



Phillip Chan
Scott Loveland
Theresa Tai

WebSphere Portal Server Cloning on Linux for zSeries and z9

Introduction

Installing WebSphere® Portal can be a time consuming process. There is a great deal of software involved and even though the actual installation process is largely automated, the elapsed time required to simply lay it all down can be significant. For availability and throughput reasons, it is also common to go beyond a single instance of WebSphere Portal by clustering several of them together. However, the additional installations required to complete the cluster act as a multiplier, driving elapsed time quite high. Wouldn't it be more efficient to go through the installation process once, then clone that instance as many times as required across the cluster? We think so, and this paper reveals an approach for doing just that.

To put the cloning phase in context, we show the entire process from beginning to end. We start by walking through a couple of sample installations—first a default (recommended) install and then a custom install for those who need to follow that path. Then we review the cloning process, including some XML and bash shell scripts to automate much of the busy work. Next we cluster the WebSphere Portal images together using WebSphere Application Server Network Deployment. Finally, we migrate the datasource from CloudScape to DB2® on either Linux or z/OS®. The end result is a fully functional and serviceable WebSphere Portal cluster deployment on Linux for System z™.

This IBM Redpaper is intended for clients that are new to Portal Server on Linux for zSeries and for new installations of Portal environment. The intent is to come up with a working (pre-production) Portal environment with more than a handful of Portal instances—we want to provide zSeries clients with a better end-user experience with quick deployment to allow users the opportunity to explore the rich set of Portal functions/features.

This Redpaper is not intended for on-Demand rapid deployment; however, following is a link to a developerWorks article on that topic as the next step for more sophisticated Portal deployments:

<http://www-128.ibm.com/developerworks/autonomic/library/ac-odp3.html>

We do not cover details on LDAP setup and configuration in this paper because that is a post install task; however, it is important to note that if you choose LDAP for Portal user authentication, please see the WebSphere InfoCenter for details as WebSphere Application Server provides the user authentication function.

Overview of WebSphere Portal

WebSphere Portal for Multiplatforms allows companies to build scalable and reliable custom portals at a single, central location for personalized interaction with applications, content, processes, and people. WebSphere Portal for Multiplatforms includes the following two offerings:

- ▶ **Portal Enable** is the base offering and provides personalization, content and document management, search, and workflow functions along with the scalable portal framework.
- ▶ **Portal Extend** adds powerful collaborative, extended search and web analysis features to enhance portal effectiveness.

We used WebSphere Portal Extend for Multiplatforms for our testing. WebSphere Portal requires WebSphere Application Server and WebSphere Business Integration Server Foundation. The WebSphere Portal installation program detects and verifies if there is an installed version of WebSphere Application Server. Our installation occurred on new SLES 9 64-bit images (kernel 2.6.5-7.97-s390x), all running under z/VM® 5.2 within a single zSeries® z990 LPAR.

From this point on, we use the following reference:

- ▶ WebSphere Business Integration Server Foundation as WBISF

The Linux file system structure for our WebSphere Portal images was spread across both ECKD™ and FCP SCSI disks on an Enterprise Storage Server® (ESS). The /opt directory by default is where WebSphere installs its products, and so it needs to be rather large (minimum of 2.5 GB). The WebSphere Info Center recommends that you carefully plan the size of the file system and provides the following disk space requirements for WebSphere Portal:

- ▶ Make the / (root) file system 1.5 GB or more in size.
- ▶ Make the /opt file system 2.5 GB or more in size. The default directory to install WebSphere Portal on Linux is /opt, which you can change to any directory you like later. By default, /opt is mounted under the file system. If you choose to install WebSphere Portal under /usr, 3.5 GB or more is recommended.
- ▶ Make the /home file system 500 MB or more in size.

We chose to place the /opt directory in its own file system (/opt) on a single SCSI LUN. The rest of our WebSphere Portal Linux image resided on an ECKD disk. This structure is depicted in Table 1.

