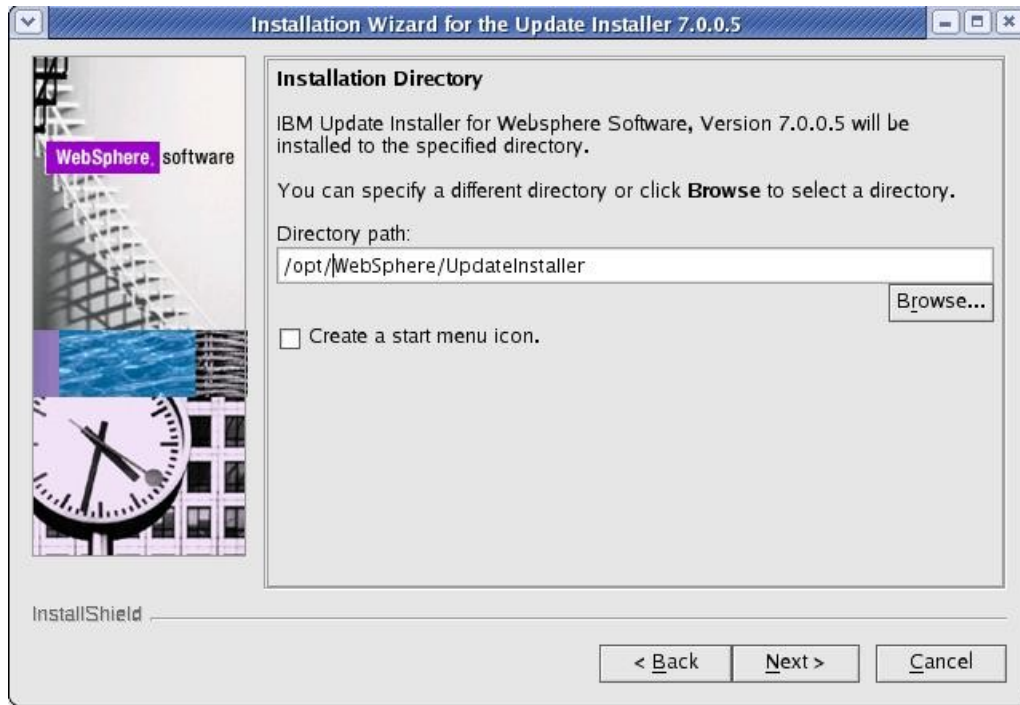


15. Accept the license and click 'Next':



16. Click 'Next' on the System Pre-requisite check screen.

17. Select the path where you would like to install the WAS Update Installer:



18. After the installation completes, uncheck the 'Launch' button and click Finish to exit the installer.

19. Download the WAS 7.0.0.5 fixpack and the corresponding JDK upgrade:

<http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg24023705>

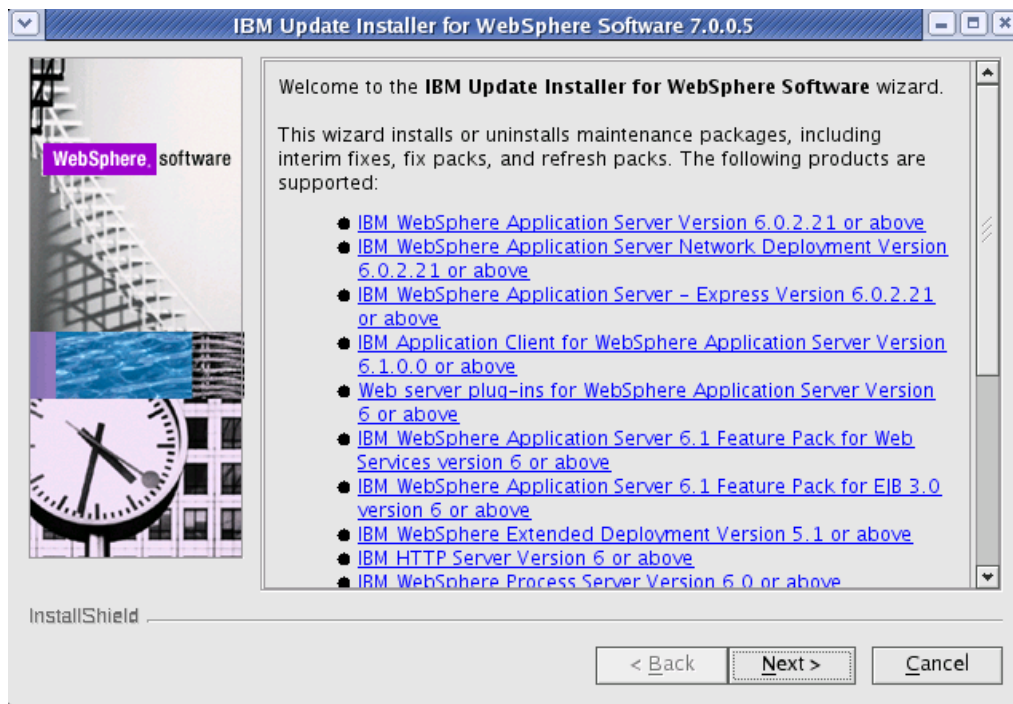
<http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg24023708>

20. Copy the 7.0.0.5 fixpack and the JDK upgrade to the <UpdateInstaller root>/maintenance directory, where <UpdateInstaller root> is the location you selected in step 17.

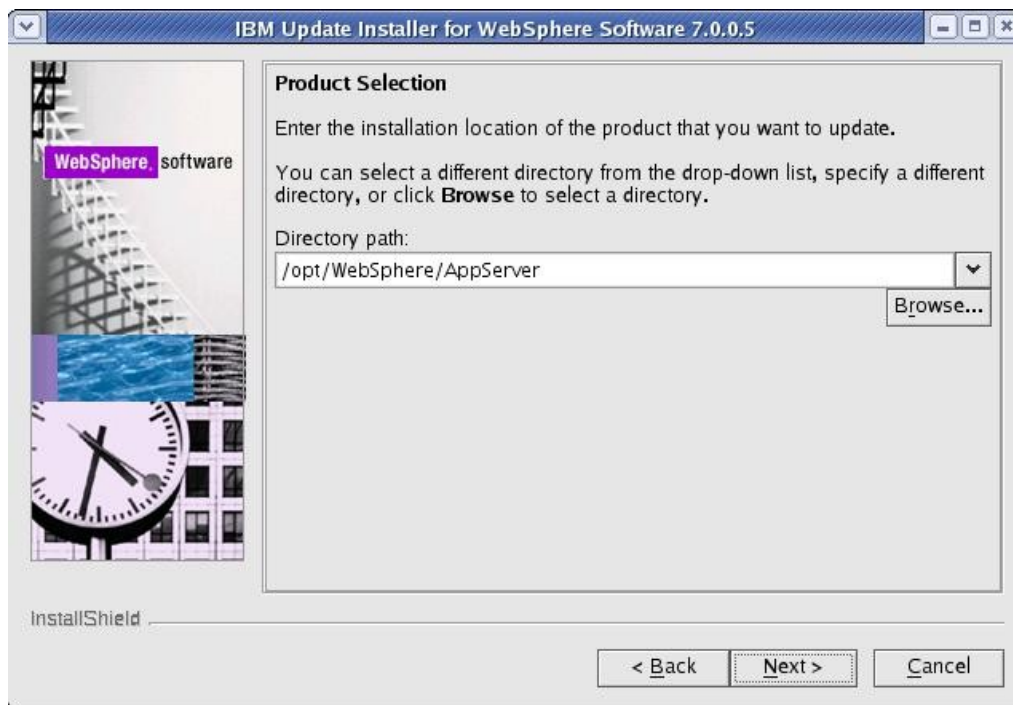
21. Launch the WAS Update Installer from the UpdateInstaller directory you set from step 17:

`./update.sh`

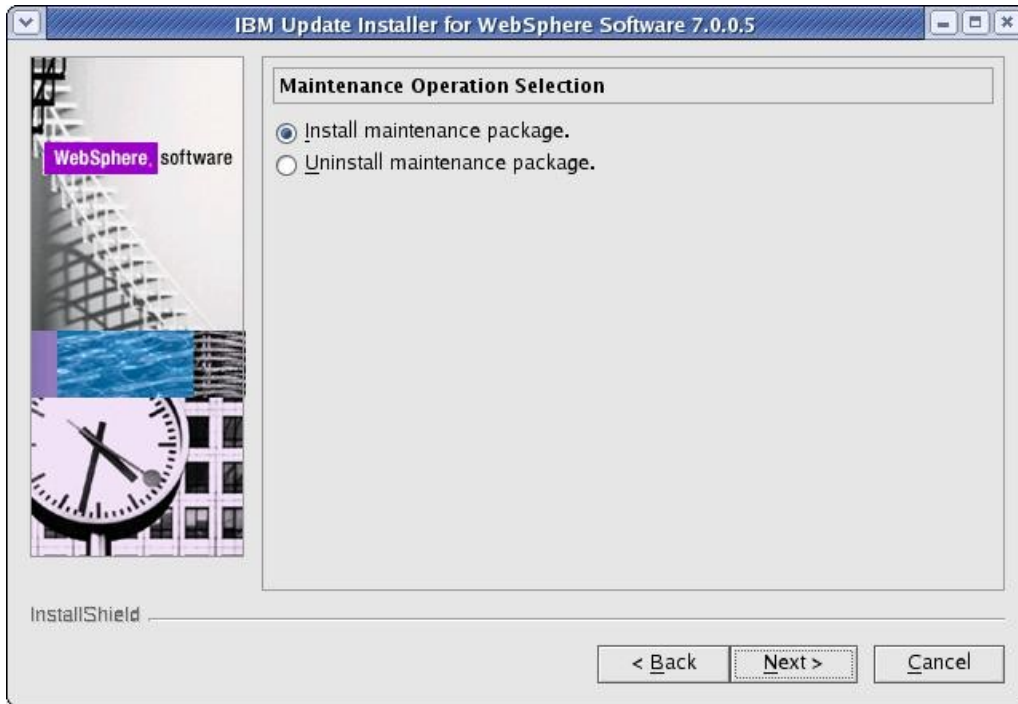
22. Click 'Next' on the Welcome Screen:



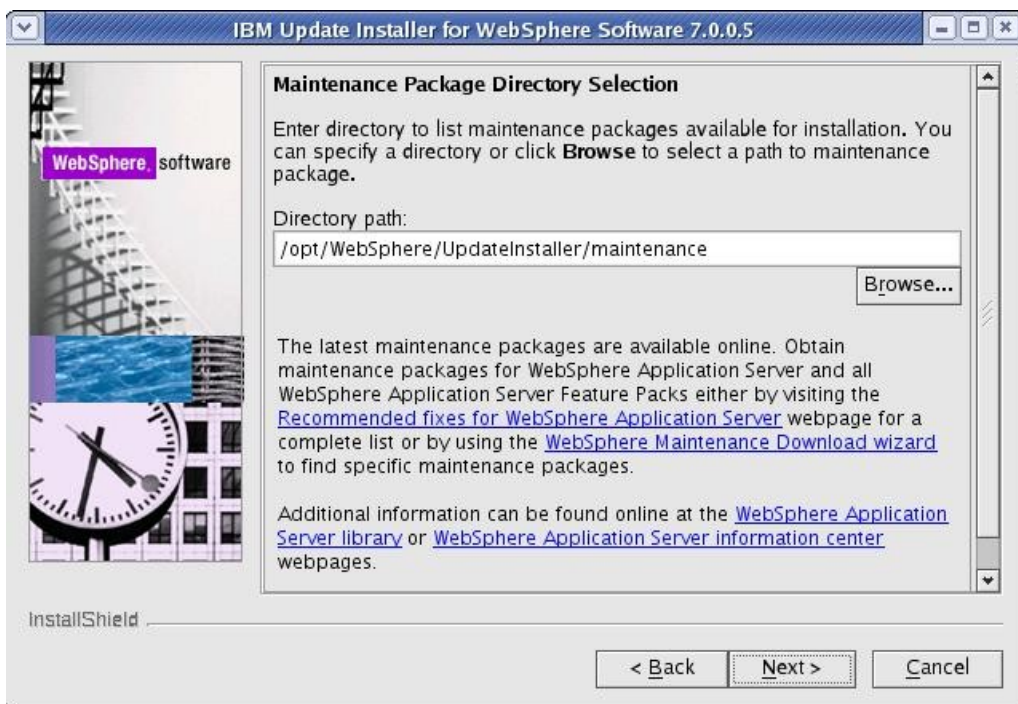
23. Select the WebSphere Application Server directory you wish to upgrade and click 'Next':



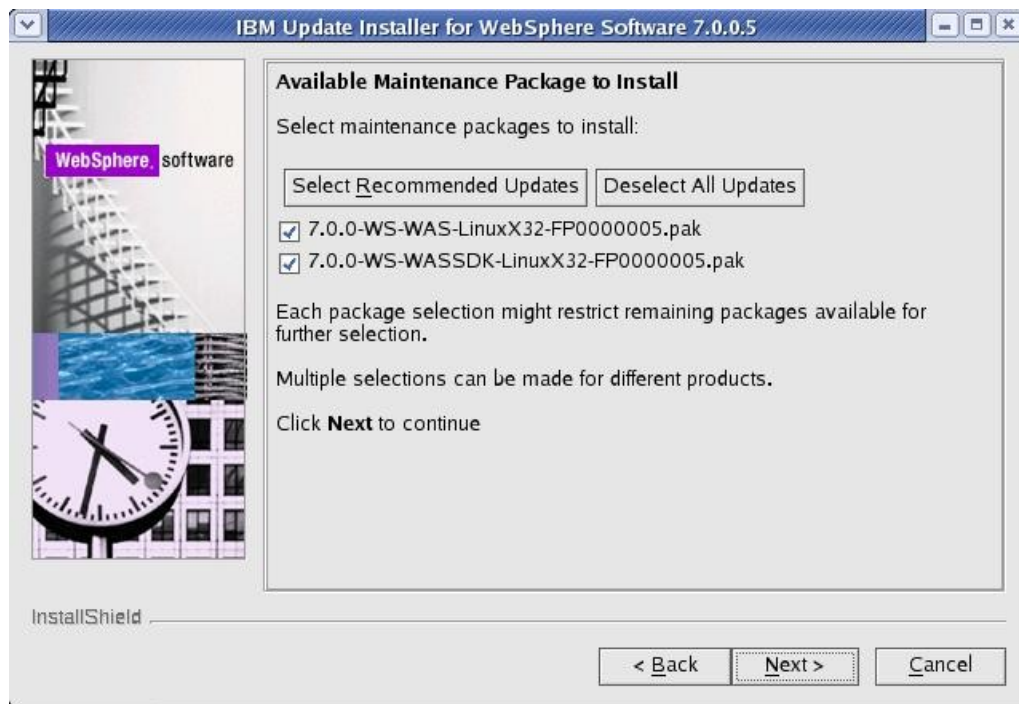
24. Select the 'Install Maintenance package" option and click 'Next':



25. Select the directory that contains the 7.0.0.5 and JDK packages:



26. Check the boxes for the 7.0.0.5 and JDK packages and click 'Next':



27. On the installation summary screen, click 'Next' to begin the upgrade.

28. After the upgrade completes, click 'Finish' to exit the update installer.

29. Download the required WebSphere Application Server interim fixes for WAS v7.0.0.5 when using WebSphere Portal v6.1.0.3/6.1.5 from the WebSphere Portal Support site:

<http://www-01.ibm.com/support/docview.wss?rs=688&uid=swg24023148>

30. Download the fixes into a temporary directory. This should include the following interim fixes:

PK90343  
PK91698  
PK92047  
PK93952  
PK96275  
PK97321  
PK98302  
PK98741  
PK99787  
PM00692

31. Copy the fixes into the <UpdateInstaller root>/maintenance directory where <UpdateInstaller root> is the location you selected in step 17.
32. Repeat steps 21-28 to install the interim fixes. (Note that for step 26 you will select to install the interim fixes listed in Step 30, not the 7005 and JDK fixpacks).



## Install the Primary Portal Node

In this section, you will install the primary Portal node. You will use the WAS v7005 that you installed from the previous section as the base for this Portal installation. All of the steps in this section will be done on the server you intend to use as your primary node.

1. Open a terminal window and enter:

```
ping yourserver.yourcompany.com
```

where *yourserver.yourcompany.com* is your actual fully qualified hostname.

2. Enter:

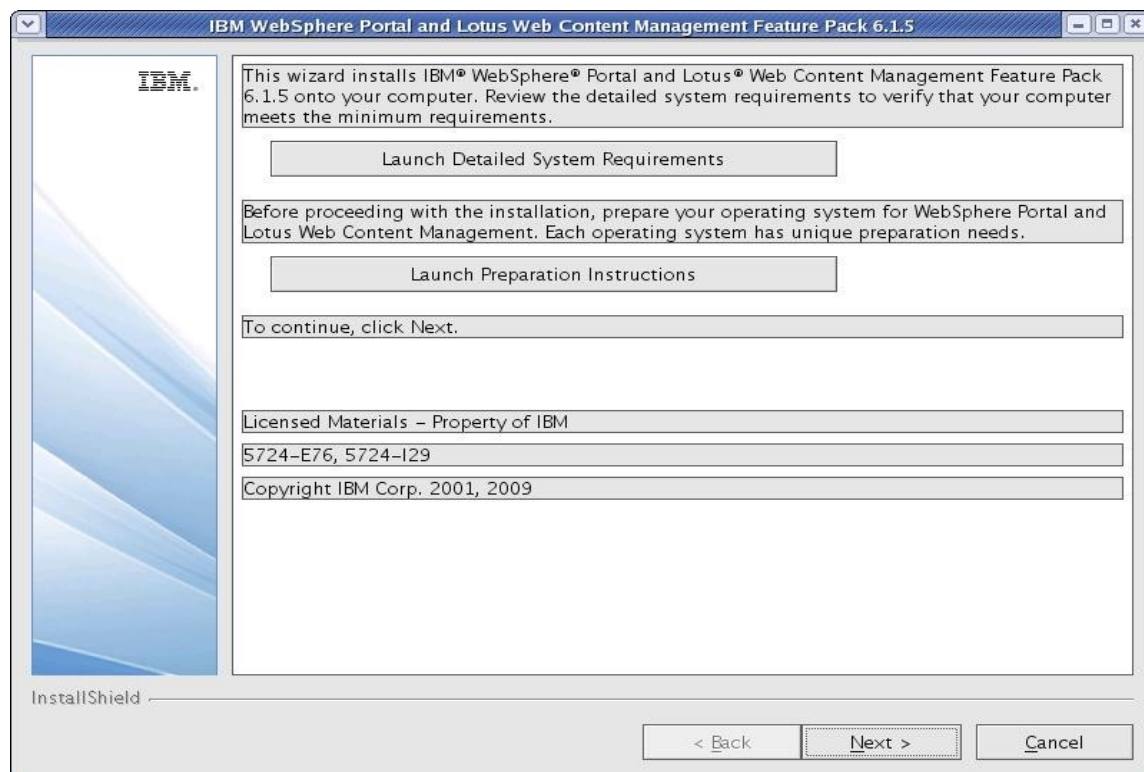
```
ping localhost
```

to verify the “localhost” network settings are configured properly on your machine.

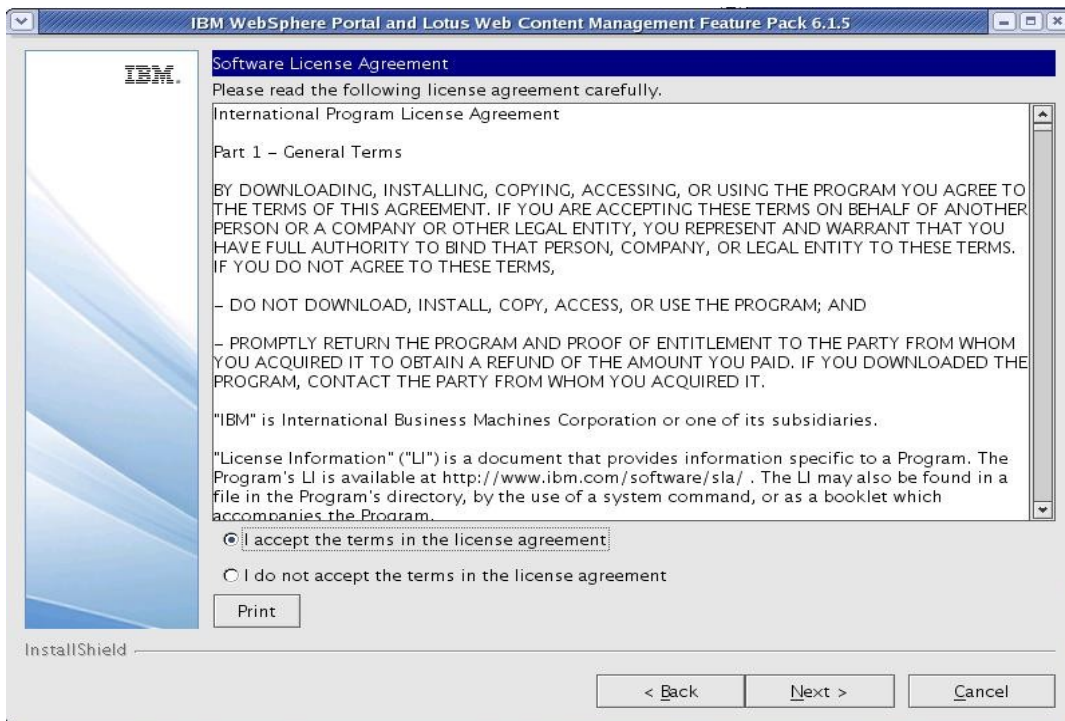
3. From the IL-Setup CD, launch the WebSphere Portal installer:

```
./install.sh
```

4. Click 'Next' on the Welcome screen.

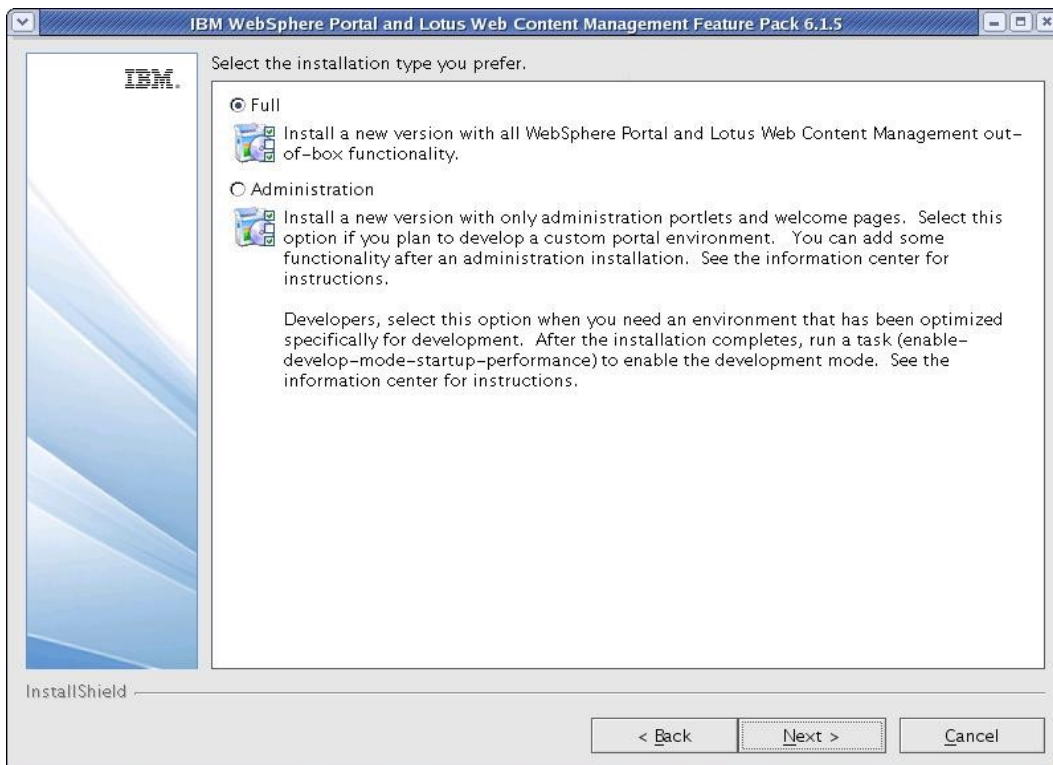


5. Accept the license agreement and click 'Next':

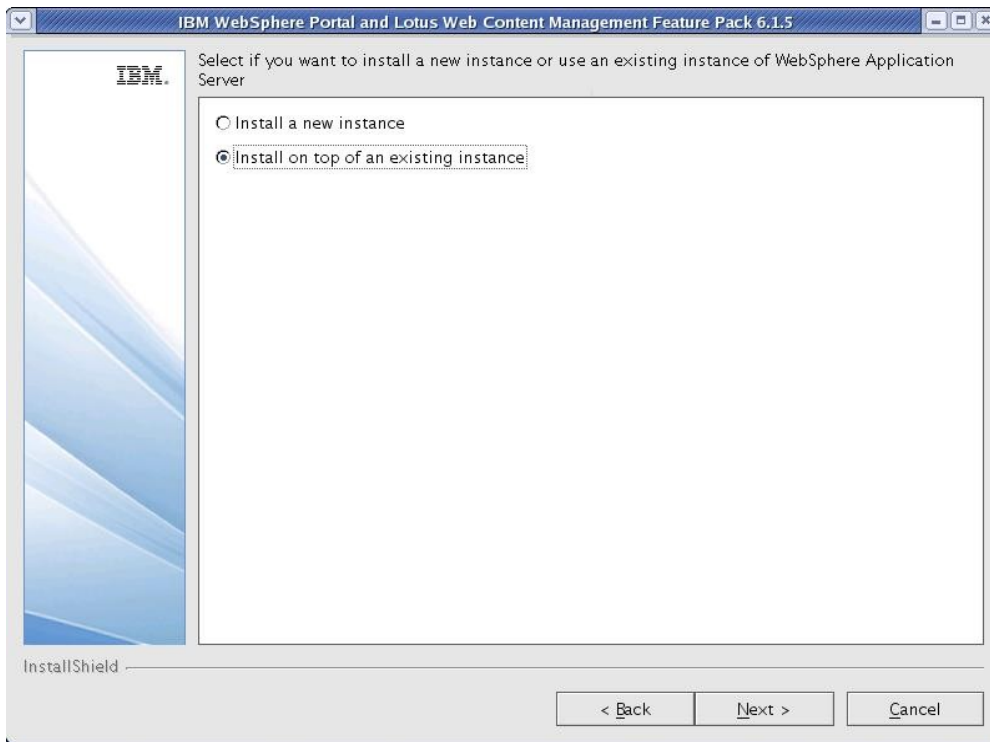


6. On the installation type screen, select 'Full' and click 'Next'

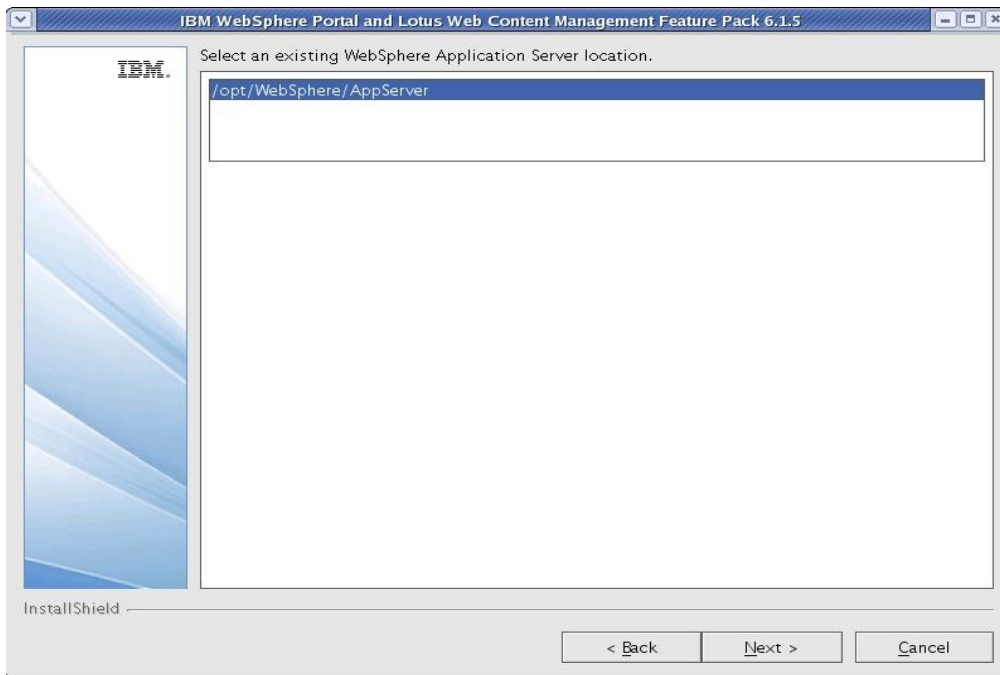
**NOTE:** Select Administration to install only administrative portlets.



7. Select to install to on top of an existing WAS installation:

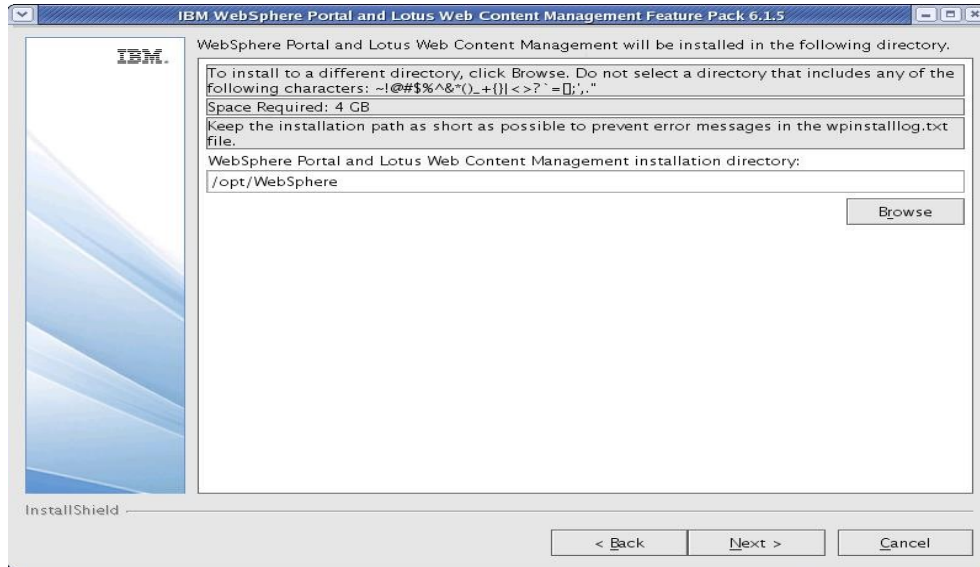


8. Select the path of the existing WAS installation:



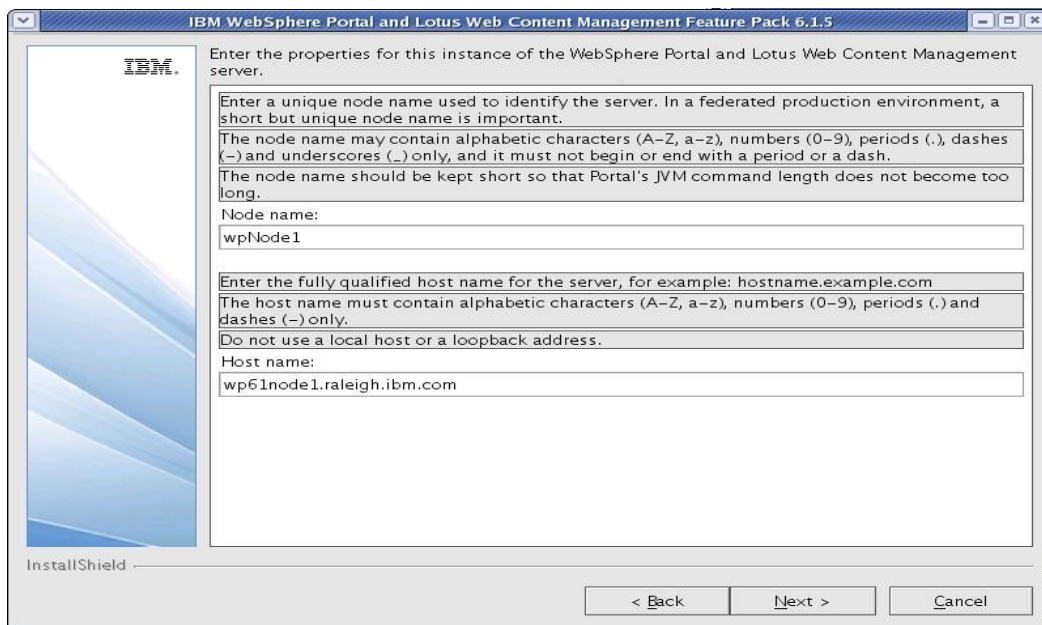
9. Select the desired path for the WebSphere directory and click 'Next'

**NOTE:** Both the profile directory and the PortalServer directory will be created in this WebSphere directory.

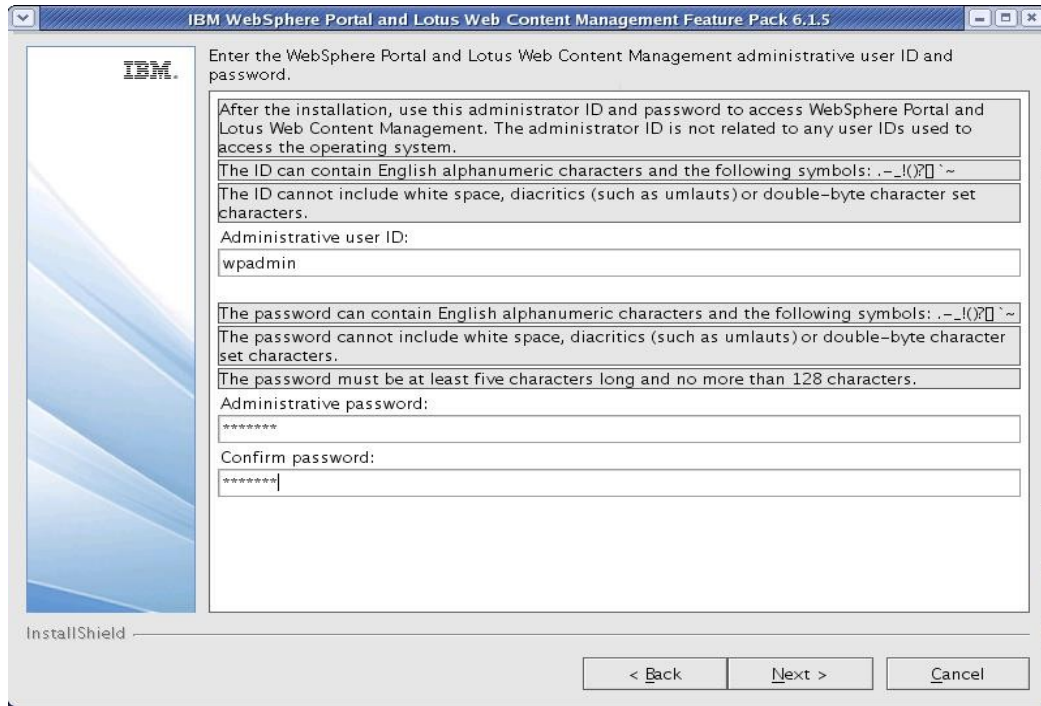


10. Enter a node name and the fully qualified hostname of your server and click 'Next'.

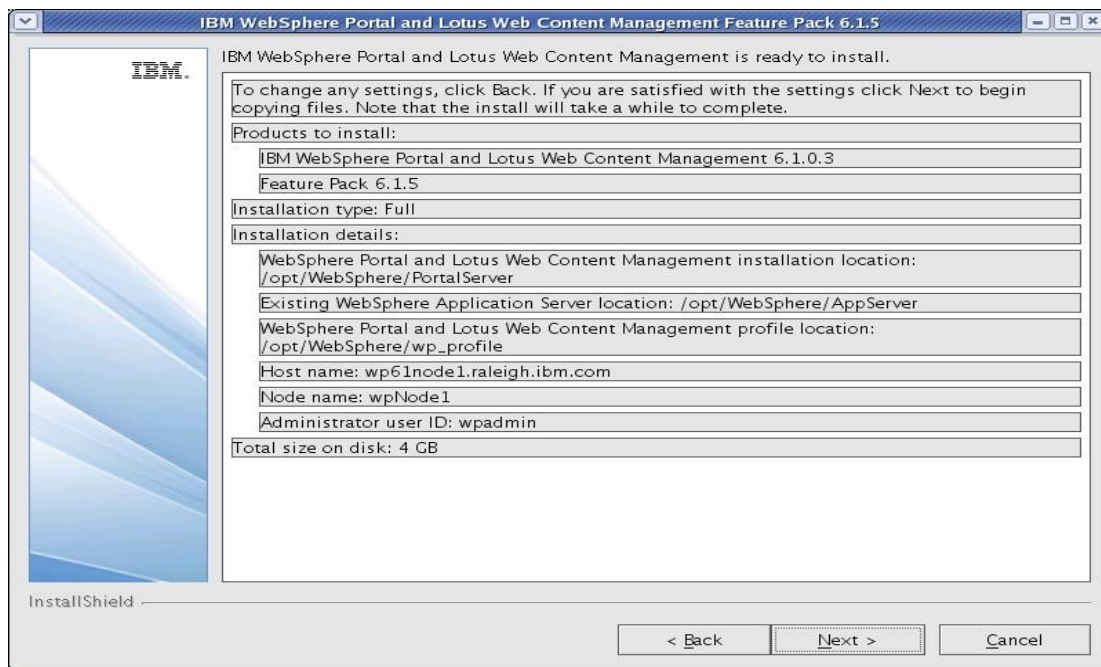
**NOTE:** The value for node name will also be used as the cell name in the standalone environment.



11. Security is enabled for Portal by default. Enter a user ID and password you wish to use. This ID will be used to access both server1 and WebSphere\_Portal after installation completes.



12. Verify the information is accurate in the summary screen and click 'Next' to begin the installation.



13. Once the installation finishes, uncheck Launch First Steps and Launch the Configuration Wizard. Click 'Finish'.

14. Verify you can access Portal in a web browser. The default URL is:

<http://yourserver.yourcompany.com:10039/wps/portal>

At this point you have successfully installed WebSphere Portal v6.1.0.3 on WebSphere Application Server v7.0.0.5.

## ***Install IBM Support Assistant Lite***

In this section, you will install IBM Support Assistant Lite for WebSphere Portal (ISALite). This step is optional but **highly** recommended. ISALite provides automatic log collection and symptom analysis support for WebSphere Portal problem determination scenarios. Installing this tool now can save you time in the future if you have any problems with WebSphere Portal that require you to contact support.

1. Visit the website below and download ISALite for WebSphere Portal v6.1 to a temporary directory:

<http://www-01.ibm.com/support/docview.wss?rs=688&uid=swg24008662>

2. Extract the downloaded zip file into the wp\_profile/PortalServer directory. This will create a directory called ISALite.
3. The tool is installed and ready for use. If you have an issue with WebSphere Portal and require remote technical support, instructions for using this tool can be found in Appendix D.

## ***Install the Deployment Manager***

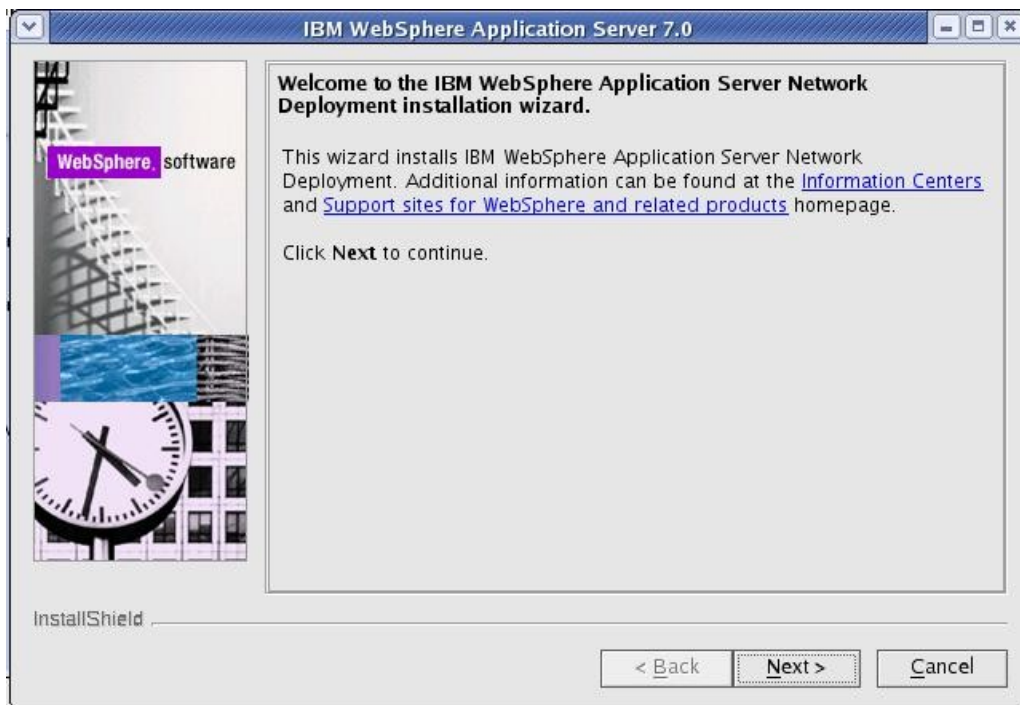
In this section, you will install the Deployment Manager. All of the following steps will be completed on the server you intend to use as your deployment manager.

Alternatively, you can use the existing WAS v7 installation to create a Deployment Manager profile on the same server as your primary Portal node. If you would like to do that instead, please follow Appendix A, then return to the 'Configure the Deployment Manager' section.

1. From the WAS v7 installation CD or image, launch the installer from the WAS directory:

`./install`

2. Click 'Next' on the Welcome Screen:

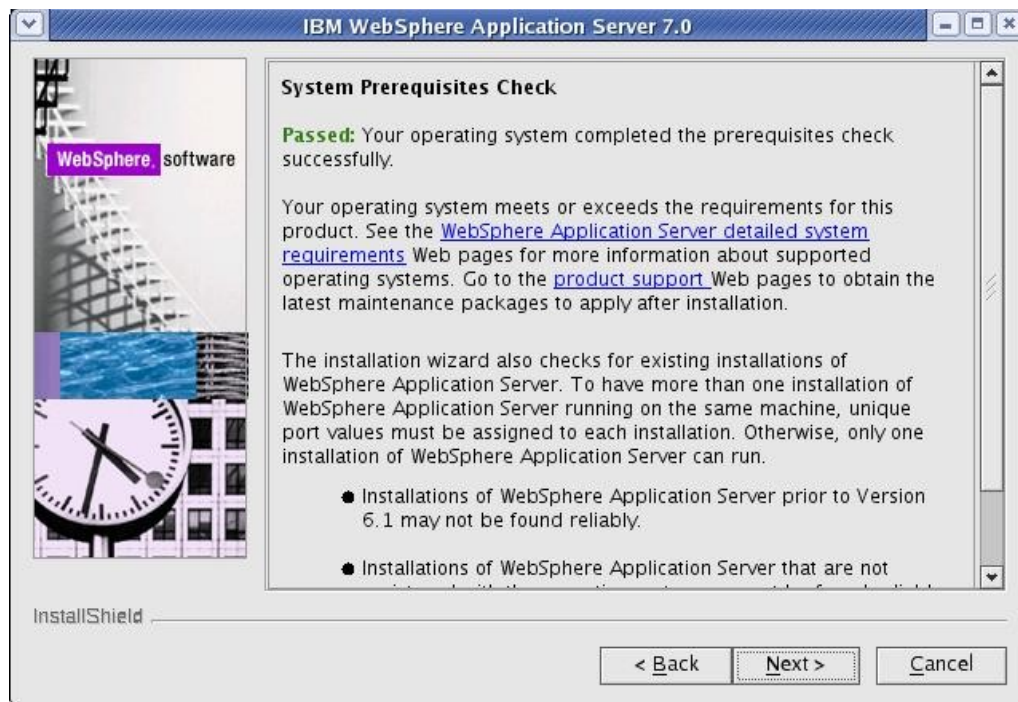




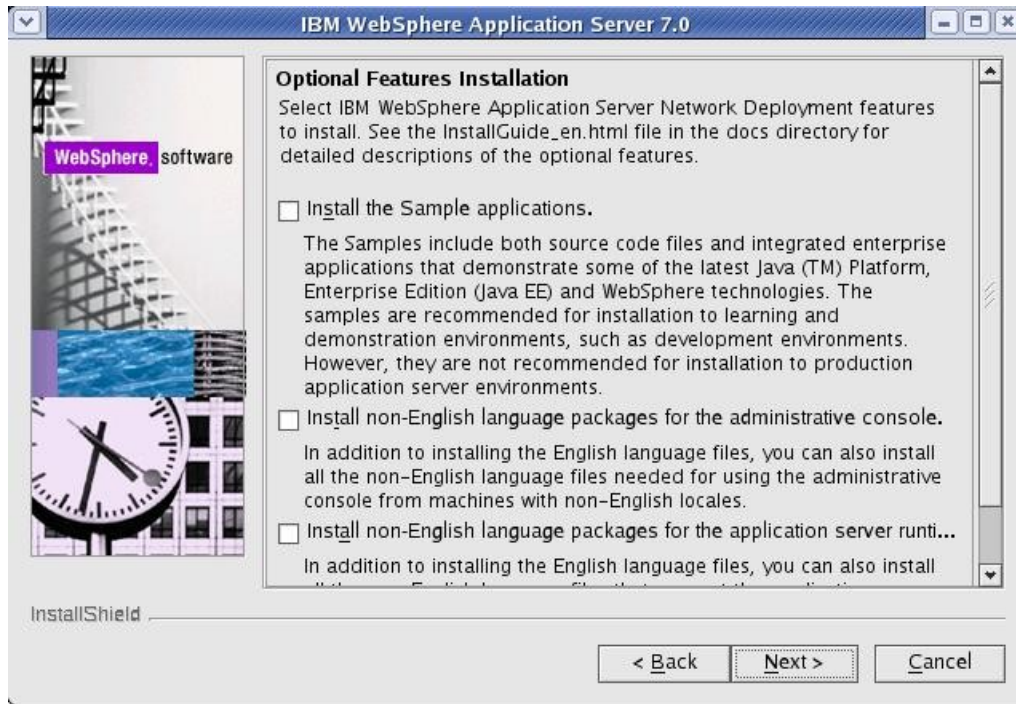
3. Accept the license and click 'Next':



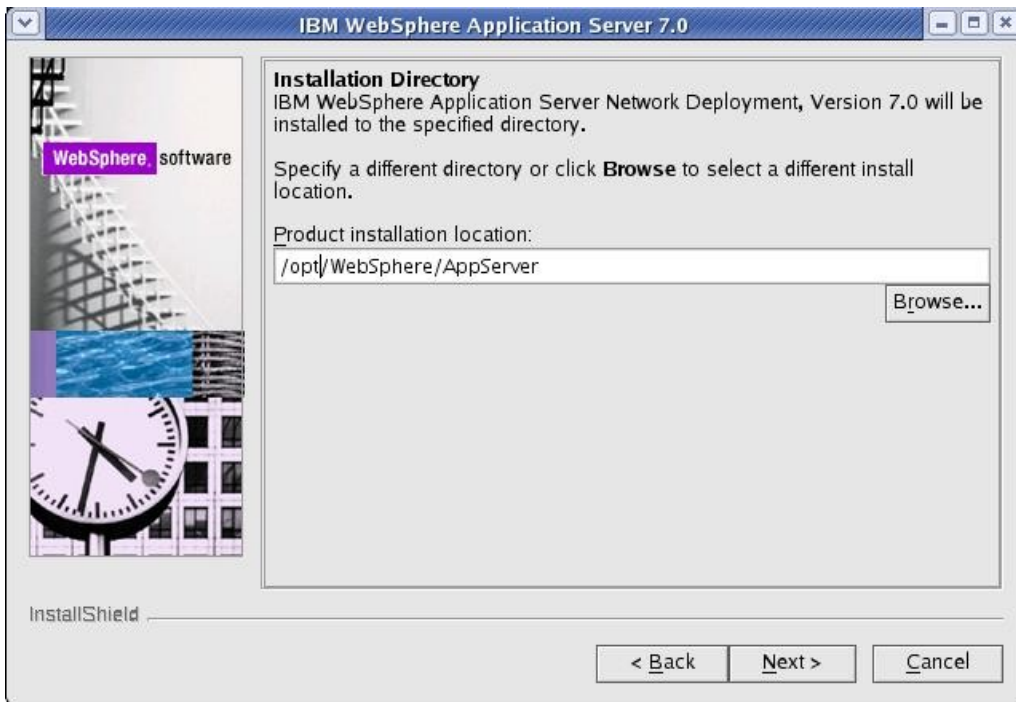
4. Click 'Next' on the Systems Prerequisite Check screen:



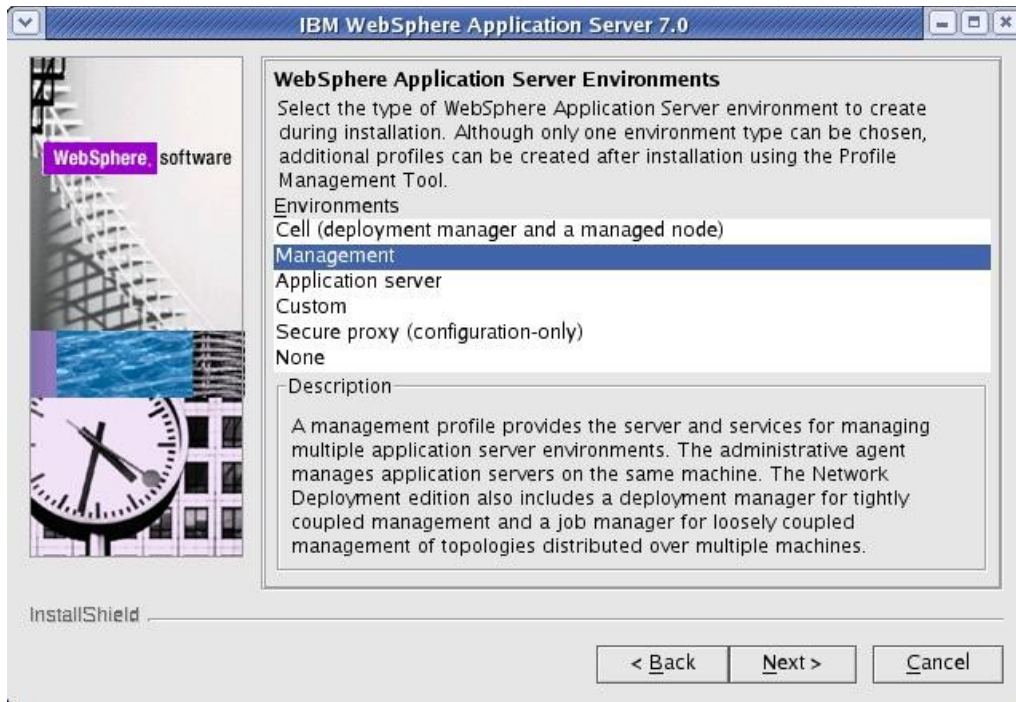
5. **Do not** select any options, click 'Next'.



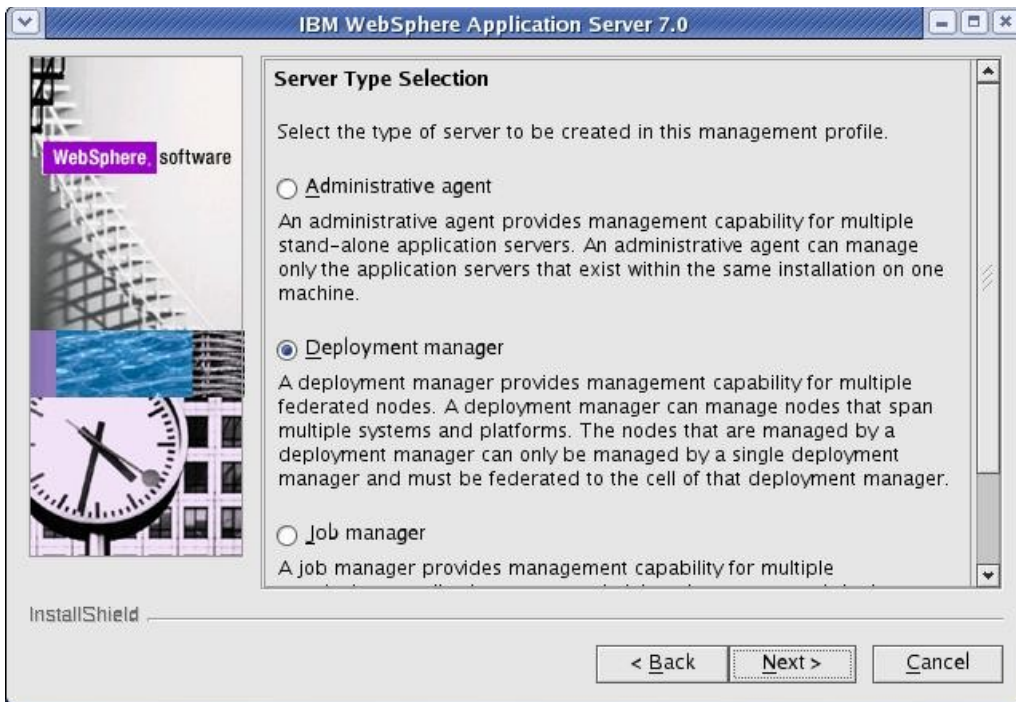
6. Select your installation directory and click 'Next':



7. Select to create a **Management** profile.



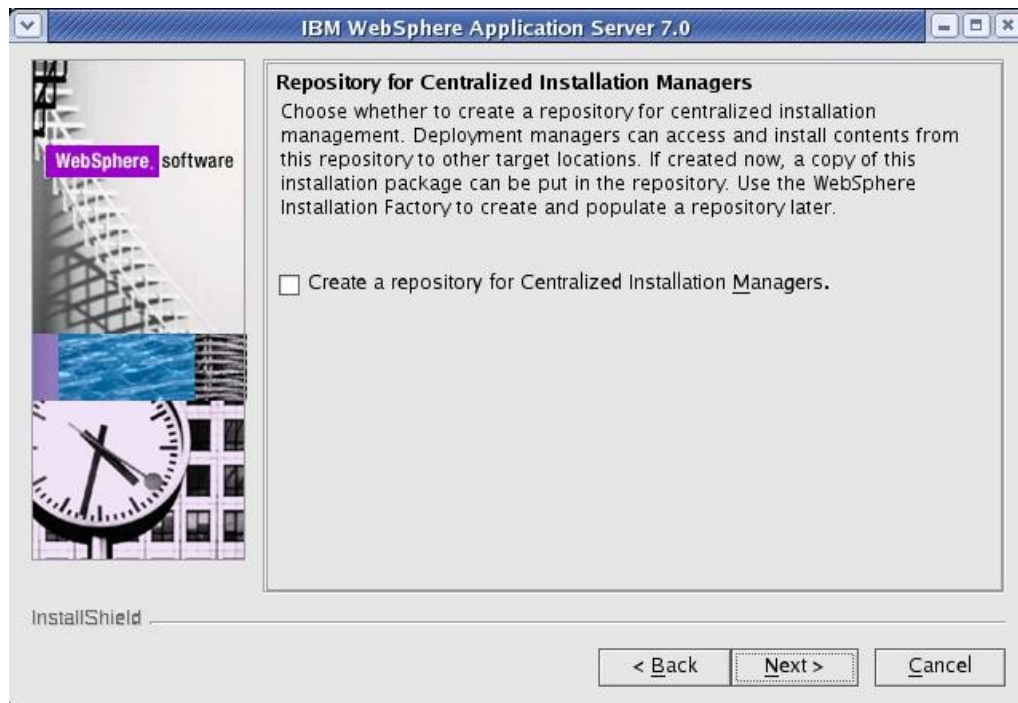
8. For Server Type, select Deployment Manager:



9. For the 'Enable Administrative Security' screen, **check** the box to enable security and provide a user ID and Password. **Use the same user ID and password you specified for the WebSphere Portal installation.**