Table 1 File systems used in this example

Directory	Disk Architecture	FS Type	Disk Space
/	ECKD	ext3	2.3 GB
/opt	SCSI	ext3	9.2 GB

Note: The following output shows the disk usage with WebSphere Portal already installed.

```

litwps01:~ # df -h -T
Filesystem      Type      Size  Used Avail Use% Mounted on
/dev/dasda1     ext3      2.3G  1.6G  592M  74% /
/dev/sda1       ext3      9.2G  6.7G  2.1G  77% /opt

```

Figure 1 and Figure 2 on page 4 depict our target environments. Figure 1 shows the environment after we transfer the database from Cloudscape™ to DB2 on Linux for zSeries or System z.

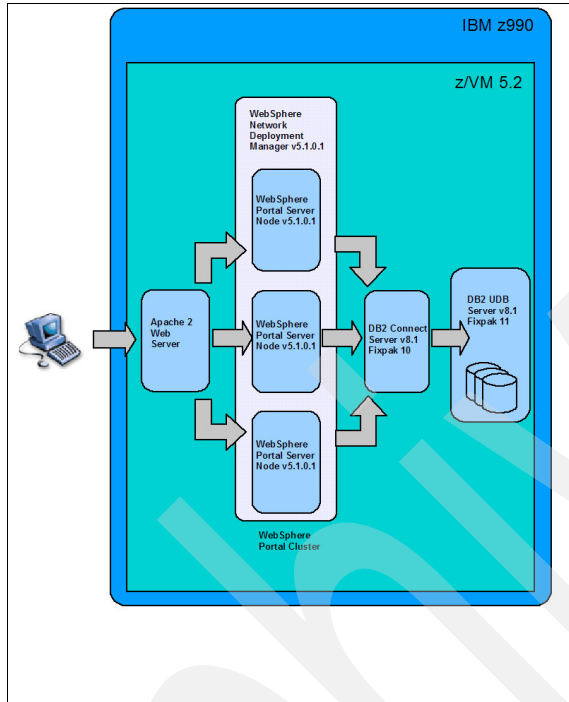


Figure 1 An environment using DB2 running on Linux for zSeries

Figure 2 on page 4 shows our alternative environment, after we transfer the database from Cloudscape to DB2 on z/OS.

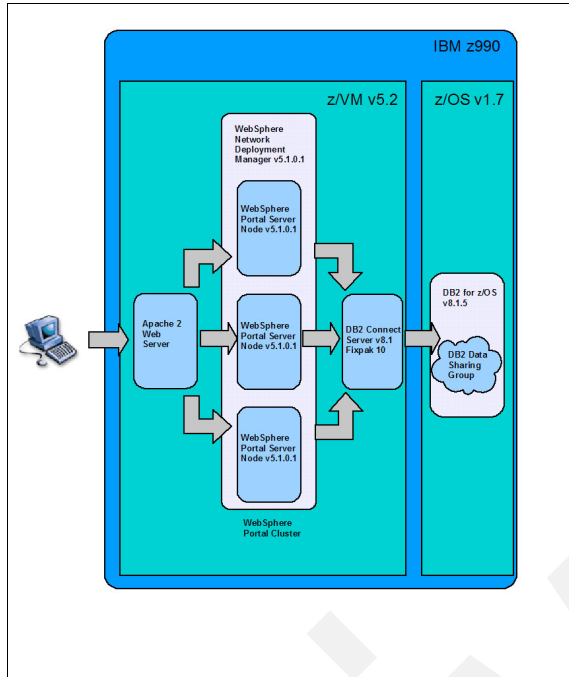


Figure 2 An environment using DB2 running on z/OS

The flow of the overall installation and configuration process is as follows:

1. Installing WebSphere Portal
 - a. Full Default Installation
 - b. Customer Installation
2. Configuring and Cloning WebSphere Portal images
 - a. Cloning using WebSphere's XML approach
 - b. Our custom bash scripts
3. Clustering WebSphere Portal images
4. Migrating datasource from Cloudscape to external database
 - a. DB2 UDB on Linux
 - b. DB2 for z/OS

The following sections will describe each step in this flow.