10. Check the option to create a repository for Centralized Installation Managers if you'd like and click Next:



11. Review the information on the summary screen and click 'Next' to begin the installation.

12. After the installation completes, click 'Finish' to exit the installation program.

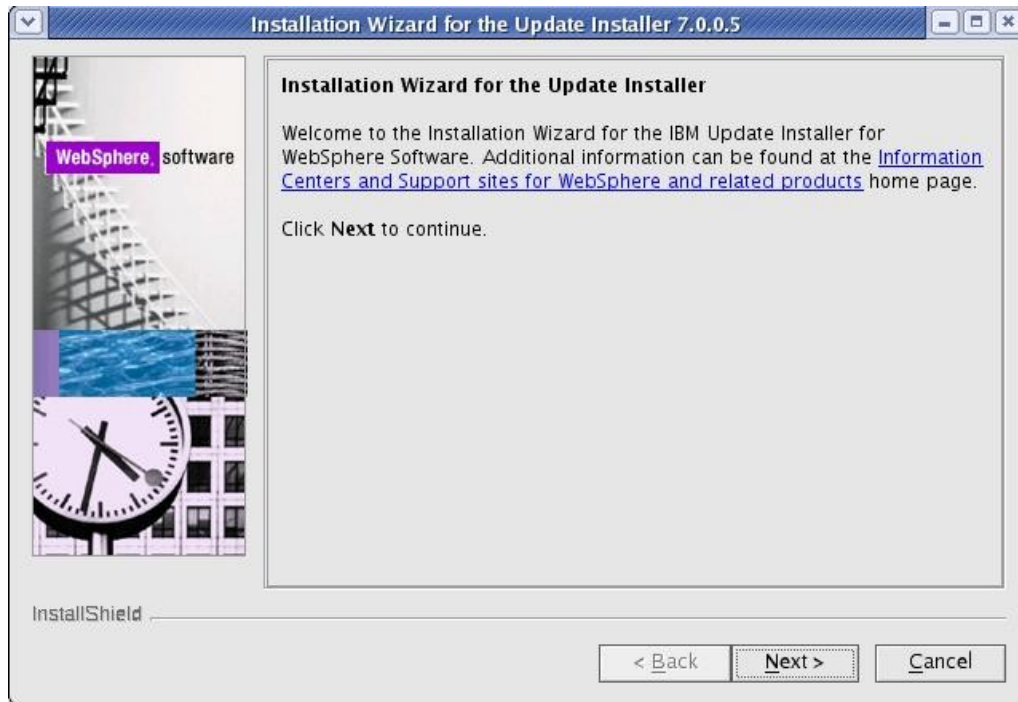
13. Download the WebSphere Application Server v7 Update Installer:

<http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg24020446>

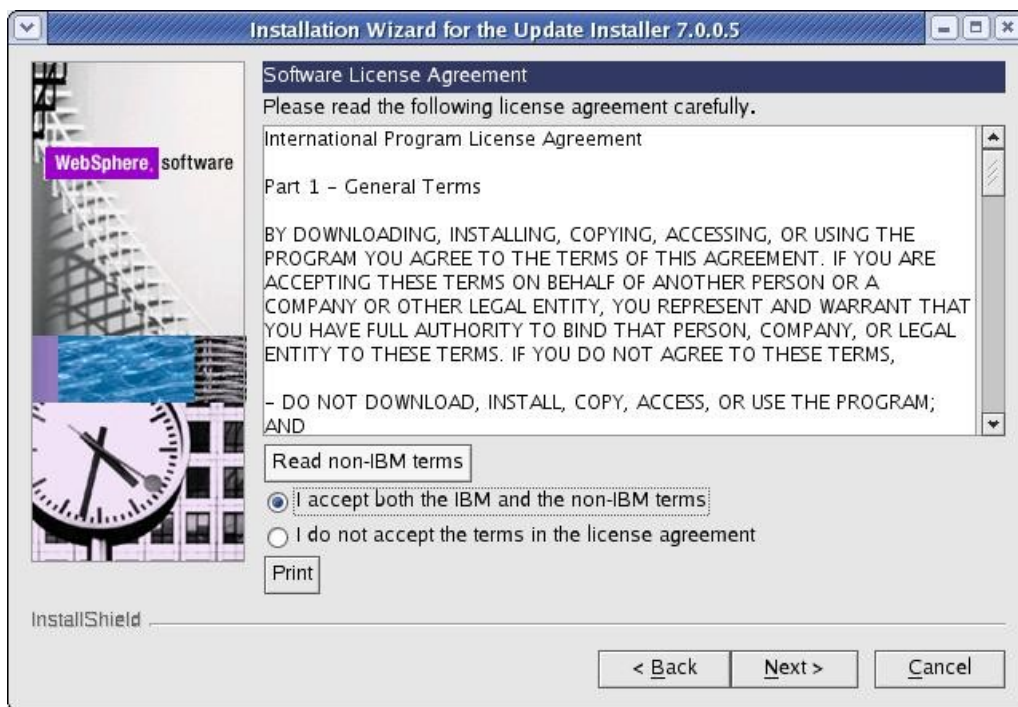
14. Extract the download into a temporary directory and launch the installer located in the <temp location>/UpdateInstaller directory:

`./install`

15. Click 'Next' on the Welcome Screen:

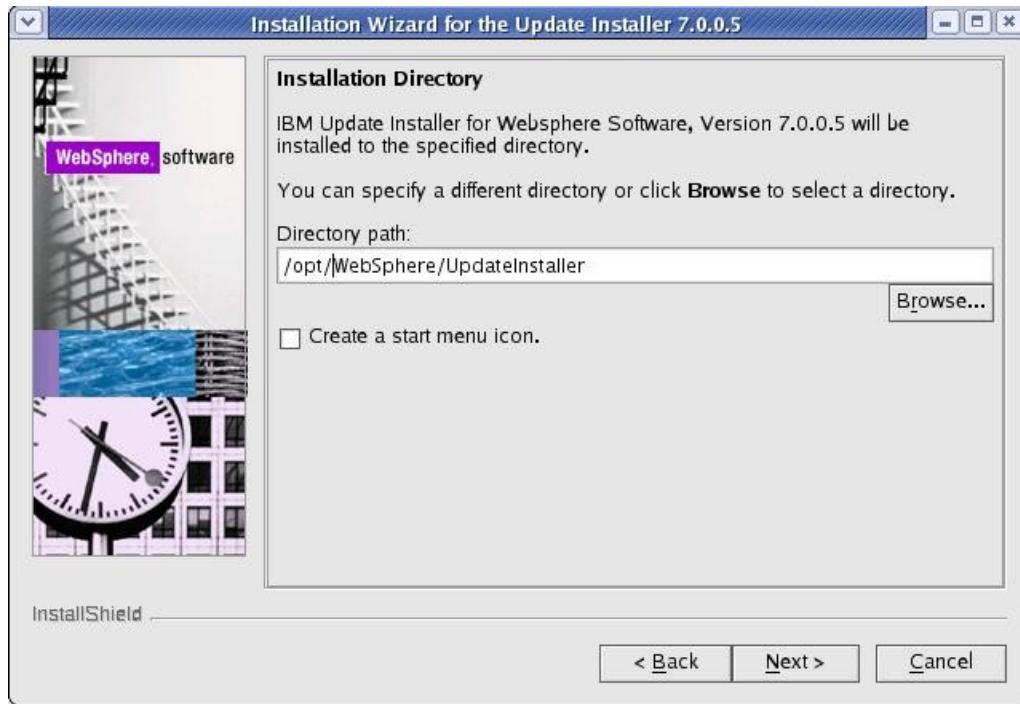


16. Accept the license and click 'Next':



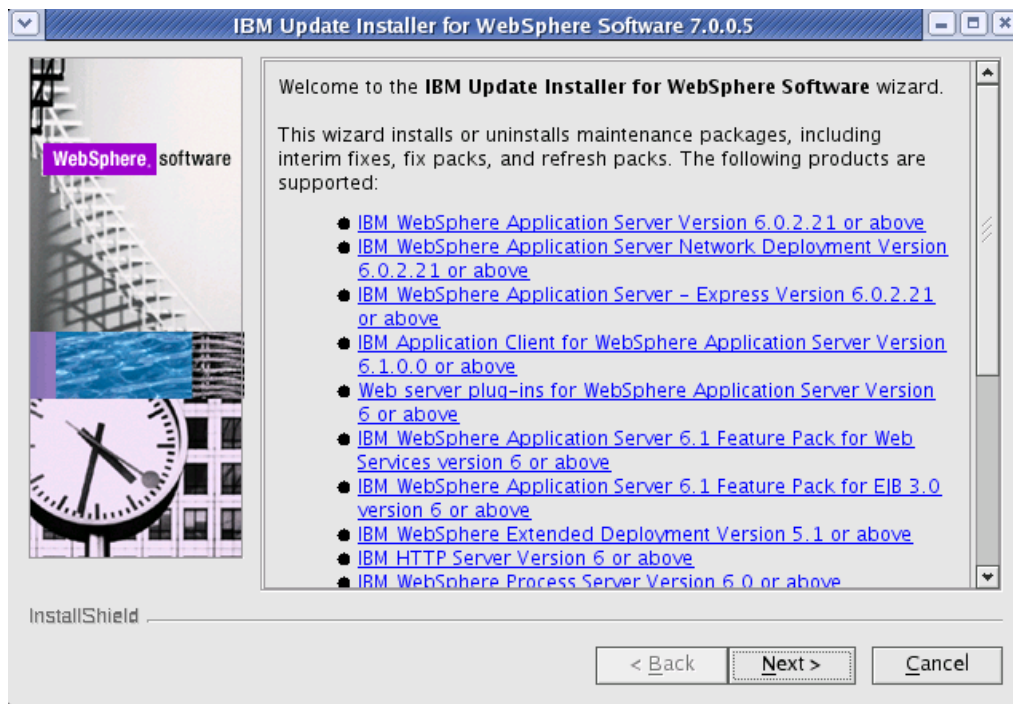
17. Click 'Next' on the System Pre-requisite check screen.

18. Select the path where you would like to install the WAS Update Installer:

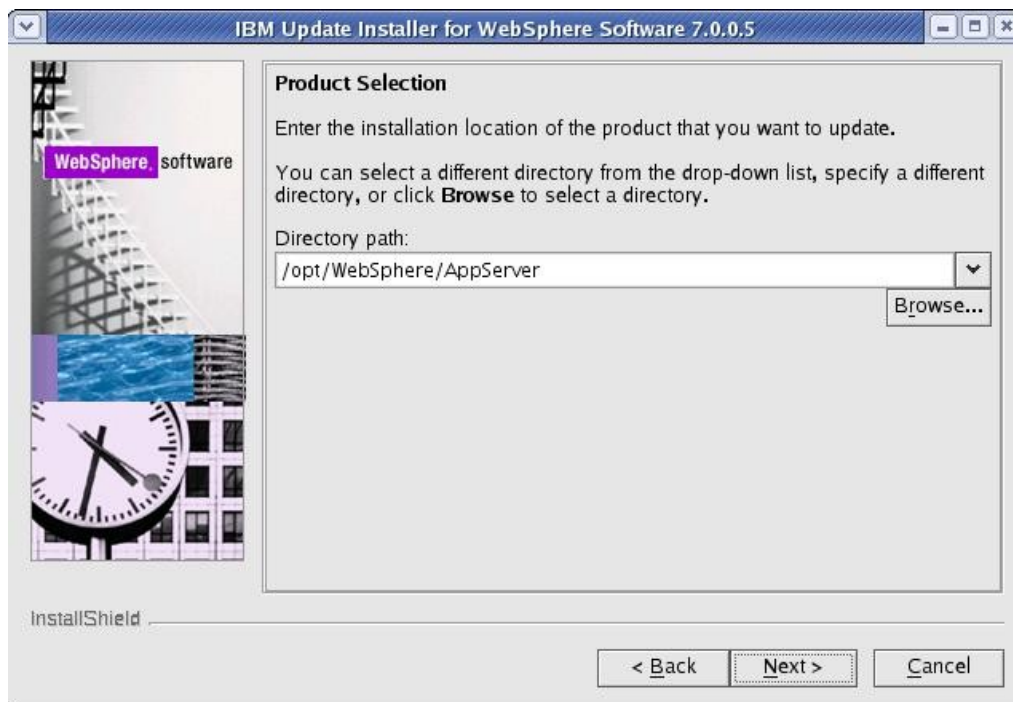


19. After the installation completes, click Finish to exit the installer.
20. Download the WAS 7.0.0.5 fixpack and the corresponding JDK upgrade:  
<http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg24023705>  
<http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg24023708>
21. Copy the 7.0.0.5 fixpack and the JDK upgrade to the <UpdateInstaller root>/maintenance directory, where <UpdateInstaller root> is the location you selected in Step 18.
22. Launch the WAS Update Installer from the UpdateInstaller directory you set from step 18:  
`./update.sh`

23. Click 'Next' on the Welcome Screen:

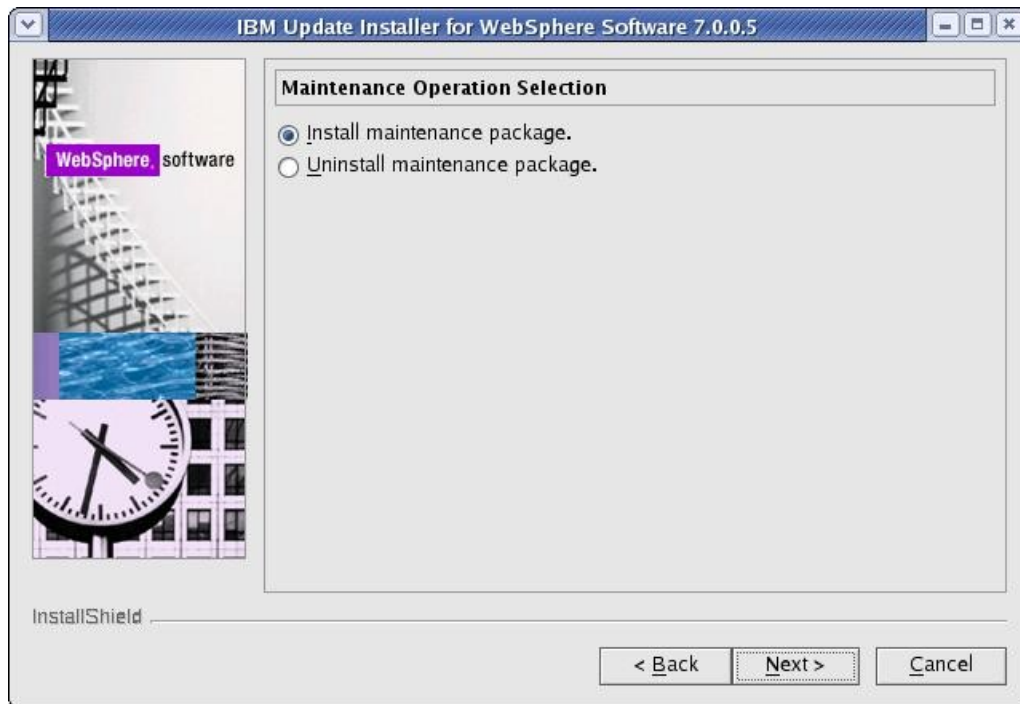


24. Select the WebSphere Application Server directory you wish to upgrade and click 'Next':

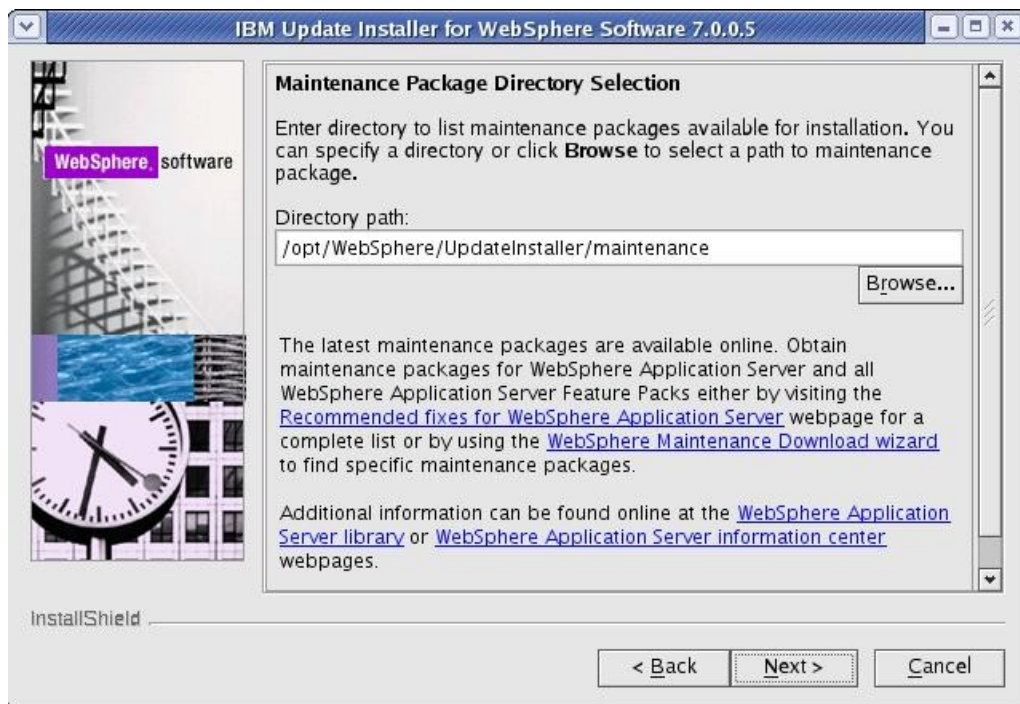




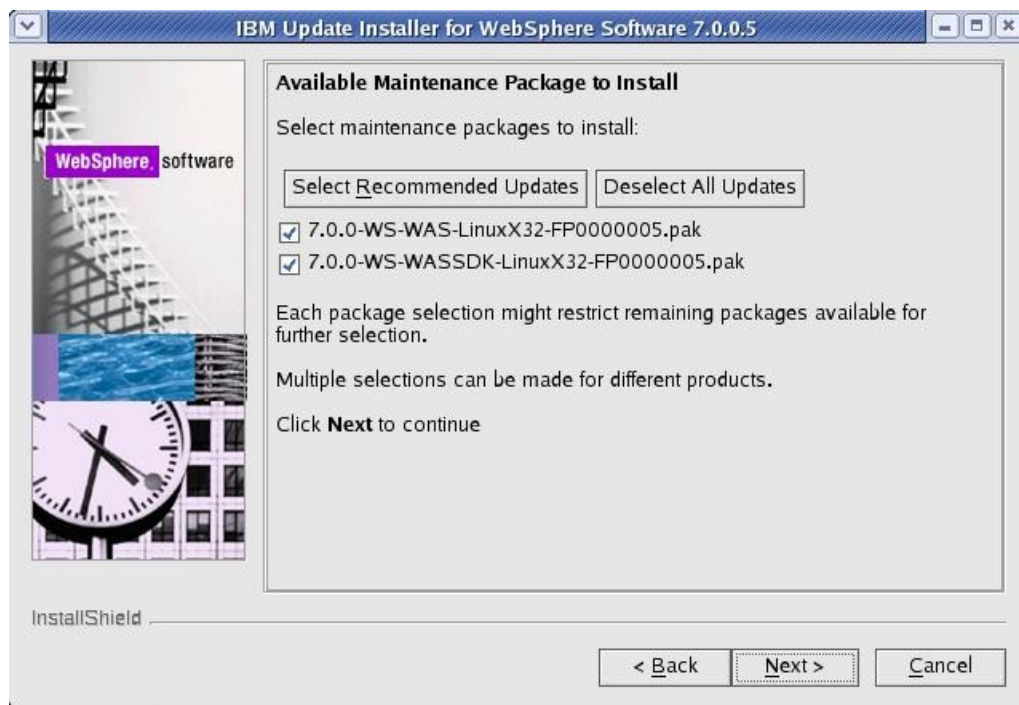
25. Select the 'Install Maintenance package' option and click 'Next':



26. Select the directory that contains the 7.0.0.5 and JDK packages:



27. Check the boxes for the 7.0.0.5 and JDK packages and click 'Next':



28. On the installation summary screen, click 'Next' to begin the upgrade.

29. After the upgrade completes, click 'Finish' to exit the update installer.

30. Download the required WebSphere Application Server interim fixes for WAS v7.0.0.5 when using WebSphere Portal v6.1.0.3/6.1.5 from the WebSphere Portal Support site:

<http://www-01.ibm.com/support/docview.wss?rs=688&uid=swg24023148>

31. Extract the zip file into a temporary directory. This should include the following interim fixes:

PK90343  
PK91698  
PK92047  
PK93952  
PK96275  
PK97321  
PK98302  
PK98741  
PK99787  
PM00692

32. Copy the fixes into the <UpdateInstaller root>/maintenance directory where <UpdateInstaller root> is the location you selected in step 18.

33. Repeat steps 22-29 to install the interim fixes. (Note that for step 27 you will select to install the interim fixes listed in Step 31, not the 7005 and JDK fixpacks).

At this point, the Deployment Manager has been installed and the DMGR profile has been created.

## **Configure the Deployment Manager**

In this section, you will configure the Deployment Manager and prepare it for the future Portal cluster. All of the following steps will be completed on the server you intend to use as your deployment manager.

1. From a command window, navigate to <dmgr\_profile>/bin
2. Execute the following command to start the Deployment Manager:

```
./startManager.sh
```

3. Once the DMGR is open for e-business, launch a web browser and access the DMGR Administrative Console:

```
http://<yourhostname>:9060/ibm/console
```

**NOTE:** The default port is 9060.

4. Enter the User ID and Password you used during the Deployment Manager installation and click 'Log in'
5. Increase the HTTP Connection timeouts for the deployment manager:
  - a) Navigate to **System Administration -> Deployment Manager -> Web Container Transport Chains**
  - b) For each entry in the table (WCInboundAdmin and WCInboundAdminSecure), complete the following:
    1. Click HTTP Inbound Channel
    2. Change Read Timeout to 180
    3. Change Write Timeout to 180
    4. Click OK
    5. Save configuration changes

6. Change the timeout request period for the Java Management Extensions (JMX) connector.
  - a) Navigate to **System Administration -> Deployment Manager -> Administration Services -> JMX connectors -> SOAPConnector -> Custom Properties**
  - b) Select the requestTimeout property, and increase the value from 600 to 6000.
  - c) Save configuration changes.

7. Update the maximum Java heap size used by the deployment manager:
  - a) Click System administration > Deployment manager > Java and Process Management > Process Definition > Java Virtual Machine.
  - b) Specify **256** for **Initial Heap Size** and **1024** for **Maximum Heap Size**.

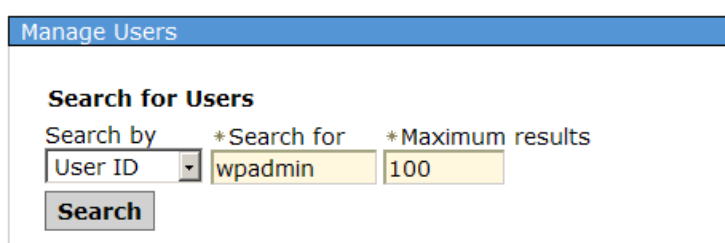
For information about appropriate heap sizes see the documentation for your operating system and the Performance Guides located on the [WebSphere Portal and Web Content Management Product Documentation](#) page.

**NOTE:** If using a 32-bit operating system, you will need to set the heap size to a lower size than a 64-bit operating system.

- c) Click OK, and then save your changes.
8. One significant change to the way clusters are created in Portal v6.1 is the security configuration. The node will inherit the security settings of the DMGR when it is federated. If you have been following the recommendations in this guide, then you already have the Portal Administrative User ID created in the Deployment Manager configuration (wpsadmin). We will need to create a Portal Administrative Group in the current DMGR security configuration and add the Portal Administrative User to it. These security settings will ultimately be used once our Portal node is federated.

Navigate to **Users and Groups -> Manage Groups**

9. Click Create
10. Create a group called **wpsadmins**. Do not use a different group name.
11. Navigate to **Users and Groups -> Manage Users**
12. Search for the user you created during DMGR profile creation.



The screenshot shows a web interface titled "Manage Users". Under the heading "Search for Users", there are three input fields: "Search by" with a dropdown menu set to "User ID", "\* Search for" with the text "wpsadmin", and "\* Maximum results" with the number "100". A "Search" button is located below these fields.

13. Add the user to the wpsadmins group:

a) Click the blue link for the user ID 'wpsadmin' in the table

1 users matched the search criteria.

Select	User ID	First name	Last name	E-mail	Unique Name
<input checked="" type="checkbox"/>	<a href="#">wpsadmin</a>	wpsadmin	wpsadmin		uid=wpsadmin,o=defaultWIMFileBasedRealm

b) On the next screen, click the Groups tab

Manage Users

**User Properties**

General **Groups**

\*User ID  
wpsadmin

c) Click the Add button

User ID  
wpsadmin

The user is a member of 0 groups.

**Add...** **Remove**

- d) Search for the group name you previously created in Step 10. Highlight the group 'wpsadmins' and click 'Add'

The screenshot shows a web interface titled "Manage Users". Under the heading "Add a User to Groups", there is a "User ID" field containing the text "wpsadmin". Below this, a prompt asks to "Specify the search criteria that you want to use to find the g". There are three input fields: "Search by" with a dropdown menu set to "Group name", "\*Search for" containing "wpsadmins", and "\*Maximum results" containing "100". A "Search" button is located below these fields. The search results section displays "1 groups matched the search criteria." and a list containing the entry "wpsadmins". At the bottom of the interface are "Add" and "Close" buttons.

- e) After the user is successfully added, click the 'Close' button

14. Logout of the Deployment Manager Admin Console and close the browser.
15. Edit the soap.client.props file from the <dmgr\_profile>/properties directory in a text editor.
16. Change the com.ibm.SOAP.timeout entry to 6000:

```
com.ibm.SOAP.requestTimeout=6000
```

17. Save the file

18. Restart the DMGR by issuing the following commands in a terminal window from the <dmgr\_profile>/bin directory:

```
./stopManager.sh -user wpaadmin -password wpaadmin
```

```
./startManager.sh
```

At this point, you have successfully prepared the Deployment Manager profile for Portal federation.



## **Configure the Primary Portal node to an external database**

In this section, Portal will be configured to use an external database. For the purposes of this document, DB2 will be used as the external database with Type 4 drivers. This may vary in your environment. For more information about other databases that can be used with Portal, please visit the WebSphere Portal v6.1.0 Information Center for configuring external databases at this link and follow the instructions there as appropriate:

[http://publib.boulder.ibm.com/infocenter/wpdoc/v6r1/index.jsp?topic=/com.ibm.wp.ent.doc\\_v615/config/linux\\_remote\\_db.html](http://publib.boulder.ibm.com/infocenter/wpdoc/v6r1/index.jsp?topic=/com.ibm.wp.ent.doc_v615/config/linux_remote_db.html)

In the environment used for this guide, 6 databases were created following the instructions in the Information Center:

RELDB  
COMDB  
CUSDB  
JCRDB  
FDBKDB  
LMDB

In addition, the database administrator user “db2inst1” will be used as the user ID for each database.

If you choose to use DB2, the contents of the SQL file used to create and prepare the databases is included in **Appendix B**.

1. From the primary Portal node, ensure the WebSphere\_Portal and server1 servers are stopped by executing the following commands from the terminal window in the <wp\_profile>/bin directory:  

```
./stopServer.sh WebSphere_Portal -user <admin user> -password <admin pwd>  
./stopServer.sh server1 -user <admin user> -password <admin pwd>
```
2. Ensure the database client is installed and configured on the node. Since we are using Type 4 drivers for DB2, all that is needed is to copy the db2jcc.jar and db2jcc\_license\_cu.jar files from the DB2 server to some directory on the primary Portal server.
3. Ensure the remote DB2 server is started.

4. From the <wp\_profile>/ConfigEngine/properties directory, make a backup of the following files:

```
wkplc.properties  
wkplc_dbtype.properties  
wkplc_comp.properties
```

5. Edit the wkplc\_dbtype.properties file and make the following changes:

```
db2.DbDriver=com.ibm.db2.jcc.DB2Driver  
db2.DbLibrary=/opt/ibm/db2/V9.1/java/db2jcc.jar:/opt/ibm/db2/V9.1/java/db2jcc  
_license_cu.jar  
db2.JdbcProviderName=wpdbJDBC_db2
```

**NOTE:** The entry for db2.DbLibrary is an example only. Please ensure this is a valid path on your system.

6. Edit the wkplc\_comp.properties file and make the following changes:

```
feedback.DbType=db2  
feedback.DbName=fdbkdb  
feedback.DbSchema=FEEDBACK  
feedback.DataSourceName=wpdbDS_fdbk  
feedback.DbUrl=jdbc:db2://mydbserver.ibm.com:50000/fdbkdb:returnAlias=0;  
feedback.DbUser=db2inst1  
feedback.DbPassword=password
```

```
likeminds.DbType=db2  
likeminds.DbName=lmdb  
likeminds.DbSchema=likeminds  
likeminds.DataSourceName=wpdbDS_lmdb  
likeminds.DbUrl=jdbc:db2://mydbserver.ibm.com:50000/lmdb:returnAlias=0;  
likeminds.DbUser=db2inst1  
likeminds.DbPassword=password
```

```
release.DbType=db2  
release.DbName=reldb  
release.DbSchema=release  
release.DataSourceName=wpdbDS_reldb  
release.DbUrl=jdbc:db2://mydbserver.ibm.com:50000/reldb:returnAlias=0;  
release.DbUser=db2inst1  
release.DbPassword=password
```

```
community.DbType=db2  
community.DbName=comdb  
community.DbSchema=community  
community.DataSourceName=wpdbDS_comdb  
community.DbUrl=jdbc:db2://mydbserver.ibm.com:50000/comdb:returnAlias=0;  
community.DbUser=db2admin  
community.DbPassword=password
```

```
customization.DbType=db2
customization.DbName=cusdb
customization.DbSchema=customization
customization.DataSourceName=wpdbDS_cusdb
customization.DbUrl=jdbc:db2://mydbserver.ibm.com:50000/cusdb:returnAlias=0;
customization.DbUser=db2inst1
customization.DbPassword=password
```

```
jcr.DbType=db2
jcr.DbName=jcrdb
jcr.DbSchema=jcr
jcr.DataSourceName=wpdbDS_jcrdb
jcr.DbUrl=jdbc:db2://mydbserver.ibm.com:50000/jcrdb:returnAlias=0;
jcr.DbUser=db2inst1
jcr.DbPassword=password
```

7. Copy the following files from the WebSphere Portal server to a temporary directory on the DB2 server:

<PortalServer root>/jcr/prereq/jcr/config/collation.jar

<PortalServer root>/jcr/prereq/jcr/config/registerCollationUDFTemplate.sql

8. From the DB2 server, open a terminal window and change directories to:

<db2 instance home>/sqllib/function

9. From the DB2 server, Execute the following command:

```
jar -xvf <temporary location>/collation.jar
```

**NOTE:** In order for the above command to work, you must have java installed on the DB2 server and the JAVA\_HOME environment variable set to the java root directory.

10. From the DB2 server, edit the <temporary location>/registerCollationUDFTemplate.sql file in a text editor.

11. Change all *SCHEMA* references in this file to the value you set for jcr.DbSchema in wkplc\_dbdomain.properties. In this case, the schema value is 'jcr'.

12. Change the following line:

```
"VALUE VARCHAR(32672),"
```

to

```
"VALUE VARCHAR(100),"
```

13. Change the following line:

```
"RETURNS VARCHAR(32672) FOR BIT DATA"
```

to

```
"RETURNS VARCHAR(100) FOR BIT DATA"
```

14. Save the registerCollationUDFTemplate.sql file

15. From the DB2 server, connect to the JCR database by executing the following command in a terminal window:

```
db2 connect to jcrdb using db2inst1 using password
```

16. From the same terminal window, execute the SQL script by running the following command:

```
db2 -tvf <temporary location>/registerCollationUDFTemplate.sql
```

17. Disconnect from the JCRDB and restart the DB2 instance.

18. Switch over to the Primary Portal node, and from a terminal window, change directories to <wp\_profile root>/ConfigEngine

19. Execute the following ConfigEngine scripts to validate the database properties:

```
./ConfigEngine.sh validate-database-driver -DWasPassword=<password>  
./ConfigEngine.sh validate-database-connection -DWasPassword=<password>
```

20. Execute the following ConfigEngine script to transfer the database from Derby to DB2:

```
./ConfigEngine.sh database-transfer -DPortalAdminPwd=<password>  
-DWasPassword=<password>
```

21. After the database-transfer completes, change directories to <wp\_profile>/bin and execute the following command to start the Portal server:

```
./startServer.sh WebSphere_Portal
```

22. Verify that you can render Portal successfully in a web browser.

<http://myserver.mycompany.com:10039/wps/portal>

At this point, you have successfully installed WebSphere Portal and configured it to use an external database.

## **Federate and Cluster the Primary Node**

The next step is to federate and cluster the WebSphere Portal node. The clustering process has changed significantly from Portal v6.0 to v6.1. You are no longer required to manually execute the AddNode command to add the node to the Deployment Manager cell. Instead, a ConfigEngine script has been created that does this, among other things for you. After the following steps have been completed, you will have a one node cluster.

1. From the primary node, open a command window and change directories to the <wp\_profile>/ConfigEngine directory.
2. Collect files from the Portal node that will need to be added to the Deployment Manager file structure. To collect the files, execute the following ConfigEngine script:  

```
./ConfigEngine.sh collect-files-for-dmgr -DWasPassword=<password>
```

This will create a zip file called filesforDmgr.zip in the <wp\_profile root>/filesforDmgr directory.
3. Copy the filesforDmgr.zip file from your primary node to a temporary directory on your Deployment Manager server.
4. Extract the filesforDmgr.zip into a temporary directory on the Deployment Manager server.
5. **Remote DMGR only.** From the temporary directory on your Deployment Manager, copy the AppServer/lib/wkplc.comp.registry.jar and wp.wire.jar into your <AppServer root>/lib directory.
6. From the temporary directory on your Deployment Manager, copy the AppServer/plugins/com.ibm.ws.portletcontainer.deploytask\_6.1.0.jar and wp.base.jar file into your <AppServer root>/plugins directory.
7. From the temporary directory on your Deployment Manager, copy the AppServer/profiles/Dmgr01/config/.repository/metadata\_wkplc.xml file into your <dmgr profile>/config/.repository directory.
8. Stop the deployment manager by issuing the following command from the <dmgr profile>/bin directory:  

```
./stopManager.sh -user <admin user> -password <admin pwd>
```
9. Start the deployment manager by issuing the following command from the <dmgr profile root>/bin directory:  

```
./startManager.sh
```
10. Stop WebSphere\_Portal and server1 by executing the following commands from the <wp\_profile root>/bin directory:  

```
./stopServer.sh WebSphere_Portal -user <admin user> -password <admin pwd>  
./stopServer.sh server1 -user <admin user> -password <admin pwd>
```

11. On the primary node, edit the <wp\_profile>/ConfigEngine/properties/wkplc.properties file and ensure all of the following properties are set appropriately for your environment:

```
WasUserId=<DMGR admin user ID>
WasPassword=<DMGR admin password>
PortalAdminPwd=<password>
WasRemoteHostName=<fully qualified hostname of DMGR>
WasSoapPort=<soap port for DMGR; default is 8879>
ServerName=WebSphere_Portal
PrimaryNode=true
ClusterName=PortalCluster
```

**NOTE:** This guide was written specifically for Portal v6.1.0.3 and higher. If you are using WebSphere Portal v6.1.0.0, 6.1.0.1, or 6.1.0.2 then you **must** leave the `WasUserId` and `WasPassword` properties set to the **standalone** server admin values, NOT the DMGR values.

**NOTE:** For the primary node, you **must** leave `ServerName` as `WebSphere_Portal`. Do not change it to any other value.

12. Edit <wp\_profile>/ConfigEngine/properties/wkplc\_comp.properties and ensure all database user IDs and passwords are accurate.
13. Ensure that the operating system time on the Deployment Manager server and the time on the primary node are within 5 minutes of each other. This is necessary for Steps 14-15 to complete successfully.
14. In a terminal window from the primary node, change directories to <wp\_profile>/ConfigEngine

15. Add the node to the deployment manager cell by executing the following ConfigEngine script:

```
./ConfigEngine.sh cluster-node-config-pre-federation -DWasPassword=password
```

**NOTE:** If you are prompted to accept an SSL certificate, type Y and press Enter to continue

**NOTE:** This guide was written specifically for Portal v6.1.0.3 and higher. If you are using WebSphere Portal v6.1.0.0, 6.1.0.1, or 6.1.0.2 then you **must** specify -DDMgrUserid and -DDMgrPassword parameters when running the cluster-node-config-pre-federation task. For example:

```
./ConfigEngine.sh cluster-node-config-pre-federation -DDMgrUserid=<DMGRUser>  
-DDMgrPassword=<password>
```

**IMPORTANT:** If you receive a BUILD FAILED for the cluster-node-config-pre-federation script, you **MUST** do the following before running the script again:

1. Execute the following ConfigEngine script to clean up the WAS registry:

```
./ConfigEngine.sh -DWasRemoteHostName=<standalone hostname>  
-DWasSoapPort=<standalone soap port>
```

for example:

```
./ConfigEngine.sh -DWasRemoteHostName=localhost -DwasSoapPort=10033
```

If your WasUserId and WasPassword values are different for the DMGR and the standalone instance, then you must also add the following to the ConfigEngine command: -DWasUserId=<standalone ID> -DWasPassword=<standalone password>

2. Remove the node in case AddNode portion of the script went through successfully
3. Login to the DMGR and do the following (these may not exist, depending on where the failure occurred):
  - a) Remove all Enterprise applications
  - b) Remove the WebSphere\_Portal server definition
  - c) Remove the JDBC Provider information for WebSphere\_Portal



16. After the previous step completes, your node will be part of the deployment manager cell. The node is now using the Deployment Manager security configuration and cell name. The original WAS ID that had been used in the standalone environment will no longer be used.

Edit the `<wp_profile>/ConfigEngine/properties/wkplc.properties` file and ensure the following properties are all set correctly:

```
WasUserId=<dmgr admin user id>
WasPassword=<dmgr password>
CellName=<dmgr cell name>
```

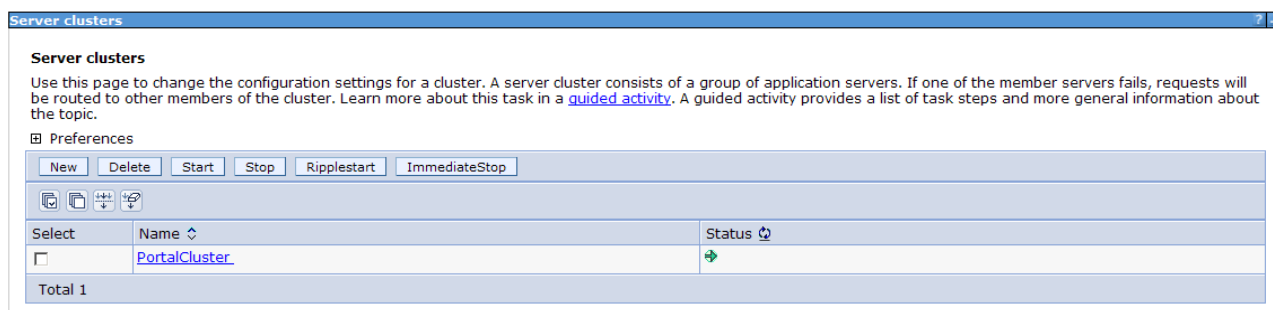
17. Update the deployment manager configuration for the new WebSphere Portal server by executing the following ConfigEngine script:

```
./ConfigEngine.sh cluster-node-config-post-federation
-DWasPassword=<password>
```

18. Create the cluster definition and add the WebSphere\_Portal server as a cluster member by executing the following ConfigEngine script:

```
./ConfigEngine.sh cluster-node-config-cluster-setup -DWasPassword=<password>
```

19. Ensure that the cluster definition was created correctly by logging into the DMGR Admin Console and browse to Server -> Clusters -> WebSphere Application Server Clusters. An entry for your Portal cluster should be present.



20. Verify Portal is functional by accessing it in your web browser:

<http://myserver.mycompany.com:10039/wps/portal>

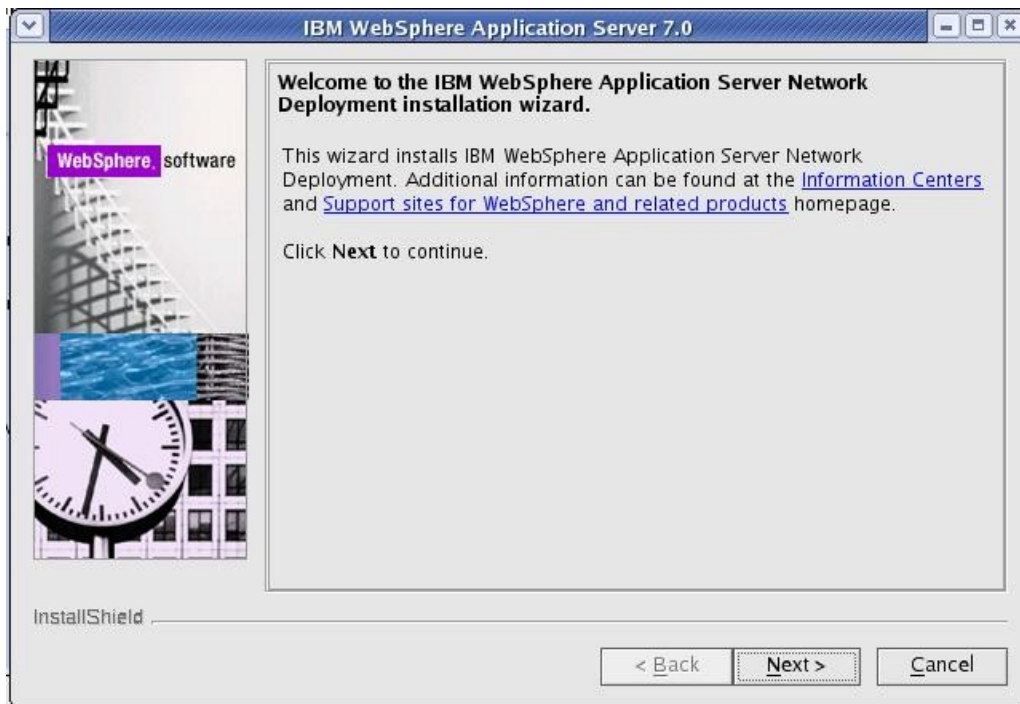
## ***Install WebSphere Application Server v7 on the future Portal Secondary Node***

In this section, you will install WebSphere Application Server v7.0.0.0 on the future Portal secondary Node, and upgrade it to v7.0.0.5. WebSphere Application Server v7 is NOT provided with the WebSphere Portal v6.1.5 bundle so you must obtain the installation media and license elsewhere.

1. From the WAS v7 installation CD or image, launch the installer from the WAS directory:

`./install`

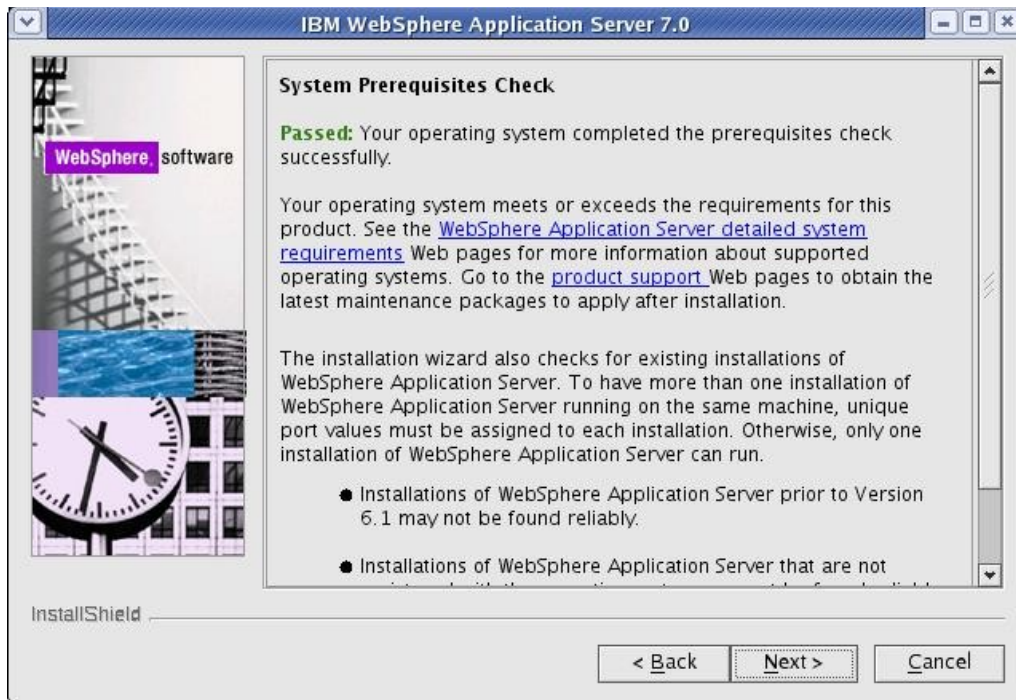
2. Click 'Next' on the Welcome Screen:



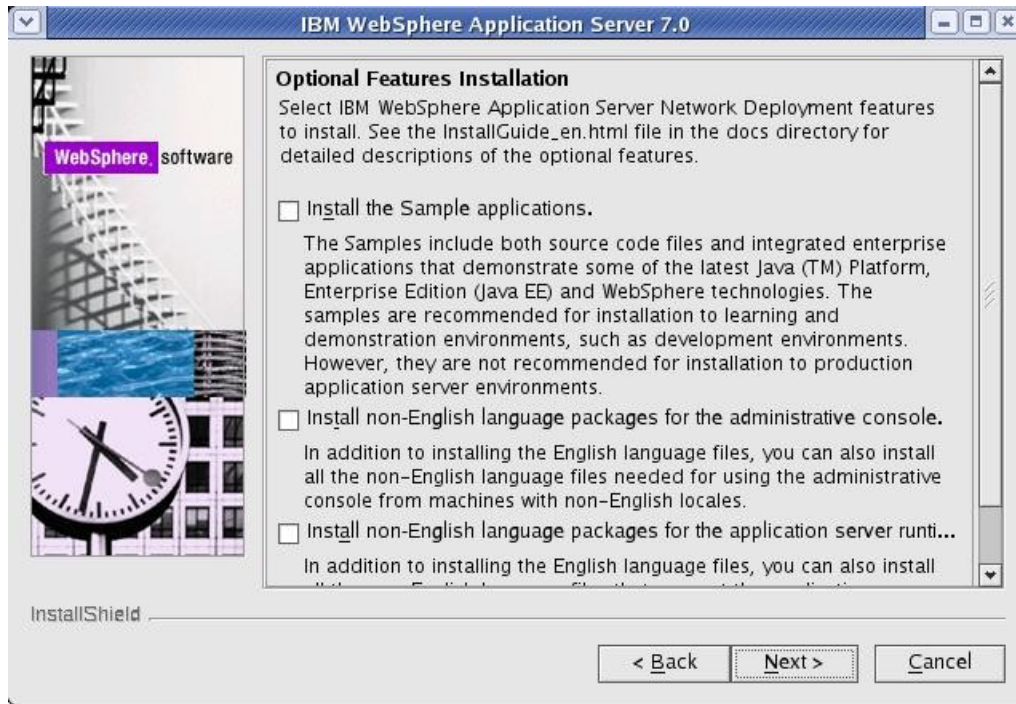
3. Accept the license and click 'Next':



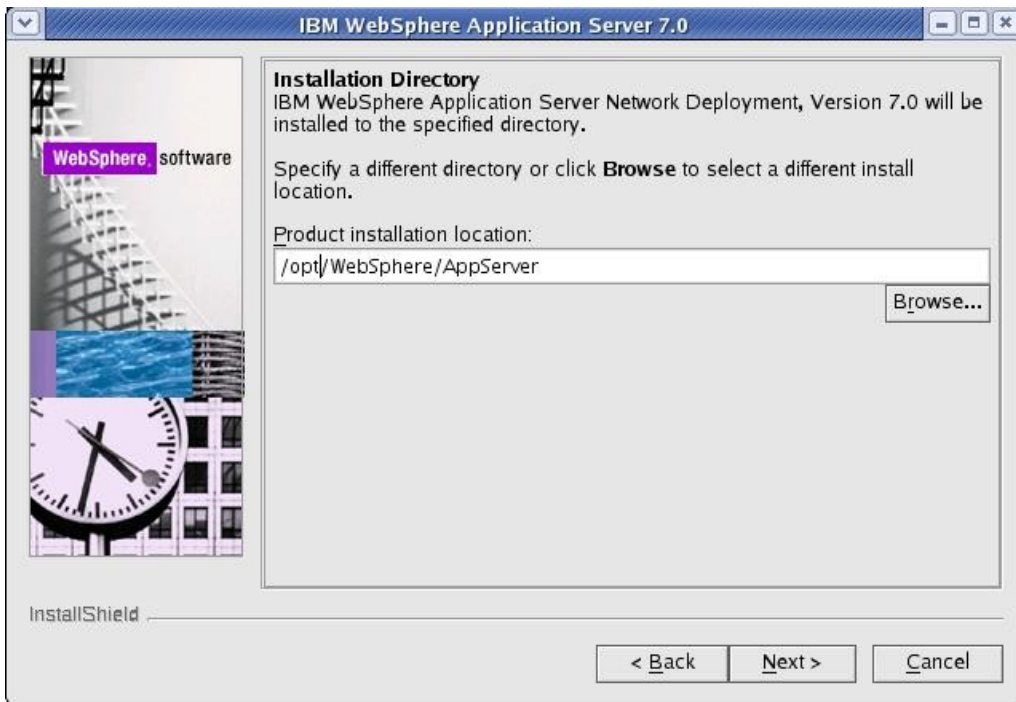
4. Click 'Next' on the Systems Prerequisite Check screen:



5. **Do not** select any options, click 'Next'.

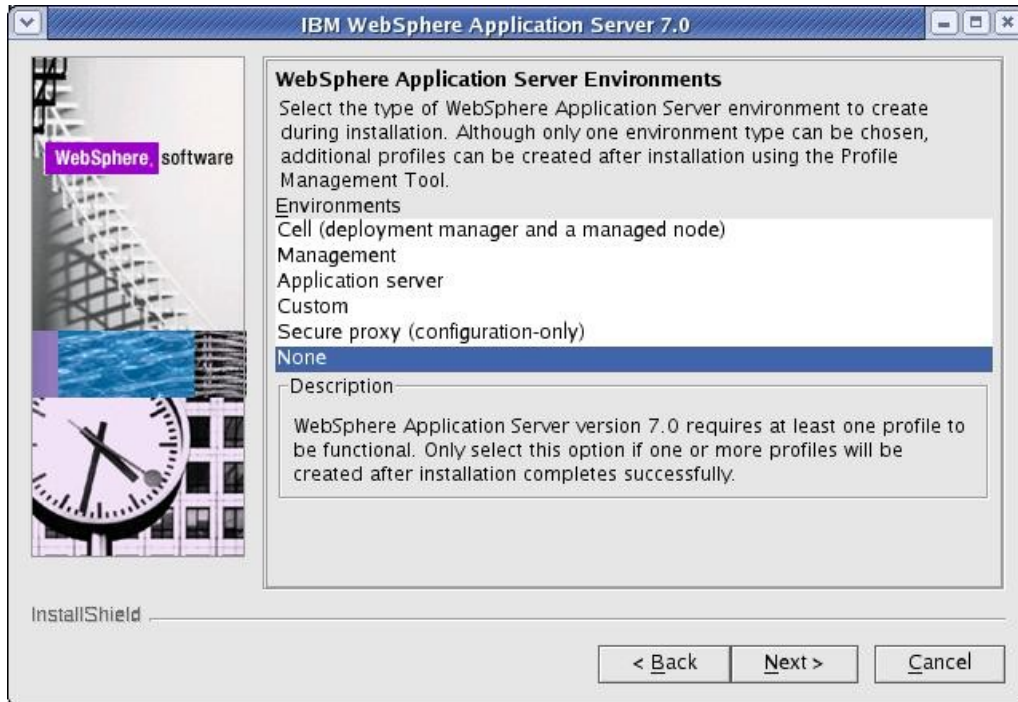


6. Select your installation directory and click 'Next':



7. **Do not** select to create a profile.

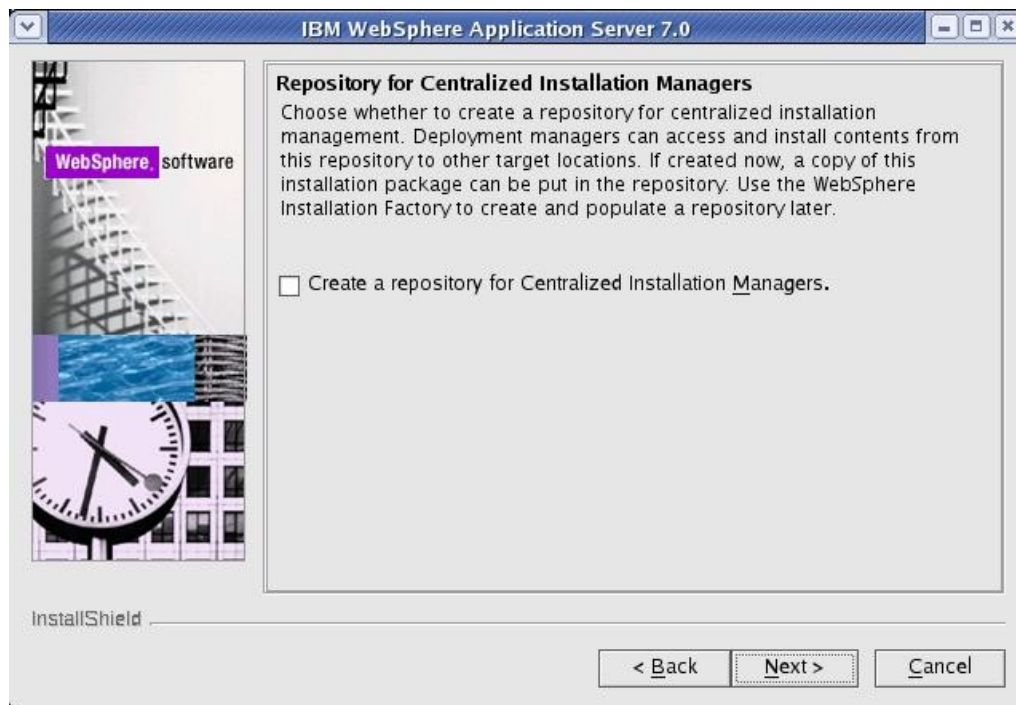
**Note:** The WebSphere Portal installer will create its own WAS profile so there is no need to create a profile here. If you do create a profile, WebSphere Portal will not use it.