Installing the first WebSphere Portal instance

Before you can clone WebSphere Portal, at least one instance of it must be installed somewhere. There are various installation paths to follow. We tried two that we describe in this Redpaper. In the first path, which we recommend, we did a full install of WebSphere Portal v5.1.0.1, and let it control the installation of WebSphere Application Server and WBISF for us. In the second (and more complex) path, we did a custom install, handling the installation of each piece separately.

In both cases, we downloaded all our product software from an internal code server, where it was packaged as tar or zip files. The WebSphere Portal installation process periodically calls out for the product libraries it needs by requesting particular product CDs. Moving these libraries from a code server (or from the CDs themselves) to the install image can either be

done in advance of starting the install or via a second ssh session during the install as the CDs are requested by the installation wizard. For each path we took, we list the files used.

Note on 64-bit issue: WebSphere Portal 5.1.0.1 has an issue that erroneously prevents its installation from proceeding if it detects that it is running on a 64-bit Linux image. A technote describes the bypass for this problem, and should be followed exactly. The technote can be found at the following Web address:

<http://d02was006.southbury.ibm.com/support/dcf/preview.wss?host=d02dbs88.southbury.ibm.com&db=support/swg/swgtech.nsf&unid=874D339B13C6345F852570690072EE81&taXOC=SSCPKQ9&MD=2005/08/29%2013:08:01&sid=>

Full installation

For the full WebSphere Portal installation, the required CDs and packages are listed in Table 2.

Table 2 CDs and packages required for a full WebSphere Portal installation

WebSphere Portal CD	File name	Directory used on our install image	Setup CD title
Setup	C82XSML.zip	/opt/wpscd1	<i>WebSphere Portal v5.1.0.1 - Portal Install (Setup), v5.1.0.1 Multilingual</i>
CD2	C82Y1ML.zip	/opt/wpscd2	<i>WebSphere Portal v5.1.0.1 - Portal Server, v5.1.0.1 (2) Multilingual</i>
CD3	C82Y2ML.zip	/opt/wpscd3	<i>WebSphere Portal v5.1.0.1 - Portal Server, v5.1.0.1 (3) Multilingual</i>
CD1-11	C814RML.taz	/opt/wps/CD1-11	<i>WebSphere Portal v5.1 - Certain Components of WebSphere Business Integration Server Foundation for Linux on zSeries, V5.1 (1-11) Multilingual</i>
CD1-12	C814SML.taz	/opt/wps/CD1-12	<i>WebSphere Portal v5.1 - Certain Components of WebSphere Business Integration Server Foundation for Linux on zSeries, V5.1 (1-12) Multilingual</i>

WebSphere Portal CD	File name	Directory used on our install image	Setup CD title
CD4-6	C82WJML.taz	/opt/wps/CD4-6	<i>WebSphere Portal v5.1.0.1 - WebSphere Application Server Archive Install for Linux on zSeries, v5.1.1.3 (4-6) Multilingual</i>
CD5-2	C82WMML.taz	/opt/wps/CD5-2	<i>WebSphere Portal v5.1.0.1 - WebSphere Portal Archive Install for Unix, v5.1.0.1 (5-2) Multilingual</i>
CD5-4	C82WPML.taz	/op/wps/CD5-4	<i>WebSphere Portal v5.1.0.1 - WebSphere Portal Archive Install for Unix, v5.1.0.1 (5-4) Multilingual</i>

The Linux system for our install was running SUSE SLES 9, 64-bit. We began the install process by defining the following user and groups needed for the embedded messaging feature in WebSphere Application Server:

```
groupadd mqm
groupadd mqbrkrs
useradd mqm -g mqm
usermod -g root -G mqm,mqbrkrs root
```

We logged off and on so that the new credentials would take effect. We checked that root had the correct credentials:

```
$> id
uid=0(root) gid=0(root) groups=0(root),1000(mqm),1001(mqbrkrs)
```