8. Click 'Yes' on the warning that pops up when you select no profile:



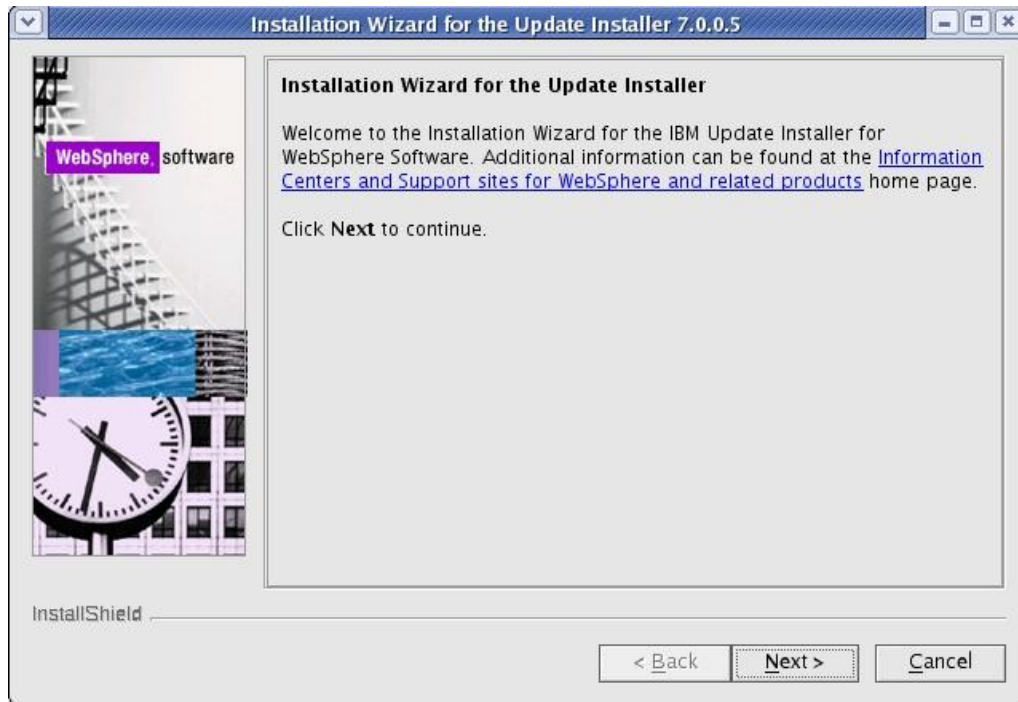
9. Check the option to create a repository for Centralized Installation Managers if you'd like and click Next. In this guide, the option is not checked:



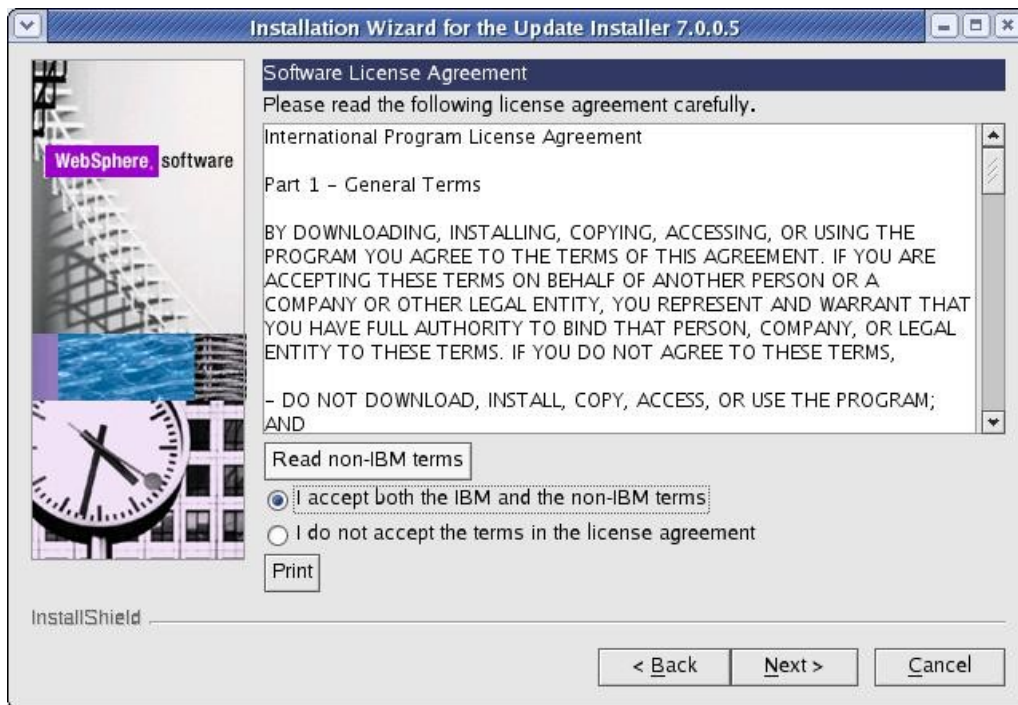
10. Review the information on the summary screen and click 'Next' to begin the installation.
11. After the installation completes, uncheck the option to create a new profile and click 'Finish' to exit the installation program.
12. Download the WebSphere Application Server v7 Update Installer:  
<http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg24020446>
13. Extract the download into a temporary directory and launch the installer located in the <temp location>/UpdateInstaller directory:

```
./install
```

14. Click 'Next' on the Welcome Screen:

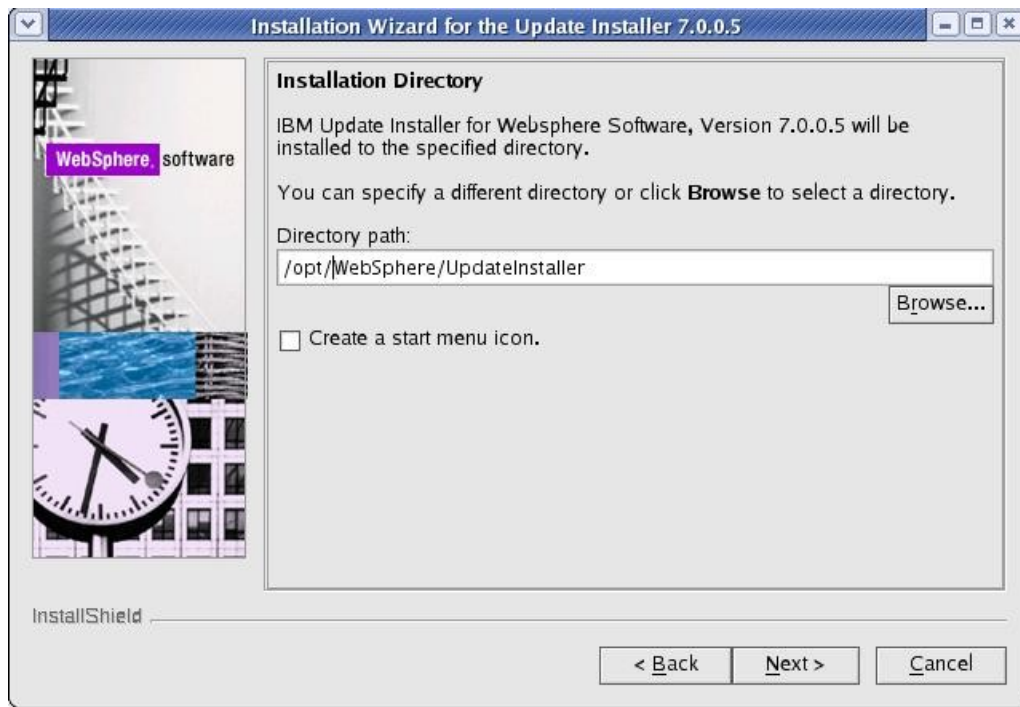


15. Accept the license and click 'Next':



16. On the system prerequisite screen, click 'Next'.

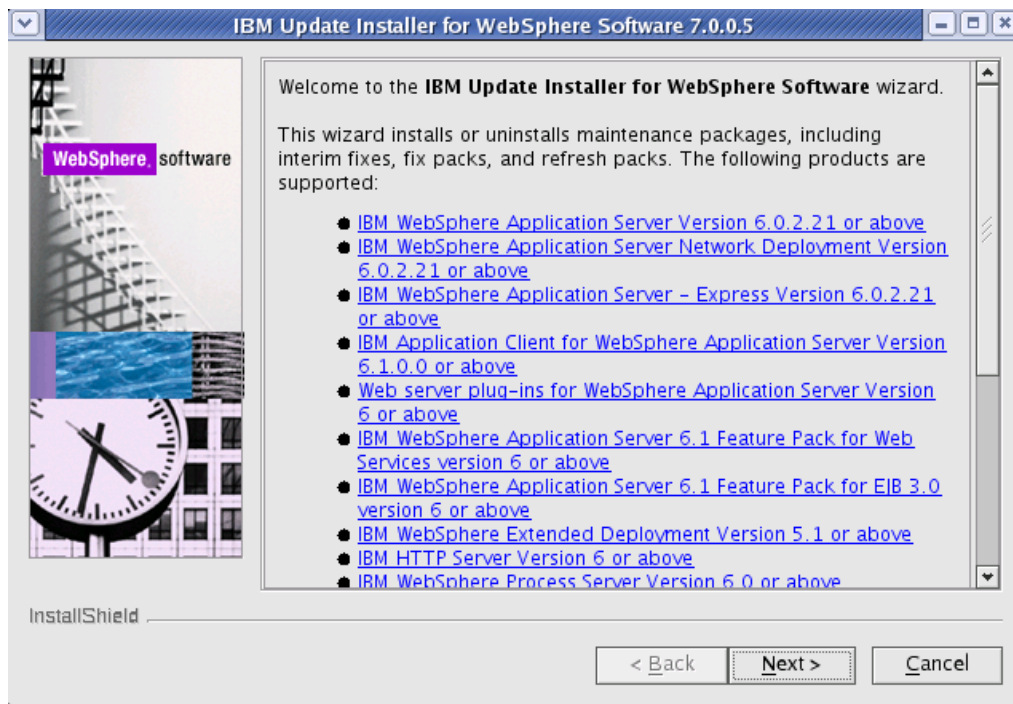
17. Select the path where you would like to install the WAS Update Installer:



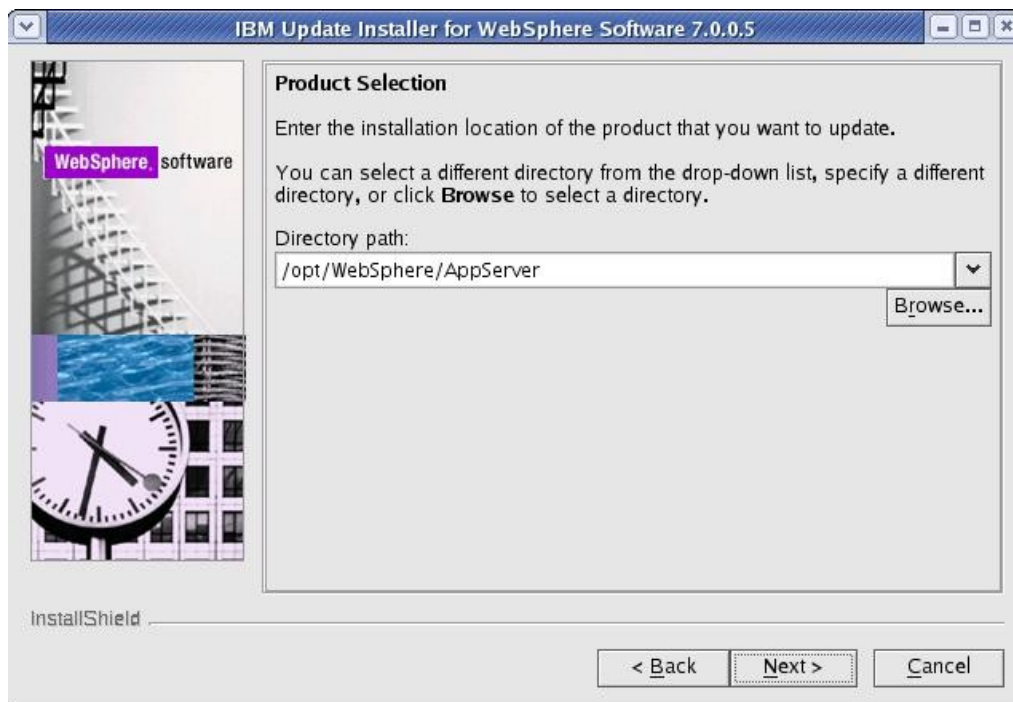
18. After the installation completes, click Finish to exit the installer.
19. Download the WAS 7.0.0.5 fixpack and the corresponding JDK upgrade:  
<http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg24023705>  
<http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg24023708>
20. Copy the 7.0.0.5 fixpack and the JDK upgrade to the <UpdateInstaller root>/maintenance directory, where <UpdateInstaller root> is the location you selected in step 17.
21. Launch the WAS Update Installer from the UpdateInstaller directory you set from step 17:  
`./update.sh`



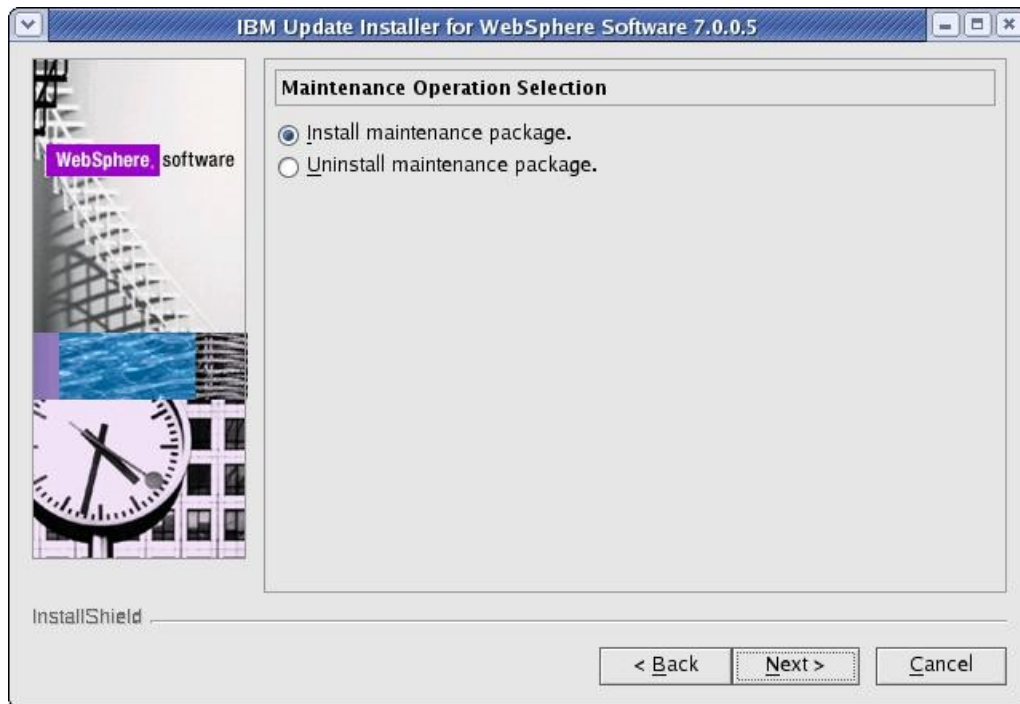
22. Click 'Next' on the Welcome Screen:



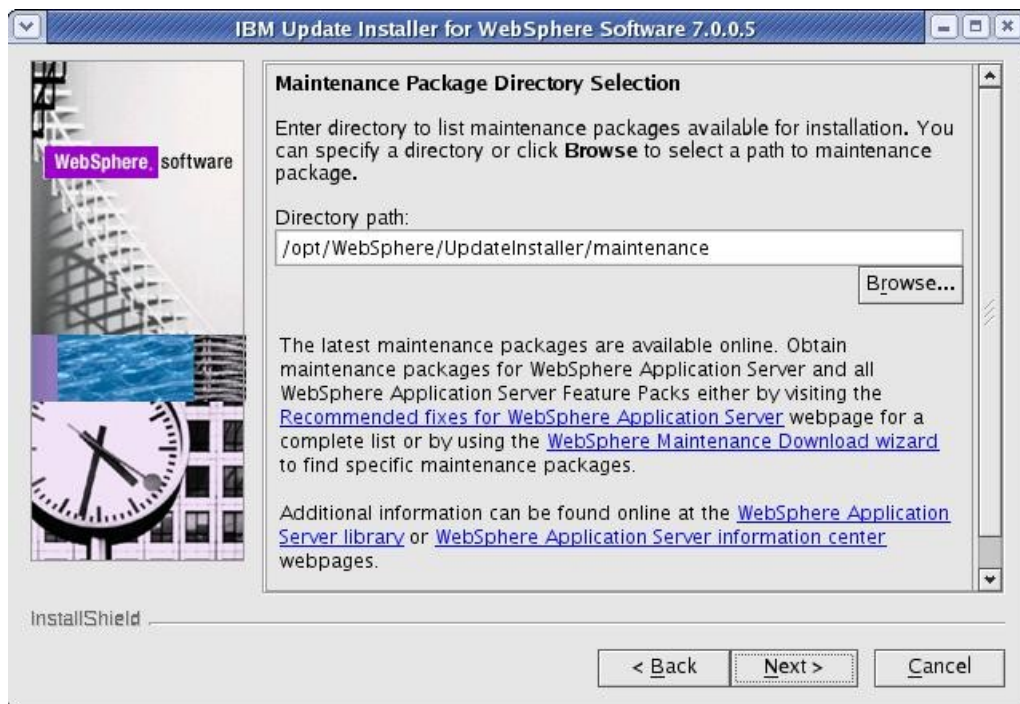
23. Select the WebSphere Application Server directory you wish to upgrade and click 'Next':



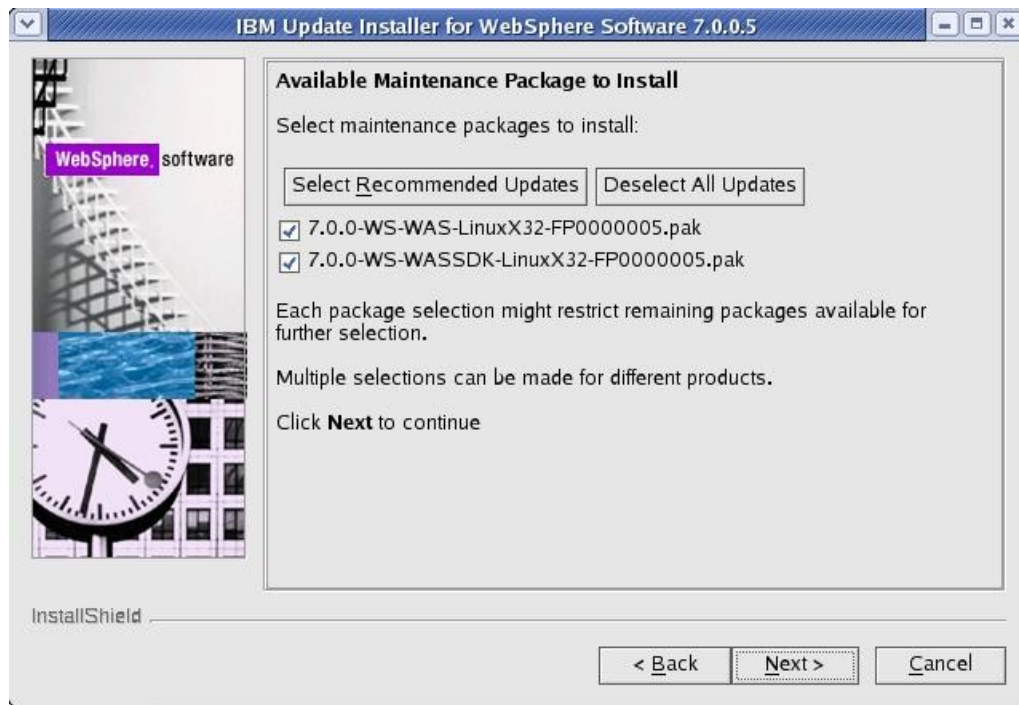
24. Select the 'Install Maintenance package' option and click 'Next':



25. Select the directory that contains the 7.0.0.5 and JDK packages:



26. Check the boxes for the 7.0.0.5 and JDK packages and click 'Next':



27. On the installation summary screen, click Next to begin the upgrade.

28. After the upgrade completes, click 'Finish' to exit the update installer.

29. Download the required WebSphere Application Server interim fixes for WAS v7.0.0.5 when using WebSphere Portal v6.1.0.3/6.1.5 from the WebSphere Portal Support site:

<http://www-01.ibm.com/support/docview.wss?rs=688&uid=swg24023148>

30. Extract the zip file into a temporary directory. This should include the following interim fixes:

PK90343  
PK91698  
PK92047  
PK93952  
PK96275  
PK97321  
PK98302  
PK98741  
PK99787  
PM00692

31. Copy the fixes into the <UpdateInstaller root>/maintenance directory where <UpdateInstaller root> is the location you selected in step 17.

32. Repeat steps 21-28 to install the interim fixes. (Note that for step 26 you will select to install the interim fixes listed in Step 30, not the 7005 and JDK fixpacks).

## Install the Secondary Portal Node

In this section, you will install the secondary Portal node. You will use the WAS v7005 that you installed from the previous section as the base for this Portal installation. All of the steps in this section will be done on the server you intend to use as your primary node.

1. Open a terminal window and enter:

```
ping yourserver.yourcompany.com
```

where *yourserver.yourcompany.com* is your actual fully qualified hostname.

2. Enter:

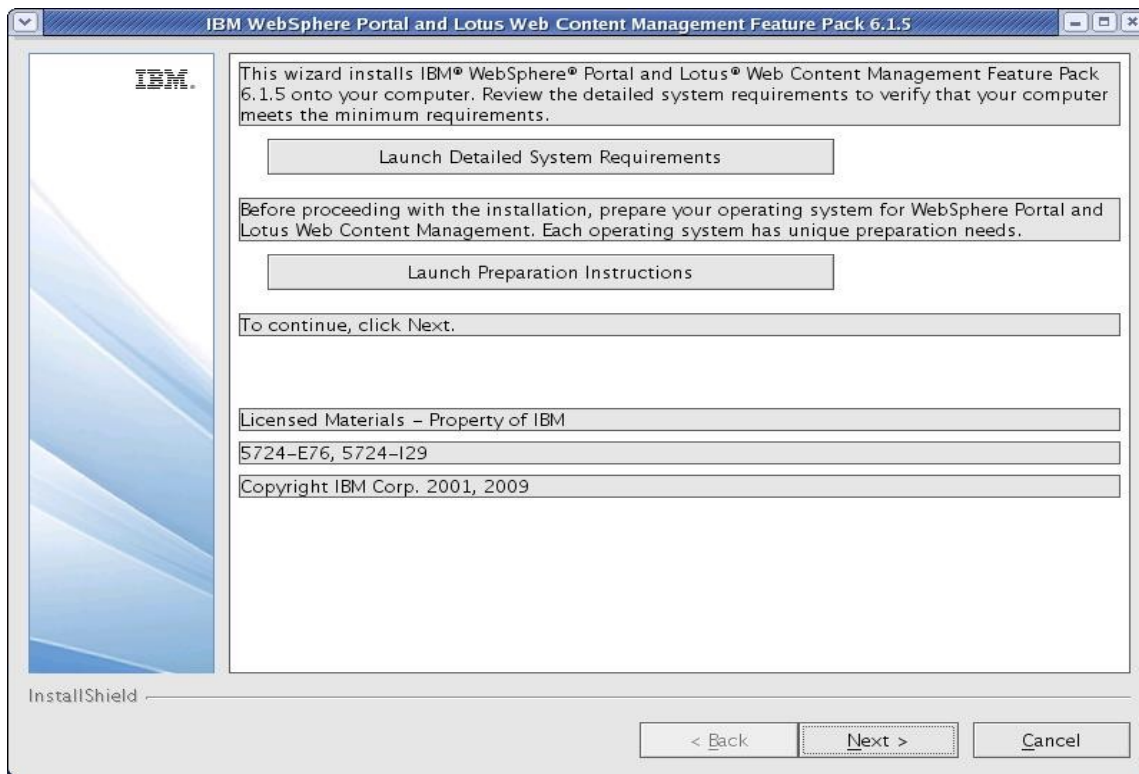
```
ping localhost
```

to verify the network settings are configured properly on your machine.

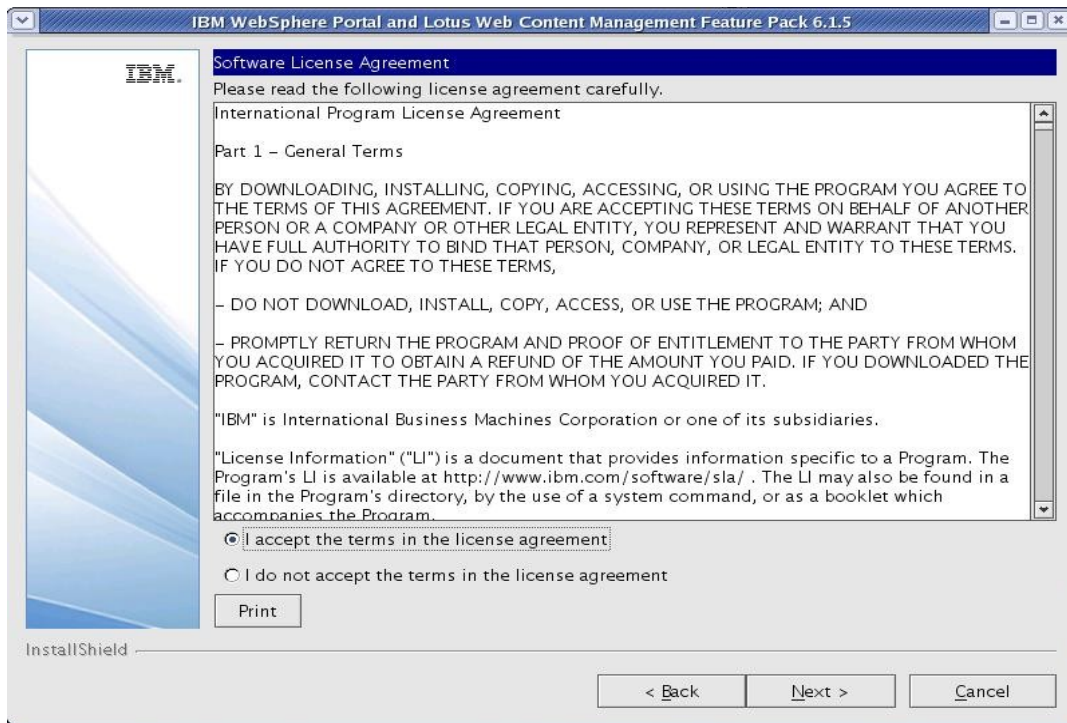
3. From the IL-Setup CD, launch the WebSphere Portal installer:

```
./install.sh
```

4. Click 'Next' on the Welcome screen.

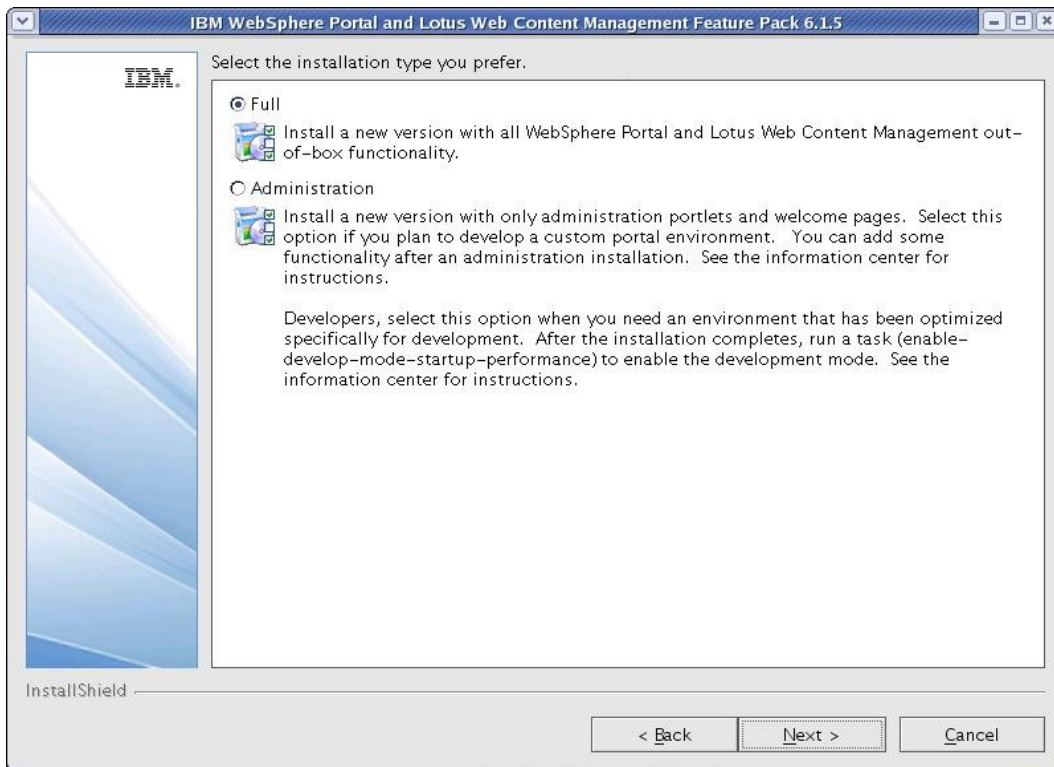


5. Accept the license agreement and click 'Next':

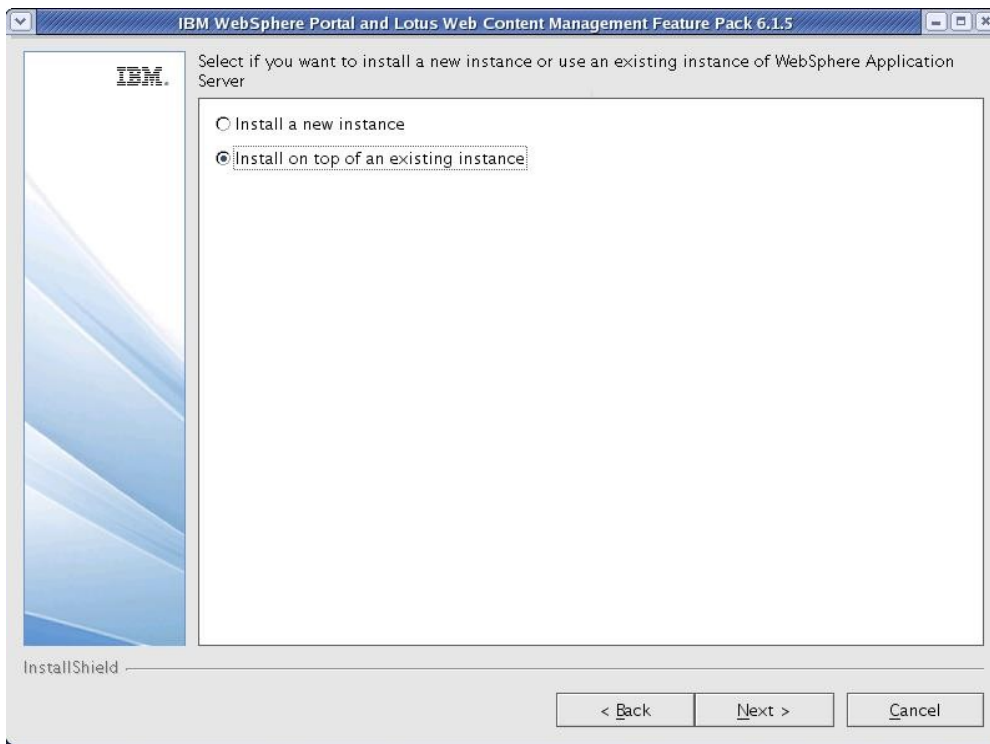


6. On the installation type screen, select 'Full' and click 'Next'

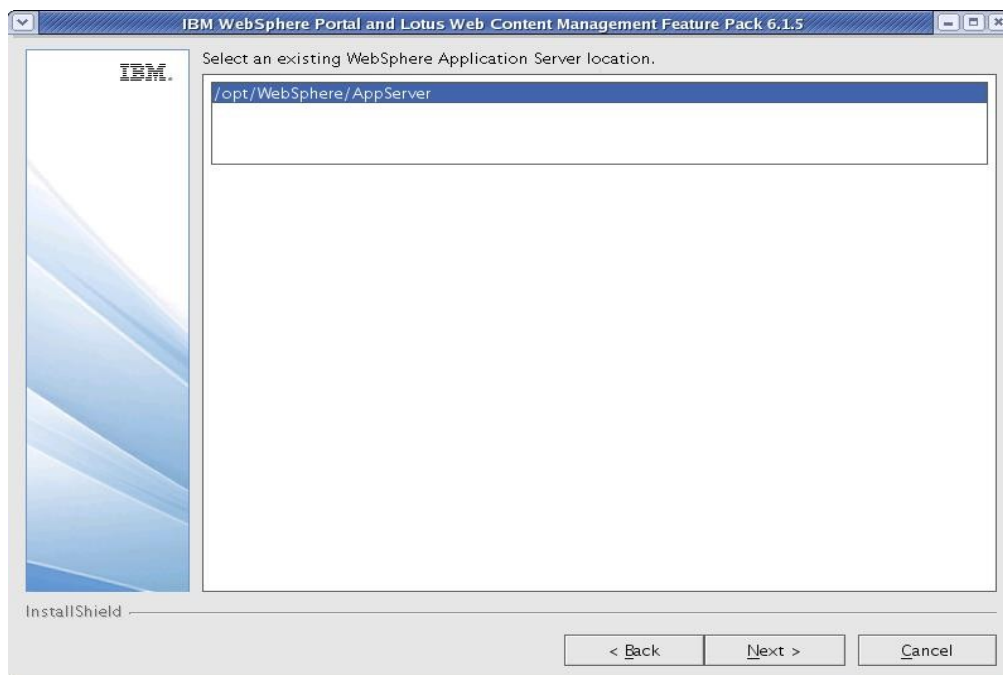
**NOTE:** Select Administration to install only administrative portlets.



7. Select to install to on top of an existing WAS installation:

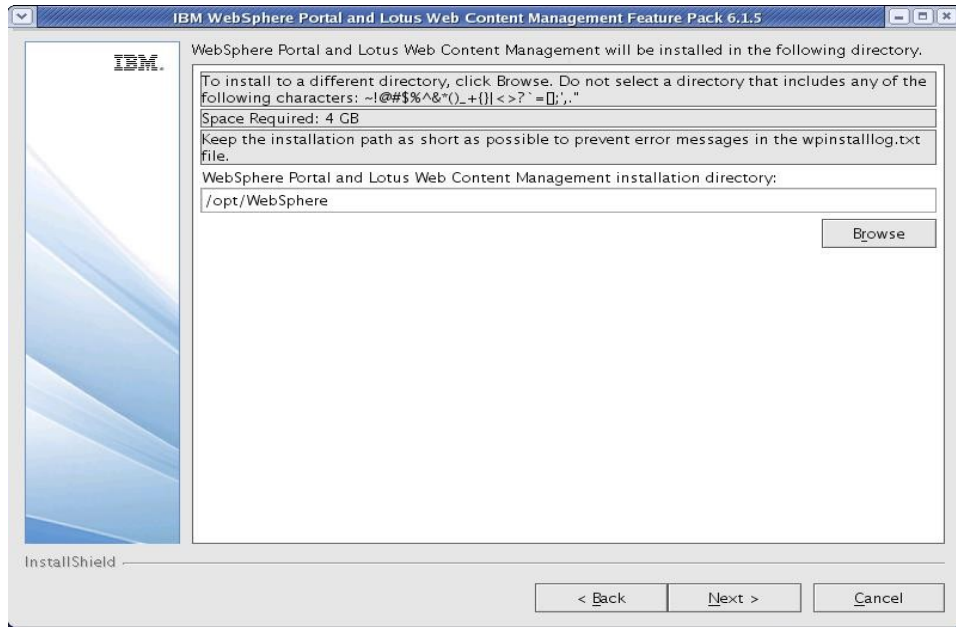


8. Select the path of the existing WAS installation:



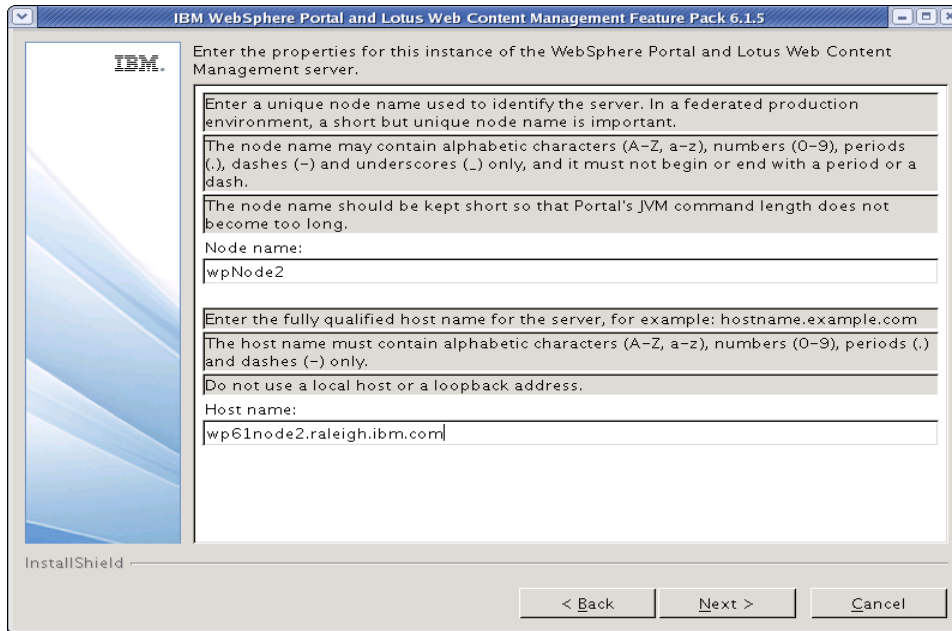
9. Select the desired path for the WebSphere directory and click 'Next'

**NOTE:** Both the profile directory and the PortalServer directory will be created in this WebSphere directory.



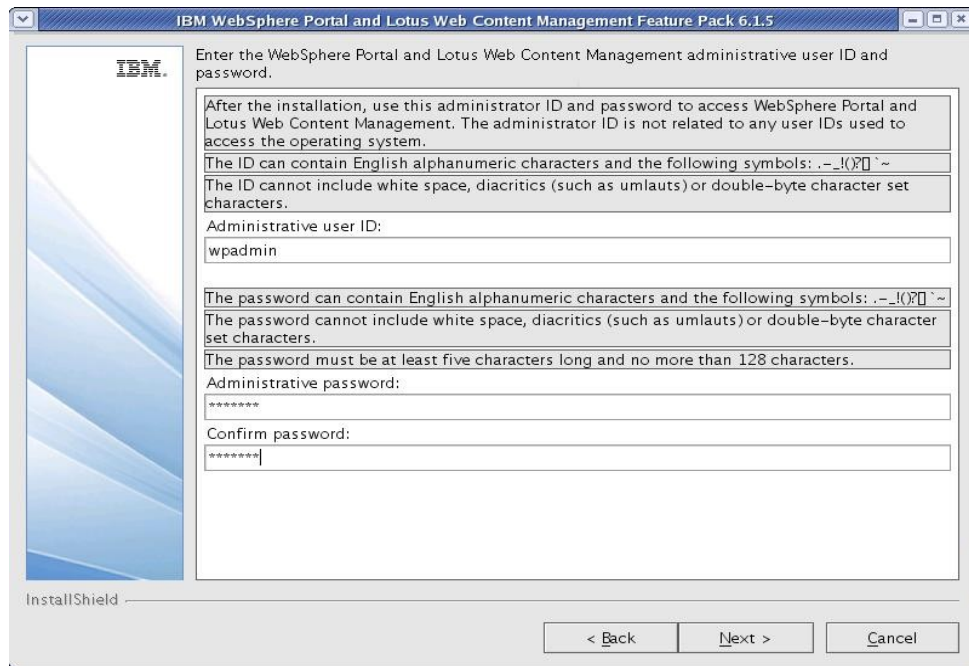
10. Enter a node name and the fully qualified hostname of your server and click 'Next'.

**NOTE:** The value for node name will also be used as the cell name in the standalone environment.





11. Security is enabled for Portal by default. Enter **the same** user ID and password you used for the Primary node installation.



12. Verify the information is accurate in the summary screen and click 'Next' to begin the installation.
13. Once the installation finishes, uncheck Launch First Steps and Launch the Configuration Wizard. Click 'Finish'.
14. Verify you can access Portal in a web browser. The default URL is:

<http://yourserver.yourcompany.com:10039/wps/portal>

## ***Install IBM Support Assistant Lite***

In this section, you will install IBM Support Assistant Lite for WebSphere Portal (ISALite). This step is optional but **highly** recommended. ISALite provides automatic log collection and symptom analysis support for WebSphere Portal problem determination scenarios. Installing this tool now can save you time in the future if you have any problems with WebSphere Portal that require you to contact support.

1. Visit the website below and download ISALite for WebSphere Portal v6.1 to a temporary directory:

<http://www-01.ibm.com/support/docview.wss?rs=688&uid=swg24008662>

2. Extract the downloaded zip file into the wp\_profile/PortalServer directory. This will create a directory called ISALite.
3. The tool is installed and ready for use. If you have an issue with WebSphere Portal that requires support, instructions for using this tool can be found in Appendix D.

## ***Federate and Cluster the Secondary Portal node***

This section covers adding the secondary node to the Deployment Manager cell and adding its WebSphere\_Portal server as a secondary member to the previously created cluster. Once this section is completed, you will have a functional two-node cluster using the default VMM security configuration and configured to an external database.

1. Ensure the database client is installed and configured on the secondary node. For DB2 with Type 4 drivers, copy the db2jcc.jar and db2jcc\_license\_cu.jar files from the DB2 server to some directory on the secondary Portal server.
2. From the <wp\_profile>/ConfigEngine/properties directory, make a backup of the following files:

```
wkplc.properties  
wkplc_dbtype.properties  
wkplc_comp.properties
```

3. Copy the wkplc\_comp.properties and wkplc\_dbtype.properties from Node1 to Node2 to ensure the same database configuration.

**NOTE:** Ensure that the value of db2.DbLibrary and derby.DbLibrary in wkplc\_dbtype.properties contain valid directory paths for this node.

4. On the secondary node, edit the <wp\_profile>/ConfigEngine/properties/wkplc.properties file and ensure all of the following properties are set appropriately for your environment:

```
WasUserId=<DMGR admin user ID>  
WasPassword=<DMGR admin password>  
PortalAdminPwd=<password>  
WasRemoteHostName=<fully qualified hostname of DMGR>  
WasSoapPort=<soap port for DMGR; default is 8879>  
PrimaryNode=false  
ClusterName=PortalCluster
```

**NOTE:** This guide was written specifically for Portal v6.1.0.3 and higher. If you are using WebSphere Portal v6.1.0.0, 6.1.0.1, or 6.1.0.2 then you **must** leave the WasUserId and WasPassword properties set to the **standalone** server admin values, NOT the DMGR values.

**NOTE:** Ensure that the value for ClusterName matches the value for ClusterName on the primary node.

5. Stop WebSphere\_Portal and server1 by executing the following commands from the <wp\_profile root>/bin directory:

```
./stopServer.sh WebSphere_Portal -user <admin user> -password <admin pwd>  
./stopServer.sh server1 -user <admin user> -password <admin pwd>
```

6. Ensure the DMGR is STARTED by running the following command from the <dmgr\_profile>/bin directory.

```
./startManager.sh
```

7. In a command window from the secondary node, change directories to <wp\_profile>/ConfigEngine

8. Add the node to the deployment manager cell by executing the following ConfigEngine script:

```
./ConfigEngine.sh cluster-node-config-pre-federation -DWasPassword=password
```

**NOTE:** If you are prompted to accept an SSL certificate, type Y and press Enter to continue

**NOTE:** This guide was written specifically for Portal v6.1.0.3 and higher. If you are using WebSphere Portal v6.1.0.0, 6.1.0.1, or 6.1.0.2 then you **must** specify -DDMgrUserid and -DDMgrPassword parameters when running the cluster-node-config-pre-federation task. For example:

```
./ConfigEngine.sh cluster-node-config-pre-federation -DDMgrUserid=<DMGRUser>  
-DDMgrPassword=<password>
```

**IMPORTANT:** If you receive a BUILD FAILED for the cluster-node-config-pre-federation script, you **MUST** run the following ConfigEngine script to clean up the WAS registry:

```
./ConfigEngine.sh -DWasRemoteHostName=<standalone hostname>  
-DWasSoapPort=<standalone soap port>
```

for example:

```
./ConfigEngine.sh -DWasRemoteHostName=localhost -DWasSoapPort=10033
```

9. After the previous step completes, your node will be part of the deployment manager cell. As a result, this node is now using the Deployment Manager security configuration and cell name. The original WAS ID that had been used in the standalone environment will no longer be used.

Edit the <wp\_profile>/ConfigEngine/properties/wkplc.properties file and ensure the following properties are set correctly:

```
WasUserId=<dmgr admin user>  
WasPassword=<dmgr password>  
CellName=<dmgr cell name>
```

10. Update the deployment manager configuration for the new WebSphere Portal server by executing the following ConfigEngine script:

```
./ConfigEngine.sh cluster-node-config-post-federation -DWasPassword=password
```

11. Ensure the NodeAgent is started on this node by running the following command from the <wp\_profile>/bin directory:

```
./StartNode.sh
```

12. Specify the name of the future secondary cluster member.

Edit the <wp\_profile>/ConfigEngine/wkplc.properties file and change the following property:

```
ServerName=<name of new cluster member>
```

**NOTE:** When you open the properties file, you should see `WebSphere_Portal_nodename`. You can use this value if you like. Otherwise you can change this to anything EXCEPT 'WebSphere\_Portal'. DO NOT use the value of 'WebSphere\_Portal' for your secondary cluster member.

13. Ensure that the operating system time on the Deployment Manager server and the time on the primary node are within 5 minutes of each other. This is necessary for Steps 12 to complete successfully.
14. Add this newly federated WebSphere\_Portal server as a cluster member to the existing cluster by executing the following ConfigEngine script:

```
./ConfigEngine.sh cluster-node-config-cluster-setup -DWasPassword=password
```

**NOTE:** This will automatically add a secondary cluster member to your existing cluster based on whatever value you set for `ServerName` in step 10. In this example, the default value was used. The node name is `wpNode2` so our cluster member will be called `WebSphere_Portal_wpNode2`.

15. Allow **30 minutes** for ear expansion to complete on the secondary node. Failure to do so may result in several applications being unavailable on this node.

16. Start the new cluster member `WebSphere_Portal_nodename` from the `wp_profile/bin` directory:

```
./startServer.sh WebSphere_Portal_nodename
```

17. To verify that the cluster was created successfully, log in to the DMGR Administrative Console and browse to:

```
Servers -> Clusters -> WebSphere Application Server Clusters -> ClusterName  
-> Cluster Members
```

An entry for `WebSphere_Portal_nodename` should be available.

[WebSphere application server clusters](#) > [PortalCluster](#) > Cluster members

Use this page to view and manage application servers that belong to a cluster. You can also use this page to change the weight of any of the listed application servers. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic. The configuration of new cluster members is based on a server configuration template that is stored as part of the cluster data. This template is based on the first cluster member and is used to create all subsequent cluster members. Modifications to the configuration of an individual cluster member has no effect on the cluster member template.

Preferences

Select	Member name	Node	Host Name	Version	Configured weight	Runtime weight	Status
<input type="checkbox"/>	<a href="#">WebSphere_Portal</a>	wpNode1	wp61Node1.raleigh.ibm.com	ND 7.0.0.5	<input type="text" value="2"/>	<input type="text" value="2"/>	
<input type="checkbox"/>	<a href="#">WebSphere_Portal_wpNode2</a>	wpNode2	wp61node2.raleigh.ibm.com	ND 7.0.0.5	<input type="text" value="2"/>	<input type="text"/>	

Total 2

**NOTE:** In this example, the `WebSphere_Portal_wpNode2` server is a new server in this configuration. The original `WebSphere_Portal` server from the secondary node gets removed during the `cluster-node-config-cluster-setup` ConfigEngine script. As a result, new port numbers have been assigned to the `WebSphere_Portal_nodename` server. To check what ports are in use with this server, navigate to:

```
Servers -> Server Types -> Application Servers -> WebSphere_Portal_nodename  
-> Ports
```

The `WC_defaulthost` is the port used to access Portal. The default port in this case is 10050.

If you need to change these port numbers, you can do so from this screen. Alternatively, ConfigEngine scripts are provided to modify port numbers. Details can be found in Step 9 of the Information Center instructions found here:

[http://publib.boulder.ibm.com/infocenter/wpdoc/v6r1/index.jsp?topic=/com.ibm.wp.ent.doc\\_v615/install/linux\\_inst\\_wp\\_clus.html](http://publib.boulder.ibm.com/infocenter/wpdoc/v6r1/index.jsp?topic=/com.ibm.wp.ent.doc_v615/install/linux_inst_wp_clus.html)

18. Verify functionality of the secondary node by accessing it in a web browser:

<http://mycompany.myserver.com:10050/wps/portal>

At this point, you have successfully built a two-node WebSphere Portal cluster using a remote database and VMM file federated security.

## Configure the Portal Cluster for Security

This section covers changing the security configuration from the default user registry to a standalone LDAP Server. For more details about LDAP/Security configuration, please refer to the Information Center:

[http://publib.boulder.ibm.com/infocenter/wpdoc/v6r1/index.jsp?topic=/com.ibm.wp.ent.doc\\_v615/install/linux\\_cfg\\_wp\\_ureg\\_clus.html](http://publib.boulder.ibm.com/infocenter/wpdoc/v6r1/index.jsp?topic=/com.ibm.wp.ent.doc_v615/install/linux_cfg_wp_ureg_clus.html)

Security configuration has changed significantly in Portal v6.1.0.x. The 'disable-security' script from Portal v5.1.x and v6.0.x no longer exists. Instead, a single ConfigEngine script is executed to change from one user registry to another, or to update an existing user registry. There are several different options for security configuration and we encourage you to review all options in the Information Center from the link above to determine what is best for your environment.

In this guide, we will configure security in our cluster to a standalone ldap server using IBM Tivoli Directory Server v6.1.

1. From the primary node, edit the wp\_security\_ids.properties file in the <wp\_profile>/ConfigEngine/config/helpers directory.
2. Modify the following properties in this helper file to match your LDAP configuration. The values used in this guide are listed below:

```
standalone.ldap.id=PortalLdap
standalone.ldap.host=myldapserver.rtp.raleigh.ibm.com
standalone.ldap.port=389
standalone.ldap.bindDN=uid=wpbind,cn=users,dc=ibm,o=com
standalone.ldap.bindPassword=wpbind
standalone.ldap.ldapServerType=IDS
standalone.ldap.userIdMap=*:uid
standalone.ldap.groupIdMap=*:cn
standalone.ldap.groupMemberIdMap=ibm-allGroups:member;ibm-
allGroups:uniqueMember
standalone.ldap.userFilter=( &(uid=%v)(objectclass=inetOrgPerson))
standalone.ldap.groupFilter=( &(cn=%v)(objectclass=groupOfUniqueNames))
standalone.ldap.serverId=uid=wpbind,cn=users,dc=ibm,o=com
standalone.ldap.serverPassword=wpbind
standalone.ldap.realm=PortalRealm
standalone.ldap.primaryAdminId=uid=wpadmin,cn=users,dc=ibm,o=com
standalone.ldap.primaryAdminPassword=wpadmin
standalone.ldap.primaryPortalAdminId=uid=wpadmin,cn=users,dc=ibm,o=com
standalone.ldap.primaryPortalAdminPassword=wpadmin
standalone.ldap.primaryPortalAdminGroup=cn=wpsadmins,cn=groups,dc=ibm,o=com
standalone.ldap.baseDN=dc=ibm,o=com
```



```
standalone.ldap.et.group.searchFilter=objectclass=groupOfUniqueNames
standalone.ldap.et.group.objectClasses=groupOfUniqueNames
standalone.ldap.et.group.objectClassesForCreate=
standalone.ldap.et.group.searchBases=cn=groups,dc=ibm,o=com
```

```
standalone.ldap.et.personaccount.searchFilter=objectclass=inetOrgPerson
standalone.ldap.et.personaccount.objectClasses=inetOrgPerson
standalone.ldap.et.personaccount.objectClassesForCreate=
standalone.ldap.et.personaccount.searchBases=cn=users,dc=ibm,o=com
```

```
standalone.ldap.gm.groupMemberName=uniqueMember
standalone.ldap.gm.objectClass=groupOfUniqueNames
standalone.ldap.gm.scope=direct
standalone.ldap.gm.dummyMember=uid=dummy
```

```
standalone.ldap.personAccountParent=cn=users,dc=ibm,o=com
standalone.ldap.groupParent=cn=groups,dc=ibm,o=com
standalone.ldap.personAccountRdnProperties=uid
standalone.ldap.groupRdnProperties=cn
```

**NOTE:** The properties in the 'Advanced Properties' section of the helper file were not modified from the defaults in this example.

3. From a terminal window, change directories to the <wp\_profile>/ConfigEngine directory and execute the following ConfigEngine script to validate the properties:

```
./ConfigEngine.sh validate-standalone-ldap
-DparentProperties=<wp_profile>/ConfigEngine/config/helpers/wp_security_ids.p
roperties -DsaveParentProperties=true -DWasPassword=<password>
```

**NOTE:** By using the

-DparentProperties=<wp\_profile>/ConfigEngine/config/helpers/wp\_security\_ids.p  
roperties -DsaveParentProperties=true flags, ConfigEngine will automatically save the  
properties from the helper file into the wkplc.properties file.

**NOTE:** WasPassword should be the **current** WAS password, NOT your intended LDAP user password.

4. Execute the following ConfigEngine script to modify the security settings from the default VMM file security settings to the new LDAP settings:

```
./ConfigEngine.sh wp-modify-ldap-security
```

**NOTE:** This script will automatically change WasUserId, PortalAdminId and PortalAdminGroupId in wkplc.properties to match that of standalone.ldap.primaryAdminId, standalone.ldap.primaryPortalAdminId, and standalone.ldap.primaryPortalAdminGroup.

- Restart the DMGR, all NodeAgents, and all Cluster Members.
- In the wkplc.properties on the secondary node, edit the following properties to reflect your LDAP values:

```
WasUserId  
WasPassword  
PortalAdminId  
PortalAdminIdPwd  
PortalAdminGroupId
```

- Copy the wp\_security\_ids.properties from the <wp\_profile>/ConfigEngine/config/helpers directory on your Primary node to the <wp\_profile>/ConfigEngine/config/helpers directory on the secondary node.
- From the secondary node, copy the contents of the helper file into the main wkplc.properties file by running the following command (all on one line):

```
./ConfigEngine.sh  
-DparentProperties=<wp_profile>/ConfigEngine/config/helpers/wp_security_ids.p  
roperties -DsaveParentProperties=true -DWasPassword=<LDAP Password>
```

- Update the Portal security information on the secondary node by executing the following ConfigEngine script from the <wp\_profile>/ConfigEngine directory on your secondary node:

```
./ConfigEngine.sh enable-jcr-security -DWasPassword=<LDAP password>
```

- Restart the secondary node's WebSphere\_Portal server by executing the following commands from the <wp\_profile>/bin directory on the secondary node:

```
./stopServer.sh WebSphere_Portal_nodename -user <WAS user ID> -password  
password  
./startServer.sh WebSphere_Portal_nodename
```

## Configure the Portal Cluster with an external web server

This section describes how to configure the Portal cluster with an external web server. For more details about web server configuration, please visit the WebSphere Portal Server Information Center at this link:

[http://publib.boulder.ibm.com/infocenter/wpdoc/v6r1/index.jsp?topic=/com.ibm.wp.ent.doc\\_v615/install/linux\\_prep\\_ihs.html](http://publib.boulder.ibm.com/infocenter/wpdoc/v6r1/index.jsp?topic=/com.ibm.wp.ent.doc_v615/install/linux_prep_ihs.html)

In this guide, we will configure the Portal cluster with IBM HTTP Server v6.1.

1. From CD IL-13, navigate to \IHS\ and run the following command:

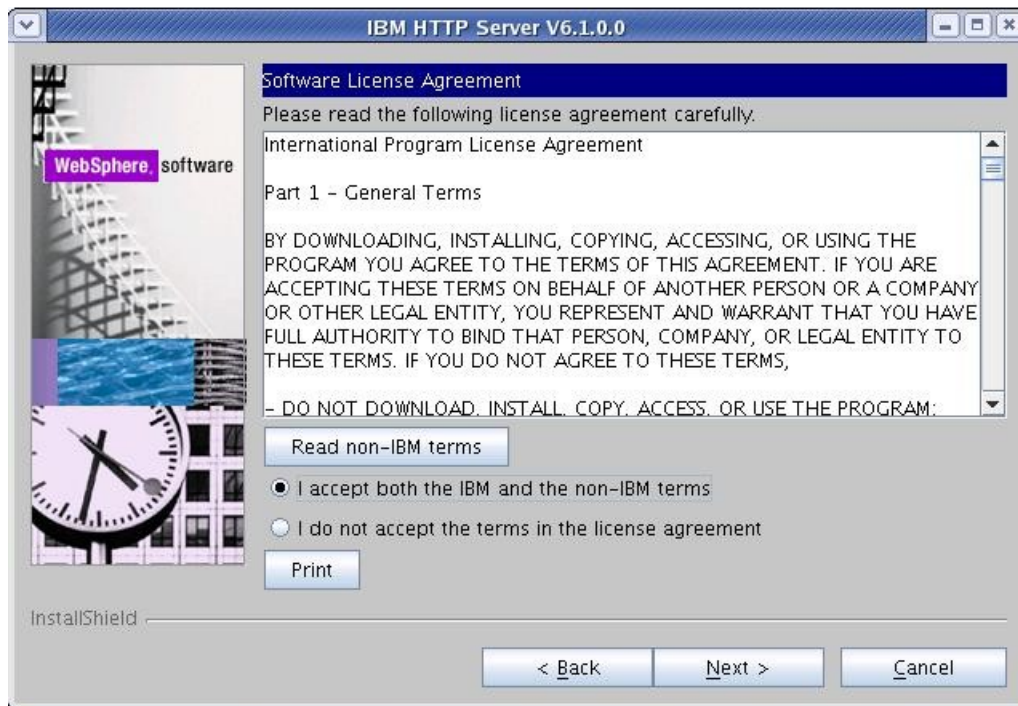
```
./install
```

**NOTE:** The CD/image that contains the IHS installer will vary on each operating system. The title of the CD/image is “Edge Components for WebSphere Application Server Network Deployment”.

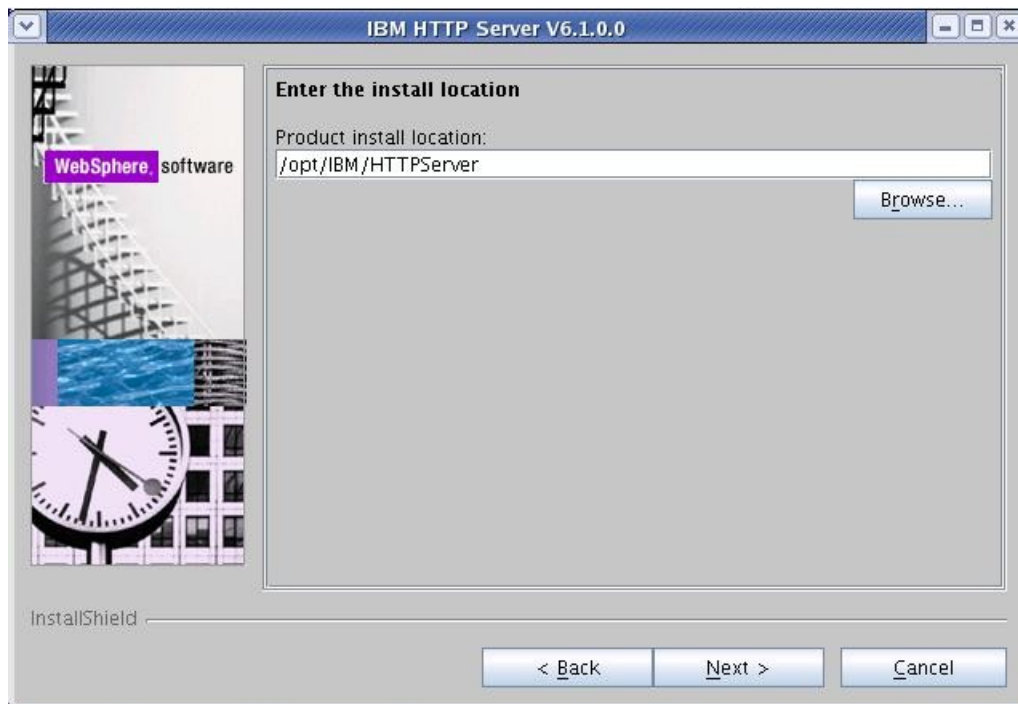
2. On the Welcome screen, click Next.



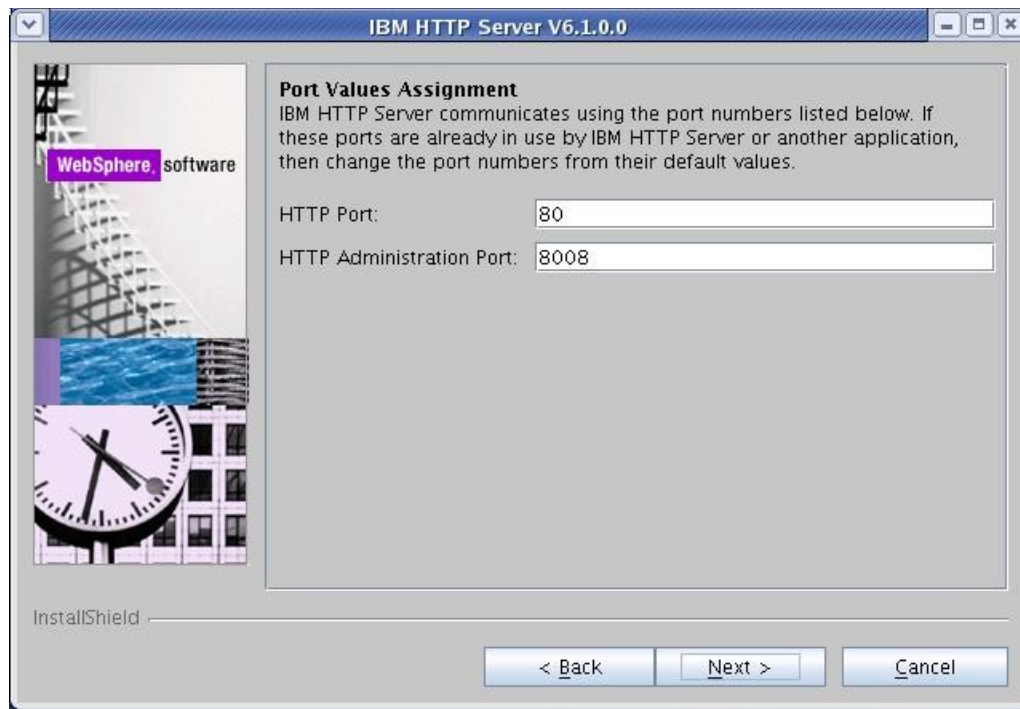
Accept the license agreement and click Next.



3. Select the installation path for the web server and click Next.



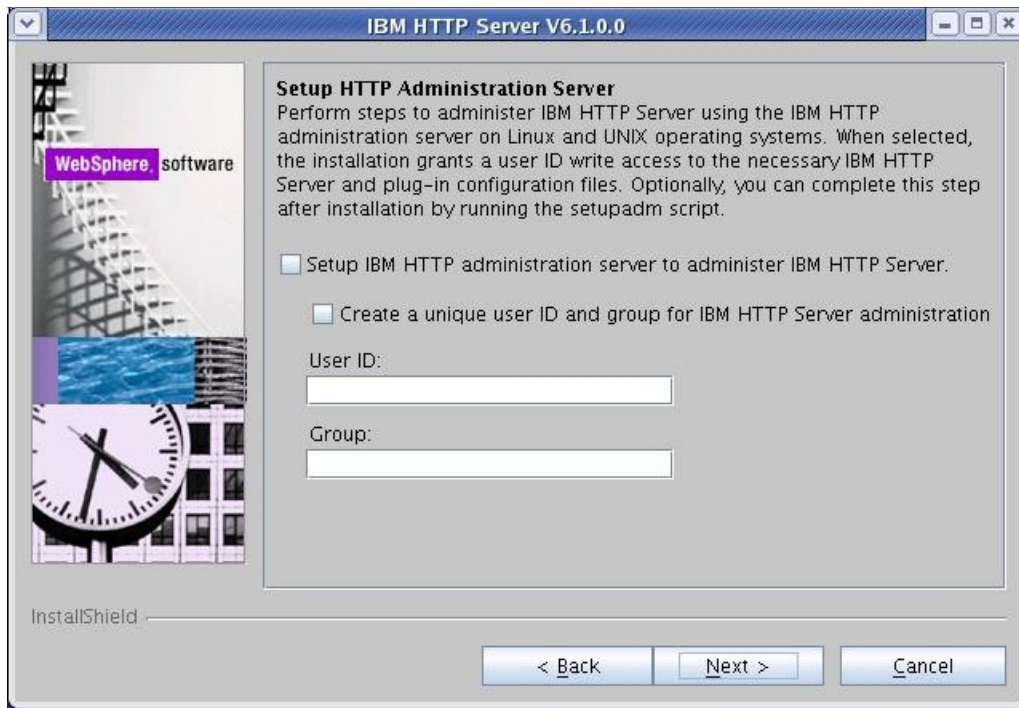
4. Change the port numbers if needed and click Next.



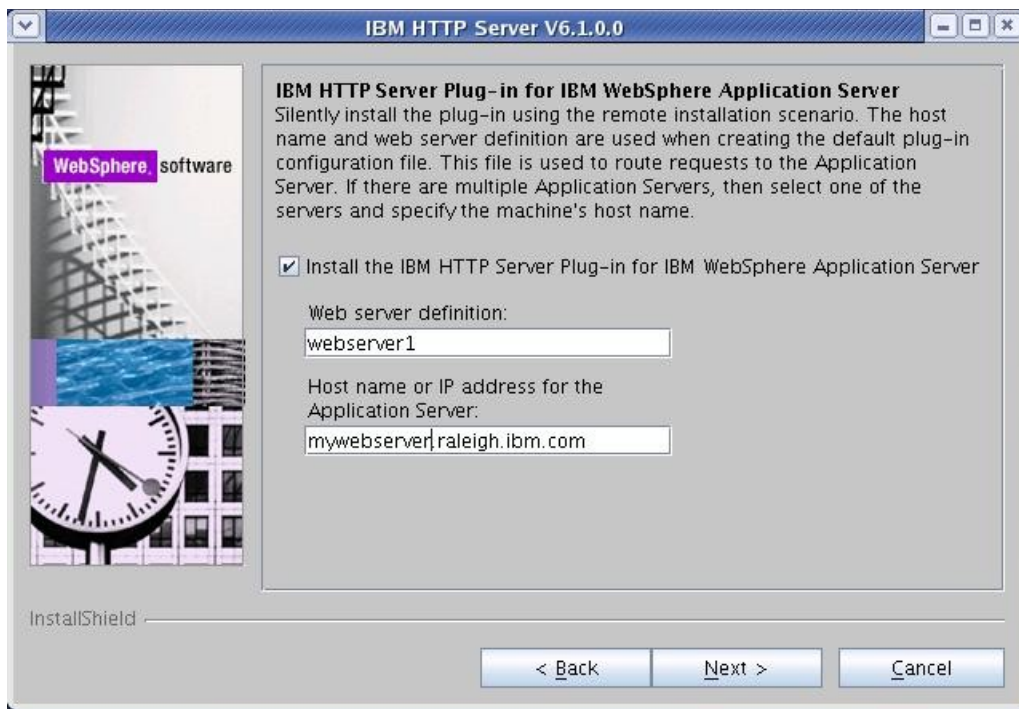
5. Create a user ID and password to be used for authentication to the IBM HTTP Administration server and click Next.



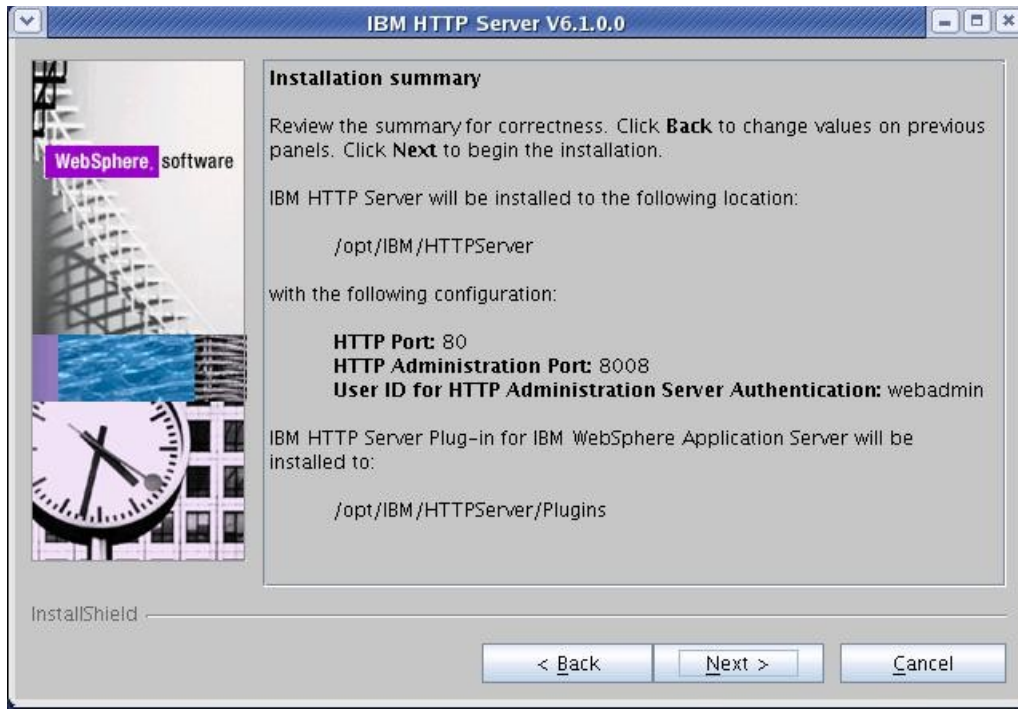
6. **Unix only.** Select to setup an IBM HTTP Administration Server if you'd like. For this guide, this option was **unchecked**. Click Next.



7. Select the checkbox to install the Web Server plugin as part of the Web Server installation. Select a web server definition value and ensure the hostname is correct for this server. Click Next.



8. On the summary screen, ensure everything is correct and click Next to begin the installation.



9. Once the installation finishes, click Finish to exit the installer.
10. Navigate to <plugin root>/bin and find the `configurewebservername.sh` script where `webservername` is the web server definition name you defined on step 8. In this case, we used `webserver1` so our script is called:

```
configurewebserver1.sh
```

11. Copy the `configurewebserver1.sh` script from the <plugin root>/bin directory to the <dmgr\_profile>/bin directory on your Deployment Manager server.
12. Ensure that the DMGR is running.
13. In a command line from the <dmgr\_profile>/bin directory, run the following command:

```
./configurewebserver1.sh -user <was_admin_user> -password password
```

**NOTE:** This script will create the web server definition in the DMGR configuration and map all of the installed applications to the web server.

14. Regenerate the web server plugin by performing the following steps:

1. Login to the DMGR Admin Console
2. Navigate to Servers -> Server Types -> Web Servers
3. Select the Checkbox for the new web server definition
4. Click the “Generate Plug-in” button

**NOTE:** This will be written to the  
<dmgr\_profile>/config/cells/<cellname>/nodes/<nodename>/servers/webserver1/plugin-  
cfg.xml file.

15. Copy the plugin-cfg.xml file to the remote web server at the following directory, overwriting the existing one:

<plugin\_root>/config/webserver1

16. Restart the DMGR, web server, and cluster.

17. Verify that you can access the Portal cluster via the web server:

<http://mywebserver.hostname.com/wps/portal>

## **Conclusion**

In this guide, you saw how to build a fully functional WebSphere Portal v6.1.0.3/6.1.5 cluster using an external database and a LDAP for security. You also saw how to configure a web server to allow for load balancing.



## Appendix A – Create a Deployment Manager profile on the Primary Portal node.

In this section you will create a deployment manager profile on that same server that contains your WebSphere Portal primary node. This is an optional section and is meant to be an alternative to installing the Deployment Manager on a separate server.

All of these steps will be completed on the server you intend to use as both your Primary Portal Node and Deployment Manager.

1. Launch a terminal window and navigate to the <AppServer root>/bin/ProfileManagement directory.

2. Launch the Profile Management Tool:

```
./pmt.sh
```

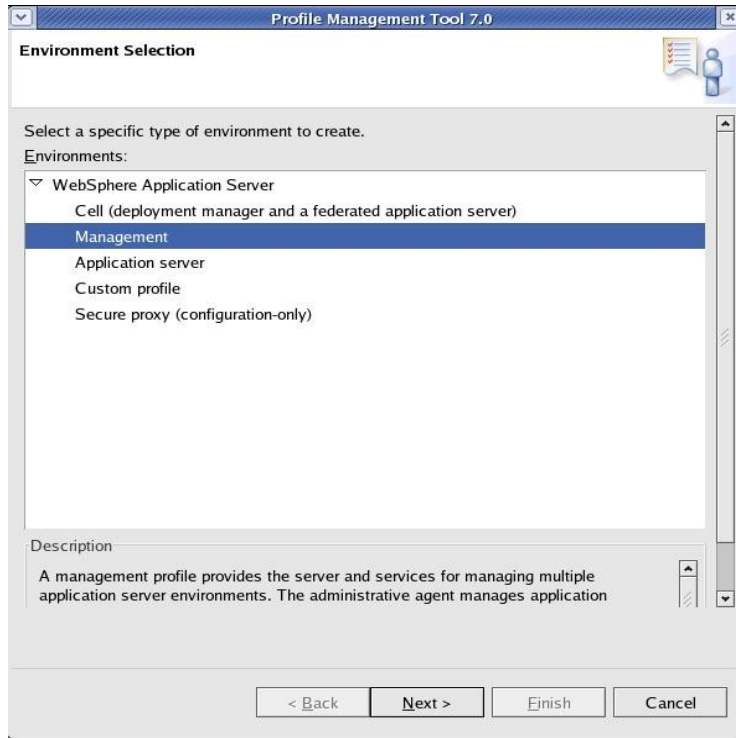
3. On the Welcome Screen, click the button for “Launch Profile Management”:



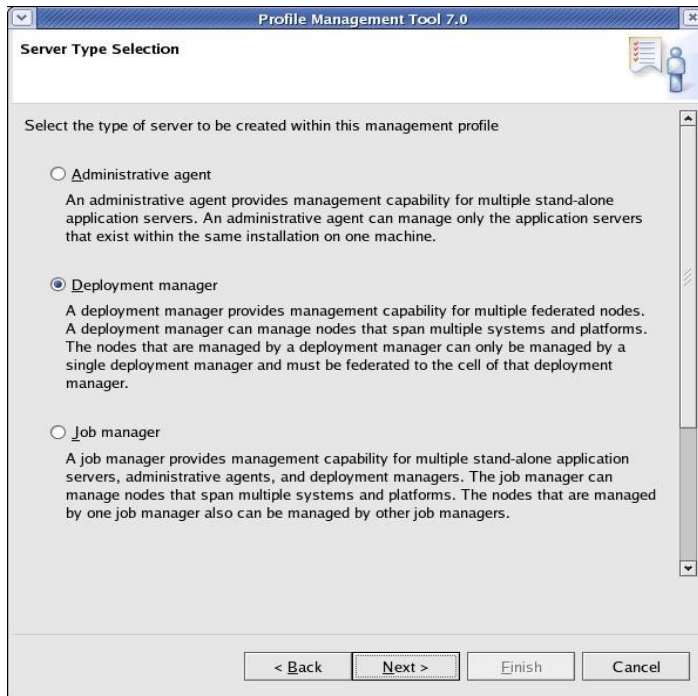
4. Click the button for “Create” to create a new profile:



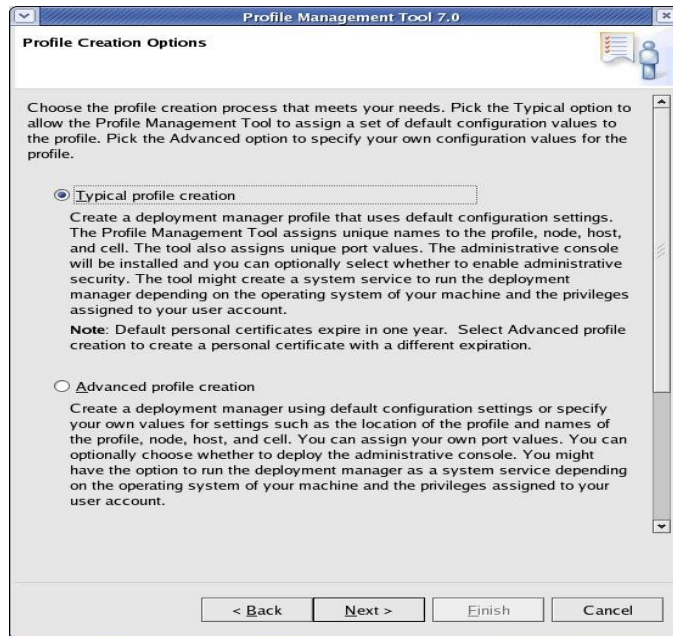
5. On the 'Environment Selection' screen, select 'Management' and click Next:



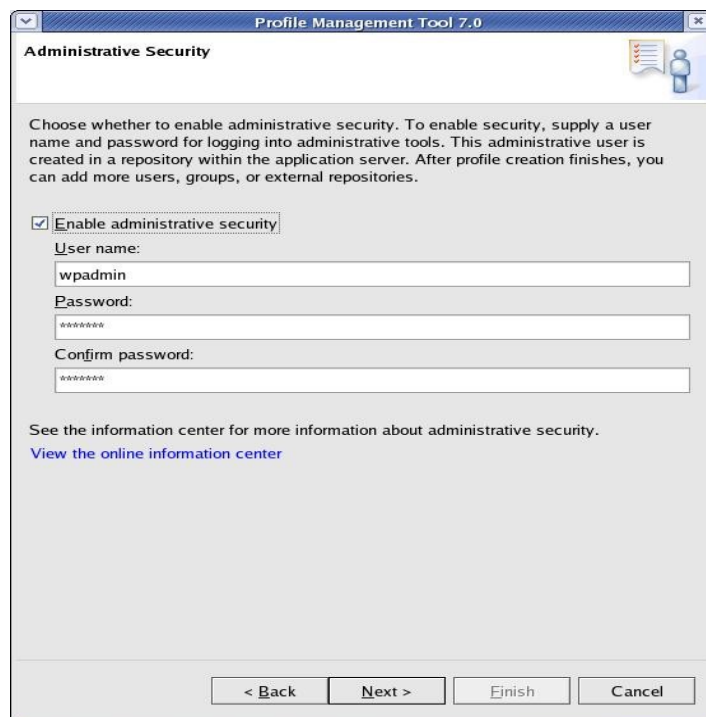
6. On the 'Server Type' screen, select 'Deployment Manager' and click Next:



7. On the 'Profile Creation Options' screen, you can select either Typical or Advanced. In this example, we use 'Typical'. Click Next:



8. On the 'Administrative Security' screen, select the checkbox to enable security and enter the **same** credentials you used for the WebSphere Portal installation. Click Next.



9. On the summary screen, review the information and click Create.
10. When the profile creation completes, uncheck the box for 'Launch First Steps' and click Finish.

At this point, the Deployment Manager profile has been created on the same server as your primary node. You can continue building your cluster starting at the 'Configuring the Deployment Manager' section.

## Appendix B – SQL Script to Create DB2 Databases

The following section contains the contents of the SQL script used to create the WebSphere Portal DB2 databases. To use this script, complete the following steps:

1. Copy the contents of this section into a text file
2. Edit the database names, user names and passwords in the file to match those of your intended environment. **Do NOT** change the JCR bufferpool or tablespace names. These **must** be the values listed here.
3. Save the file as a .sql file (for example CreateDatabases.sql)
4. Copy the file to a temporary directory on the DB2 server.
5. As the database administrator, execute the script:

```
db2 -tvf <temporary location>/CreateDatabases.sql
```

This script does all of the following:

- Creates and updates six databases (you may change these names): reldb, comdb, cusdb, jcrdb, lmdb, fdbkdb
- Creates bufferpools for jcrdb. **DO NOT** change these names: ICMLSFREQBP4, ICMLSVOLATILEBP4, ICMLSMAINBP32, CMBMAIN4.
- Creates tablespaces for jcrdb. **DO NOT** change these names: ICMLFQ32, ICMLNF32, ICMVFQ04, ICMSFQ04, CMBINV04, ICMLSSYSTSPACE32, ICMLSSYSTSPACE4

```
=====BEGIN COPY HERE=====DO NOT INCLUDE THIS LINE=====
```

```
CREATE DB reldb using codeset UTF-8 territory us PAGESIZE 8192;
UPDATE DB CFG FOR reldb USING applheapsz 4096;
UPDATE DB CFG FOR reldb USING app_ctl_heap_sz 1024;
UPDATE DB CFG FOR reldb USING stmtheap 32768;
UPDATE DB CFG FOR reldb USING dbheap 2400;
UPDATE DB CFG FOR reldb USING locklist 1000;
UPDATE DB CFG FOR reldb USING logfilsiz 4000;
UPDATE DB CFG FOR reldb USING logprimary 12;
UPDATE DB CFG FOR reldb USING logsecond 20;
UPDATE DB CFG FOR reldb USING logbufsz 32;
UPDATE DB CFG FOR reldb USING avg_appls 5;
UPDATE DB CFG FOR reldb USING locktimeout 30;
UPDATE DB CFG FOR reldb using AUTO_MAINT off;
```

```
CREATE DB comdb using codeset UTF-8 territory us PAGESIZE 8192;
UPDATE DB CFG FOR comdb USING applheapsz 4096;
UPDATE DB CFG FOR comdb USING app_ctl_heap_sz 1024;
UPDATE DB CFG FOR comdb USING stmtheap 32768;
UPDATE DB CFG FOR comdb USING dbheap 2400;
UPDATE DB CFG FOR comdb USING locklist 1000;
UPDATE DB CFG FOR comdb USING logfilsiz 4000;
UPDATE DB CFG FOR comdb USING logprimary 12;
UPDATE DB CFG FOR comdb USING logsecond 20;
UPDATE DB CFG FOR comdb USING logbufsz 32;
UPDATE DB CFG FOR comdb USING avg_appls 5;
UPDATE DB CFG FOR comdb USING locktimeout 30;
UPDATE DB CFG FOR comdb using AUTO_MAINT off;
```

```
CREATE DB cusdb using codeset UTF-8 territory us PAGESIZE 8192;
UPDATE DB CFG FOR cusdb USING applheapsz 4096;
UPDATE DB CFG FOR cusdb USING app_ctl_heap_sz 1024;
UPDATE DB CFG FOR cusdb USING stmtheap 32768;
UPDATE DB CFG FOR cusdb USING dbheap 2400;
UPDATE DB CFG FOR cusdb USING locklist 1000;
UPDATE DB CFG FOR cusdb USING logfilsiz 4000;
UPDATE DB CFG FOR cusdb USING logprimary 12;
UPDATE DB CFG FOR cusdb USING logsecond 20;
UPDATE DB CFG FOR cusdb USING logbufsz 32;
UPDATE DB CFG FOR cusdb USING avg_appls 5;
UPDATE DB CFG FOR cusdb USING locktimeout 30;
UPDATE DB CFG FOR cusdb using AUTO_MAINT off;
```

```
CREATE DB jcrdb using codeset UTF-8 territory us PAGESIZE 8192;
UPDATE DB CFG FOR jcrdb USING applheapsz 4096;
UPDATE DB CFG FOR jcrdb USING app_ctl_heap_sz 1024;
UPDATE DB CFG FOR jcrdb USING stmtheap 32768;
UPDATE DB CFG FOR jcrdb USING dbheap 2400;
UPDATE DB CFG FOR jcrdb USING locklist 1000;
UPDATE DB CFG FOR jcrdb USING logfilsiz 4000;
UPDATE DB CFG FOR jcrdb USING logprimary 12;
```

```

UPDATE DB CFG FOR jcrdb USING logsecond 20;
UPDATE DB CFG FOR jcrdb USING logbufsz 32;
UPDATE DB CFG FOR jcrdb USING avg_appls 5;
UPDATE DB CFG FOR jcrdb USING locktimeout 30;
UPDATE DB CFG FOR jcrdb using AUTO_MAINT off;

CREATE DB lmdb using codeset UTF-8 territory us PAGESIZE 8192;
UPDATE DB CFG FOR lmdb USING applheapsz 4096;
UPDATE DB CFG FOR lmdb USING app_ctl_heap_sz 1024;
UPDATE DB CFG FOR lmdb USING stmtheap 32768;
UPDATE DB CFG FOR lmdb USING dbheap 2400;
UPDATE DB CFG FOR lmdb USING locklist 1000;
UPDATE DB CFG FOR lmdb USING logfilsiz 4000;
UPDATE DB CFG FOR lmdb USING logprimary 12;
UPDATE DB CFG FOR lmdb USING logsecond 20;
UPDATE DB CFG FOR lmdb USING logbufsz 32;
UPDATE DB CFG FOR lmdb USING avg_appls 5;
UPDATE DB CFG FOR lmdb USING locktimeout 30;
UPDATE DB CFG FOR lmdb using AUTO_MAINT off;

CREATE DB fdbkdb using codeset UTF-8 territory us PAGESIZE 8192;
UPDATE DB CFG FOR fdbkdb USING applheapsz 4096;
UPDATE DB CFG FOR fdbkdb USING app_ctl_heap_sz 1024;
UPDATE DB CFG FOR fdbkdb USING stmtheap 32768;
UPDATE DB CFG FOR fdbkdb USING dbheap 2400;
UPDATE DB CFG FOR fdbkdb USING locklist 1000;
UPDATE DB CFG FOR fdbkdb USING logfilsiz 4000;
UPDATE DB CFG FOR fdbkdb USING logprimary 12;
UPDATE DB CFG FOR fdbkdb USING logsecond 20;
UPDATE DB CFG FOR fdbkdb USING logbufsz 32;
UPDATE DB CFG FOR fdbkdb USING avg_appls 5;
UPDATE DB CFG FOR fdbkdb USING locktimeout 30;
UPDATE DB CFG FOR fdbkdb using AUTO_MAINT off;

CONNECT TO jcrdb USER db2inst1 USING password;
CREATE BUFFERPOOL ICMLSFREQBP4 SIZE 1000 PAGESIZE 4 K;
CREATE BUFFERPOOL ICMLSVOLATILEBP4 SIZE 8000 PAGESIZE 4 K;

```

```
CREATE BUFFERPOOL ICMLSMMAINBP32 SIZE 8000 PAGESIZE 32 K;
CREATE BUFFERPOOL CMBMAIN4 SIZE 1000 PAGESIZE 4 K;
CREATE REGULAR TABLESPACE ICMLFQ32 PAGESIZE 32 K MANAGED BY SYSTEM USING
('ICMLFQ32') BUFFERPOOL ICMLSMMAINBP32;
CREATE REGULAR TABLESPACE ICMLNF32 PAGESIZE 32 K MANAGED BY SYSTEM USING
('ICMLNF32') BUFFERPOOL ICMLSMMAINBP32;
CREATE REGULAR TABLESPACE ICMVFQ04 PAGESIZE 4 K MANAGED BY SYSTEM USING
('ICMVFQ04') BUFFERPOOL ICMLSVOLATILEBP4;
CREATE REGULAR TABLESPACE ICMSFQ04 PAGESIZE 4 K MANAGED BY SYSTEM USING
('ICMSFQ04') BUFFERPOOL ICMLSFREQBP4;
CREATE REGULAR TABLESPACE CMBINV04 PAGESIZE 4 K MANAGED BY SYSTEM USING
('CMBINV04') BUFFERPOOL CMBMAIN4;
CREATE SYSTEM TEMPORARY TABLESPACE ICMLSSYSTSPACE32 PAGESIZE 32 K MANAGED BY SYSTEM
USING ('icmlssystspace32') BUFFERPOOL ICMLSMMAINBP32;
CREATE SYSTEM TEMPORARY TABLESPACE ICMLSSYSTSPACE4 PAGESIZE 4 K MANAGED BY SYSTEM
USING ('icmlssystspace4') BUFFERPOOL ICMLSVOLATILEBP4;

DISCONNECT jcrdb;
TERMINATE;
```

=====**END COPY HERE**====**DO NOT INCLUDE THIS LINE**=====



## Appendix C – Adding a Vertical Cluster member

After creating your cluster, you may need to add additional members to the cluster. This section will describe how to properly add a vertical cluster member to your cluster.

1. From a command window, navigate to <AppServer root>/profiles/Dmgr01/bin
2. Execute the following command:

```
./startManager.sh
```

3. Once the DMGR is open for e-business, launch a web browser and access the DMGR Administrative Console:

<http://<yourhostname>:9060/ibm/console>

4. Navigate to Servers -> Clusters -> WebSphere Application Server clusters -> *PortalCluster* -> Cluster Members

### WebSphere application server clusters > PortalCluster

Use this page to change the configuration settings for a cluster. A server cluster consists of a group of application servers. If one of the application servers t requests are routed to other members of the cluster.

The screenshot shows the configuration page for the PortalCluster. At the top, there are tabs for 'Runtime', 'Configuration', and 'Local Topology'. Below the tabs, the 'General Properties' section includes a text field for 'Cluster name' containing 'PortalCluster', a dropdown for 'Bounding node group name' set to 'DefaultNodeGroup', and two checkboxes: 'Prefer local' (checked) and 'Enable failover of transaction log recovery' (unchecked). At the bottom of this section are buttons for 'Apply', 'OK', 'Reset', and 'Cancel'. On the right side, there are two sections: 'Cluster messaging' with a link to 'Messaging engines', and 'Additional Properties' with links to 'Cluster members' (highlighted with a red box), 'Backup cluster', 'Endpoint listeners', and 'Security domain'.

5. Click 'New'

6. On the next screen, enter the following information:

Member Name - The new member name (for example WebSphere\_Portal\_3)

**NOTE: Do not use any name that contains a space**

Select Node – Select a node that is part of your cluster

Generate Unique HTTP Ports – Ensure this is checked

Step 1: Create first cluster member

→ Step 2: Create additional cluster members

Step 3: Summary

### Create additional cluster members

Enter information about this new cluster member, and click Add Member to add this cluster member to the member list. A server configuration template is created from the first member, and stored as part of the cluster data. Additional cluster members are copied this template.

\* Member name  
WebSphere\_Portal\_3

Select node  
wpNode1(ND 7.0.0.5)

\* Weight  
2 (0..20)

Generate unique HTTP ports

Add Member

Use the Edit function to edit the properties of a cluster member that is already included in this list. Use the Delete function to remove cluster member from this list. You are not allowed to edit or remove the first cluster member or an already existing cluster member.

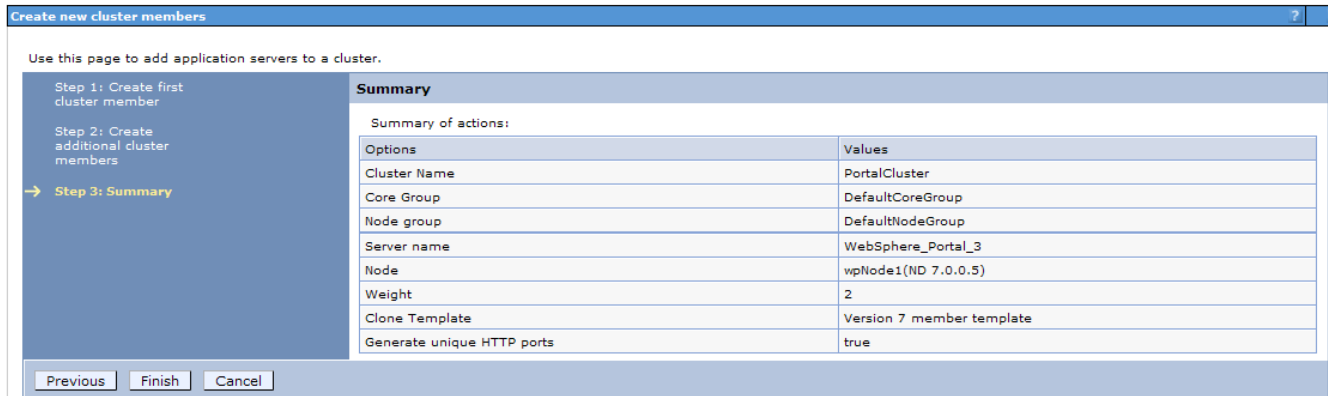
Select	Member name	Nodes	Version	Weight
	WebSphere_Portal	wpNode1	ND 7.0.0.5	2
	WebSphere_Portal_wpNode2	wpNode2	ND 7.0.0.5	2
Total				2

Previous Next Cancel

7. Click “Add Member”

8. Click “Next”

9. Review the summary screen and click Finish.



10. Save the changes

11. Navigate to Server Types -> WebSphere Application Servers -> *WebSphere\_Portal\_3* -> Ports and note the following two port values:

WC\_defaulthost  
WC\_defaulthostsecure

<input type="checkbox"/>	<a href="#">WC_defaulthost</a>	*	10050	<a href="#">View associated transports</a>
<input type="checkbox"/>	<a href="#">WC_defaulthost_secure</a>	*	10053	<a href="#">View associated transports</a>

12. Update the Virtual Hosts to include these two ports if they are not already present:

- a) Navigate to Environment -> Virtual Hosts -> default\_host -> Host Aliases
- b) Click “New”
- c) Set Hostname to \*
- d) Set Port to the value of WC\_defaulthost
- e) Click “OK”
- f) Repeat a-e for WC\_defaulthost\_secure
- g) Save changes

13. Enable Dynamic Replication on the new cluster member.
  - a) Navigate to **Server Types -> WebSphere Application Servers -> WebSphere\_Portal\_3 -> Container Services -> Dynamic Cache Service**
  - b) Set Cache Size to 3000 entries
  - c) Check the Enable Cache Replication Box
  - d) Select “Not Shared” from the Replication Type drop-down menu
  - e) Click “OK” and save changes.

14. From the Portal node that you created the vertical cluster member on, open a terminal window and change directories to the <wp\_profile root>/ConfigEngine directory.

15. Execute the following ConfigEngine script to remove server-scoped entries from the new cluster member:

**IMPORTANT:** Failure to do this step will result in an inoperable vertical cluster member

```
./ConfigEngine.sh cluster-node-config-vertical-cluster-setup  
-DServerName=WebSphere_Portal_3 -DWasPassword=password
```

where ServerName is set to your new vertical cluster member name. In this case, WebSphere\_Portal\_3 is my new vertical cluster member.

16. Synchronize the nodes and restart the DMGR, nodeagents and cluster members.

17. Verify you can access your new cluster member in a URL using the port defined for WC\_defaulthost in step 11:

<http://hostname:10050/wps/portal>

## **Appendix D – Adding a new secondary node to an existing cluster**

You may need to add a new node to your cluster in the future. In this section, we will add a new node to an existing cluster that already has standalone LDAP security enabled.

1. Install your new Portal following the section 'Install the Secondary Portal node'.

**NOTE:** At this point, you have a standalone Portal server using the default VMM federated file registry security.

2. Copy the `wp_security_ids.properties` from the `<wp_profile>/ConfigEngine/config/helpers` directory on your Primary node to the `<wp_profile>/ConfigEngine/config/helpers` directory on the secondary node.
3. From the secondary node, copy the contents of the helper file into the main `wkplc.properties` file by running the following command (all on one line):

```
./ConfigEngine.sh  
-DparentProperties=<wp_profile>/ConfigEngine/config/helpers/wp_security_ids.p  
roperties -DsaveParentProperties=true
```

**NOTE:** If you did not use a helper file when setting up security, then manually update the `standalone.ldap` values in the `wkplc.properties` file to match those of your existing nodes.

4. Ensure the database client is installed and configured on the secondary node. For DB2 with Type 4 drivers, copy the `db2jcc.jar` and `db2jcc_license_cu.jar` files from the DB2 server to some directory on the secondary Portal server.
5. Stop `WebSphere_Portal` and `server1` by executing the following commands from the `<wp_profile root>/bin` directory:

```
./stopServer.sh WebSphere_Portal -user <admin user> -password <admin pwd>  
./stopServer.sh server1 -user <admin user> -password <admin pwd>
```

6. Ensure the DMGR is **STARTED** by running the following command from the `<dmgr_profile>/bin` directory.

```
./startManager.sh
```

7. From the <wp\_profile>/ConfigEngine/properties directory, make a backup of the following files:

wkplc.properties  
wkplc\_dbtype.properties  
wkplc\_comp.properties

8. Copy the wkplc\_comp.properties and wkplc\_dbtype.properties from the primary node to the new node to ensure the same database configuration.

**NOTE:** Ensure that the values for db2.dbLibrary and derby.DbLibrary in wkplc\_dbtype.properties contain valid directory paths for this node.

9. From the <wp\_profile>/ConfigEngine/properties directory, edit the wkplc.properties file and change the following entries:

```
WasUserid=<DMGR User ID>  
WasPassword=<DMGR password>  
PortalAdminPwd=password  
WasRemoteHostName=<fully qualified hostname of DMGR>  
WasSoapPort=<soap port for DMGR; default is 8879>  
PrimaryNode=false  
ClusterName=PortalCluster
```

**NOTE:** Ensure that the value for ClusterName matches the value for ClusterName on the primary node.

10. In a terminal window from the secondary node, change directories to <wp\_profile>/ConfigEngine

11. Add the node to the deployment manager cell by executing the following ConfigEngine script:

```
./ConfigEngine.sh cluster-node-config-pre-federation  
-DWasPassword=<DMGR password>
```

**NOTE:** Ensure that the time on the Deployment Manager server and the time on the primary node are within 5 minutes of each other. Failure to do so can cause this step to fail. This will also create the NodeAgent server for you on your node.

**NOTE:** If you are prompted to accept an SSL certificate, type Y and press Enter to continue

**NOTE:** This guide was written specifically for Portal v6.1.0.3 and higher. If you are using WebSphere Portal v6.1.0.0, 6.1.0.1, or 6.1.0.2 then you **must** specify -DDMgrUserid and -DDMgrPassword parameters when running the cluster-node-config-pre-federation task. For example:

```
./ConfigEngine.sh cluster-node-config-pre-federation -DDMgrUserid=<DMGRUser>  
-DDMgrPassword=<password> -DWasPassword=<local WAS password>
```

12. After the previous step completes, your node will be part of the deployment manager cell. As a result, this node is now using the Deployment Manager cell name.

Edit the <wp\_profile>/ConfigEngine/wkplc.properties file and ensure the following property is set correctly:

```
CellName=<dmgr cell name>
```

13. Update the deployment manager configuration for the new WebSphere Portal server by executing the following ConfigEngine script:

```
./ConfigEngine.sh cluster-node-config-post-federation -DWasPassword=password
```

14. Ensure the NodeAgent is started on this node by running the following command from the <wp\_profile>/bin directory:

```
./startNode.sh
```

15. Specify the name of the future secondary cluster member.

Edit the <wp\_profile>/ConfigEngine/wkplc.properties file and change the following property:

```
ServerName=<name of new cluster member>
```

**NOTE:** When you open the properties file, you should see `WebSphere_Portal_nodename`. You can use this value if you like. Otherwise you can change this to anything EXCEPT 'WebSphere\_Portal'. DO NOT use the value of 'WebSphere\_Portal' for your secondary cluster member.

16. Add this newly federated WebSphere\_Portal server as a cluster member to the existing cluster by executing the following ConfigEngine script:

```
./ConfigEngine.sh cluster-node-config-cluster-setup -DwasPassword=<password>
```

**NOTE:** This will automatically add a secondary cluster member to your existing cluster based on whatever value you set for ServerName in step 10. In this example, the default value was used. The node name is wnode3 so our cluster member will be called `WebSphere_Portal_wnode3`.



19. Allow **30 minutes** for ear expansion to complete on the secondary node. Failure to do so may result in several applications being unavailable on this node.
20. Because the security configuration for the Portal node changed when we federated the node, we need to update the Portal configuration to reference the new Portal Admin ID and group by running the following ConfigEngine script:

```
./ConfigEngine.sh wp-change-portal-admin-user -DnewAdminId=<full DN of Portal  
admin ID> -DnewAdminPwd=<new password> -DnewAdminGroupId=<full DN of Portal  
Admin Group ID> -Dskip.ldap.validation=true
```

**Note:** The `-Dskip.ldap.validation=true` flag can be used if the script fails during ldap validation.

21. Start the new cluster member `WebSphere_Portal_nodename` from the `<wp_profile>/bin` directory of the new node:

```
./startServer.sh WebSphere_Portal_nodename
```

22. Verify you can access your new cluster member in a URL:

`http://hostname:10050/wps/portal`

## Appendix E – Running IBM Support Assistant Lite

At some point you may run into a failure when executing WebSphere Portal and require assistance from IBM Remote Technical Support. In order to save time with troubleshooting your issue, IBM Support strongly recommends you use ISALite to collect the logs and configuration information from your system.

These instructions assume you already installed the ISALite tool from earlier in this guide.

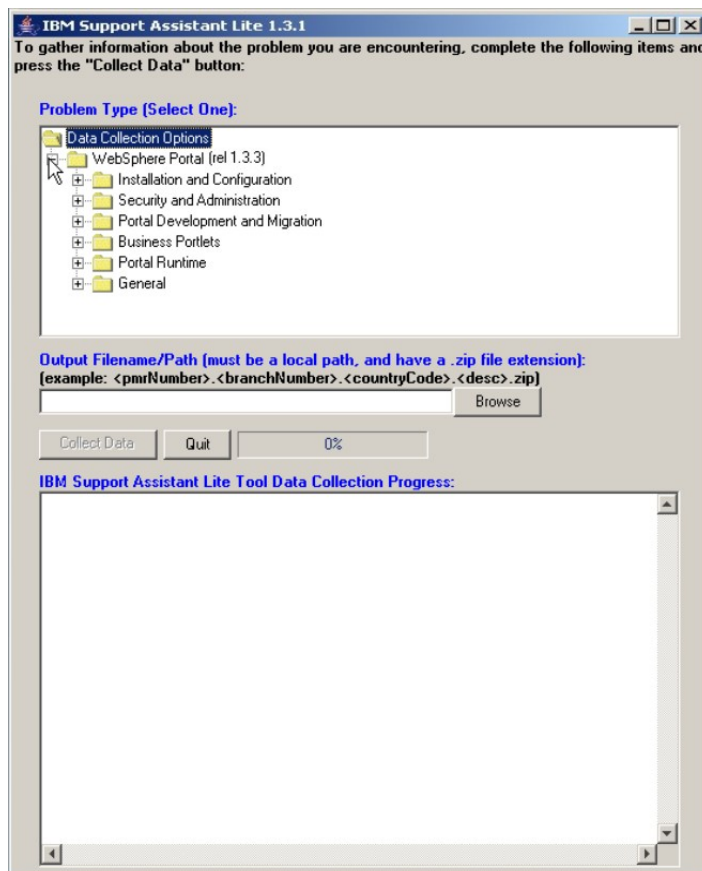
To watch a video demonstration of the tool, please visit the WebSphere Portal Wiki at this URL:

<http://www-10.lotus.com/ldd/portalwiki.nsf/dx/demo-isalite-fundamentals-version-1.3.3-for-ibm-websphere-portal>

1. Open a command prompt and change directories to <wp\_profile root>/PortalServer/ISALite.
2. Launch the tool by executing the following command:

```
runISALite.bat
```

3. When the tool launches, you should see a window similar to the following:



4. Expand WebSphere Portal and select your problem type. If you are unsure of what your problem type is, select one of the following:

**WebSphere Portal -> General -> Portal General Problem**

**WebSphere Portal -> General -> Portal General File Collection**

5. In the Output Filename field, specify the path and name of the zip file that will be created by the tool. If you have a PMR number, please use include this number in the zip name. For example:

<C:/temp/12345.123.000.PortalProblem.zip>

6. Click the button for “Collect Data”
7. You will receive several prompts as the script runs. Answer all questions you see as accurately as possible. This includes PortalServer and AppServer root, WAS credentials, and whether or not the server is part of a cluster.

**Note:** If you selected the “**Portal General File Collection**” problem type, you will not see these prompts. This option is only available in versions 1.3.3 and higher, and requires minimal user interaction.

8. Select to FTP the logs when prompted. If you choose not to do so here or are unable to do this, you can do so manually following the instructions in this link:

<http://www-01.ibm.com/support/docview.wss?rs=688&uid=swg21201571>

## Appendix F – Common Problems

This section will discuss common problems encountered when building clusters and the troubleshooting steps and/or resolutions to them.

**PROBLEM:** All or any ConfigEngine script fails immediately with exceptions such as:

```
Created config Service Proxy:
com.ibm.websphere.management.configservice.ConfigServiceProxy@36413641
CELL: dmgrCell
NODE: wpNode1
com.ibm.websphere.management.exception.ConfigServiceException:
javax.management.JMRuntimeException: ADMN0022E: Access is denied for the resolve
operation on ConfigService MBean because of insufficient or empty credentials.
.
.
.
Registry could not be loaded from WAS using current connection information.
Please verify your WAS connection properties and retry the operation. Current
input:
WasUserId: uid=wpadmin,cn=users,dc=ibm,o=com
WasPassword: PASSWORD_REMOVED
WasRemoteHostName: myDmgrServer.raleigh.ibm.com
WasSoapPort: 8879
CellName: dmgrCell
NodeName: wpNode1
```

**CAUSE:** The ConfigEngine registry is stored within the WAS configuration. Each time ConfigEngine is executed, it must connect to the WAS configuration and retrieve the registry. This depends on the following:

1. The following values being correct in wkplc.properties:

```
WasUserId
WasPassword
WasRemoteHostName
WasSoapPort
CellName
NodeName
```

So that the following sentence would be valid:

“Connect to the WasRemoteHostName at WasSoapPort using WasUserid and WasPassword to verify the CellName and NodeName are valid.”

2. If in a cluster, the DMGR must be running.

**RESOLUTION:** Ensure the properties listed in the “Cause” are correct, and if in a cluster ensure the DMGR is running. After you have done this, execute the ConfigEngine task again.

**PROBLEM:** After creating a cluster with my primary node, I cannot access Portal in a web browser. I see the following message in the web browser:

```
A WebGroup/Virtual Host to handle /wps/portal has not been defined.
```

or

```
Error 404: No target servlet configured for uri: /wps/portal
```

**CAUSE:** The cluster-node-config-pre-federation script failed at some point, and the Deployment Manager was not properly cleaned up before executing the script again. Failure to remove the enterprise applications from the DMGR will cause the target mappings between the application and the WebSphere\_Portal server to become broken.

**RESOLUTION:** Remove the node from the cluster and perform the following steps in the DMGR:

- Remove all Enterprise applications
- Remove the WebSphere\_Portal server definition
- Remove the JDBC Provider information for WebSphere\_Portal
- Start with cluster-node-config-pre-federation and add the node back to the DMGR.

**PROBLEM:** 'cluster-node-config-pre-federation' fails with invalid credentials when stopping all servers:

```
=====
action-cluster-node-federation:
  [echo] Federating node 'WebSphere_Portal' to 'mydmgr.raleigh.ibm.com' using
port '8879'
  [echo] Calling addNode with the following credentials : -username wpadmin
-password (PasswordRemoved) -includeapps
  [exec] ADMU0116I: Tool information is being logged in file
  [exec] C:\WebSphere\wp_profile\logs\addNode.log
  [exec] ADMU0128I: Starting tool with the wp_profile profile
  [exec] CWPKI0309I: All signers from remote keystore already exist in local
keystore.
  [exec] ADMU0001I: Begin federation of node node1 with Deployment Manager at
  [exec] mydmgr.raleigh.ibm.com:8879.
  [exec] ADMU0001I: Begin federation of node node1 with Deployment Manager at
  [exec] mydmgr.raleigh.ibm.com:8879.
  [exec] ADMU0009I: Successfully connected to Deployment Manager Server:
  [exec] mydmgr.raleigh.ibm.com:8879
  [exec] ADMU0505I: Servers found in configuration:
  [exec] ADMU0506I: Server name: server1
  [exec] ADMU0506I: Server name: WebSphere_Portal
  [exec] ADMU2010I: Stopping all server processes for node node1
  [exec] ADMU0027E: An error occurred during federation ADMN0022E: Access is
denied for
  [exec] the stop operation on Server MBean because of insufficient
or empty
  [exec] credentials.; rolling back to original configuration.
  [exec] ADMU0211I: Error details may be seen in the file:
  [exec] C:\WebSphere\wp_profile\logs\addNode.log
  [exec] ADMU0113E: Program exiting with error:
  [exec] com.ibm.websphere.management.exception.AdminException:
  [exec] javax.management.JMRuntimeException: ADMN0022E: Access is
denied for
  [exec] the stop operation on Server MBean because of insufficient
or empty
  [exec] credentials., resulting from: ADMN0022E: Access is denied
for the
  [exec] stop operation on Server MBean because of insufficient or
empty
  [exec] credentials.
  [exec] ADMU4113E: Verify that username and password information is correct.
If
  [exec] running tool from the command line, pass in the correct
-username
  [exec] and -password. Alternatively, update the
<conntype>.client.props
  [exec] file.
  [exec] ADMU1211I: To obtain a full trace of the failure, use the -trace
option.
  [exec] ADMU0211I: Error details may be seen in the file:
  [exec] C:\WebSphere\wp_profile\logs\addNode.log
Target finished: action-cluster-node-federation
Target finished: cluster-node-config-pre-federation
=====
```

**CAUSE:** This exception will occur during the cluster-node-config-pre-federation script if the following two conditions are met:

- The DMGR credentials are different from the local WAS credentials
- The servers on the node (WebSphere\_Portal and server1) are running

**RESOLUTION:** To resolve this issue, you must manually stop WebSphere Portal and server on the node prior to running this script.

**PROBLEM:** When I execute 'cluster-node-config-pre-federation' a second time, the ConfigEngine script fails immediately with this exception:

```
=====
RegistrySynchronized: false
Registry out of sync with WebSphere... synchronizing...
[12/07/09 16:23:18.459 CET] ssl.default.password.in.use.CWPKI0041W
[12/07/09 16:23:18.615 CET] ssl.disable.url.hostname.verification.CWPKI0027I
Created admin client: com.ibm.ws.management.AdminClientImpl@37283728
Created config Service Proxy:
com.ibm.websphere.management.configservice.ConfigServiceProxy@43744374
CELL: Nodotest
NODE: Nodotest
CELL: Nodotest
java.lang.ArrayIndexOutOfBoundsException: Array index out of range: 0
at
com.ibm.wkplc.was.registry.AdminConfigRegistry.createNewRegistry(AdminConfigRegistry.java:230)
at
com.ibm.wkplc.models.compregistry.ResourceWidget.saveResourceToAdminConfig(ResourceWidget.java:360)
.Local registry is out of the sync with the application server. Synchronization
must be performed before any configuration can continue.
Please verify your WAS connection properties and retry the operation. Current
input:
WasUserId: wasadmin
WasPassword: PASSWORD_REMOVED
WasRemoteHostName: mydmgr.ibm.com
WasSoapPort: 8879
CellName: myNode
NodeName: myNode
=====
```

**CAUSE:** This exception occurs because of a unique condition within the cluster-node-config-pre-federation script. At the end of every ConfigEngine script, an attempt is made to synchronize the ConfigEngine's registry with the WAS configuration. If cluster-node-config-pre-federation fails, this attempt to synchronize the registry at the end fails as well because of a mismatch between the WasRemoteHostName and CellName properties in wkplc.properties. On any subsequent ConfigEngine script, an attempt to synchronize the registry is made at the very beginning. The mismatch between the WasRemoteHostName and CellName still exists as it is needed for cluster-node-config-pre-federation, so the synchronization continues to fail.

**RESOLUTION:** To resolve this problem, you must run the following ConfigEngine script before executing cluster-node-config-pre-federation:

```
./ConfigEngine.sh -DWasRemoteHostName=<standalone host> -DWasSoapPort=<standalone soap port>
```

The interim fix PM02927 was created to address this issue, but it will only prevent the error from occurring again. If you have you not applied PM02927 and you hit this error, you must still run the aforementioned ConfigEngine script, then apply PM02927.



**PROBLEM:** After creating my vertical cluster member, I cannot access it in a web browser. I see '404 Initialization of one or more services failed.' In the SystemOut.log, the following exception can be seen referencing my old ID:

```
EJPF0016E: Initialization of service failed.  
com.ibm.wps.ac.DomainAdministratorNotFoundException: EJPSB0107E:  
Exception occurred while retrieving the identity of the domain admin  
user/admingroup uid=wpsadmin,o=defaultWIMFileBasedRealm.
```

**CAUSE:** This exception occurs if the 'cluster-node-config-vertical-cluster-setup' script was not executed correctly, if at all.

When you initially create a cluster, a server template is made of the first cluster member (in our case the WebSphere\_Portal server from the primary node), including all of its resources. At the time, this server has all of its resources stored at the server scope. One of these resources is a Resource Environment Provider called WP AccessControlDataManagementService. This is where the Portal Administrator ID is stored. After the cluster is created, these resources are copied to the cluster scope. If you were following the steps of this guide, at the time this is done the ID is uid=wpsadmin,o=defaultWIMFileBasedRealm. 'cluster-node-config-cluster-setup' subsequently removes any leftover resources from the server scope.

When you run the Portal ConfigEngine scripts to change security, the ID is updated in WP AccessControlDataManagementService at the cluster scope.

When you create a new vertical cluster member, a new server is created based on the aforementioned template. This results in a new server that has resources defined at the server scope, matching the resources that existed at the time the cluster was initially created. In other words, you have a new cluster member that has WP AccessControlDataManagementService defined at the server scope AND at the cluster scope. The cluster scope has the correct LDAP ID; the server scope has the original ID. The server scope is what is being read when this cluster member starts up.

The cluster-node-config-vertical-cluster-setup script removes all of the server scoped resources from the vertical cluster member.

**RESOLUTION:** Ensure you have correctly executed the cluster-node-config-vertical-cluster-setup and passed in the correct ServerName:

```
./ConfigEngine.sh cluster-node-config-vertical-cluster-setup -DServerName=<new  
member name> -DWasPassword=<password>
```

This command is **case-sensitive**. -DserverName is NOT the same as -DServerName.

**PROBLEM:** After updating security in the cluster, secondary nodes throw User ID exceptions for the JCR application during startup:

```
=====
[11/16/09 10:59:48:553 EST] 00000030 ApplicationMg A   WSVR0200I: Starting
application: jcrear
.
[11/16/09 10:59:50:161 EST] 00000030 servlet      E
com.ibm.ws.webcontainer.servlet.ServletWrapper init SRVE0100E: Uncaught init()
exception created by servlet InitServlet in application jcrear:
javax.servlet.ServletException: javax.jcr.RepositoryException: Unable to initialize
RepositoryFactory due to exception of type: java.security.PrivilegedActionException
with message: com.ibm.wps.um.exceptions.impl.MemberNotFoundExceptionImpl:
com.ibm.portal.puma.MemberNotFoundException: EJPSG0002E: Requested Member does not
exist.uid=wpadmin,o=defaultWIMFileBasedRealm/null.
    at com.ibm.icm.jcr.init.InitServlet.init(InitServlet.java:57)
    at javax.servlet.GenericServlet.init(GenericServlet.java:241)
.
Caused by: javax.jcr.RepositoryException: Unable to initialize RepositoryFactory
due to exception of type: java.security.PrivilegedActionException with message:
com.ibm.wps.um.exceptions.impl.MemberNotFoundExceptionImpl:
com.ibm.portal.puma.MemberNotFoundException: EJPSG0002E: Requested Member does not
exist.uid=wpadmin,o=defaultWIMFileBasedRealm/null.
    at com.ibm.icm.jcr.RepositoryFactory.init(RepositoryFactory.java:308)
    at
com.ibm.icm.jcr.RepositoryFactory.getRepository(RepositoryFactory.java:660)
=====
```

**CAUSE:** There is an additional ConfigEngine step that needs to be executed on secondary nodes after enabling security in a cluster:

```
./ConfigEngine.sh enable-jcr-security
```

Failure to run this step will result in this issue for all secondary nodes.

**RESOLUTION:** Complete the following steps to correct this issue:

1. Copy the security properties in `wkplc.properties` from the primary node to all secondary nodes
2. Execute the following ConfigEngine script to complete the security configuration:

```
./ConfigEngine.sh enable-jcr-security -DWasPassword=<password>
```

**NOTE:** This only applies to v6.1.0.3 or higher. In lower 6.1.0.x versions, you must run a different script on the secondary nodes:

```
./ConfigEngine.sh wp-change-portal-admin-user -DnewAdminId=<full DN of Portal  
admin ID> -DnewAdminPwd=<new password> -DnewAdminGroupId=<full DN of Portal  
Admin Group ID> -Dskip.ldap.validation=true
```