Note: If you do not define these user and groups prior to executing the installation wizard, you will receive the following error message from the install console:

```
WebSphere Application Server embedded messaging prerequisite check failed
EJPIC0212E: Unable to locate group or user
Refer to the log file /tmp/mqcheck.log for additional failure details
Refer to the WPS Infocenter for instructions to update your machine for
embedded messaging. If system environment updates are required, restart the
install after making the changes
```

The /tmp/mqcheck.log file would show the following errors in relation to the missing credentials:

```
wmsetup: 13Dec05 13:55:19 ERROR: Group "mqm" exists, but id "root"
wmsetup: 13Dec05 13:55:19 is not defined as a member of that group.
wmsetup: 13Dec05 13:55:19 Current group membership is :
wmsetup: 13Dec05 13:55:19 uid=0(root) gid=0(root) groups=0(root)
wmsetup: 13Dec05 13:55:19
wmsetup: 13Dec05 13:55:19 ... RC 8 from Check_root
wmsetup: 13Dec05 13:55:19 ERROR: User "root" not in group "mqm"
wmsetup: 13Dec05 13:55:19 Check_root mqbrkrs
wmsetup: 13Dec05 13:55:20 Checking for group "mqbrkrs" ...
wmsetup: 13Dec05 13:55:20 Checking if user "root" is in group "mqbrkrs"
wmsetup: 13Dec05 13:55:20
wmsetup: 13Dec05 13:55:20 ERROR: Group "mqbrkrs" exists, but id "root"
wmsetup: 13Dec05 13:55:20 is not defined as a member of that group.
wmsetup: 13Dec05 13:55:20 Current group membership is :
wmsetup: 13Dec05 13:55:20 uid=0(root) gid=0(root) groups=0(root)
wmsetup: 13Dec05 13:55:20
wmsetup: 13Dec05 13:55:20 ... RC 8 from Check_root
wmsetup: 13Dec05 13:55:20 ERROR: User "root" not in group "mqbrkrs"
```

There are also special considerations when installing WebSphere Portal on 64-bit systems. We followed the 64-bit installation technote described earlier. This technote includes simple pre and post install commands to issue. The pre-install commands we issued on our WebSphere Portal installation image were:

```
$> s390 bash
$> uname -m
s390
```

The latter command showed "s390" as its result, which is sufficient to make WebSphere Portal believe it is installing on a 31-bit system, so we proceeded.

We exported the display of our WebSphere Portal install image to a desktop machine. Then from the /opt/wpscd1 directory we issued `./install.sh` to begin the installation wizard.

We clicked English, and then Next to skip installing the info center. We accepted the license agreement, then chose Full Install.

The installation wizard runs until it prompts for the next CD. We SCP'd files from our code server over to the install image as they were called out by the wizard and unpacked them, as noted in Table 2 on page 5.

We received the following message from the Wizard:

Install Successful! WPS is listening on port 9081. It is available at <http://localhost:9081/wps/portal>

At this point, shutdown WebSphere Portal, and run the post-install steps described in the technote for the 64-bit installation bypass. From the Linux command line on our SLES 9 install image, we performed the following steps:

1. Exit from the bash shell by executing the following command:
`exit`
2. Shut down WebSphere Portal using the following command:
`/opt/WebSphere/AppServer/bin/stopServer.sh WebSphere_Portal`
3. Copy the 31-bit executables to a directory for 64-bit executables:
`cd /WebSphere/PortalServer/IBMTrans/bin`
`cp -pr s390_Linux/* s390x_Linux`
4. Start WebSphere Portal using the following command:
`/opt/WebSphere/AppServer/bin/startServer.sh WebSphere_Portal`

WebSphere Portal started cleanly. We verified the installation as described in “Installation verification” on page 9.

Custom installation

In this path we did a custom install, handling the installation of each piece separately. Following is the order of installation:

1. Install and apply refresh/fixes to WebSphere Application Server v5.1.1.4.

In this step, we install and apply refreshes/fixes to bring to WebSphere Application Server level v5.1.1.4. We followed the instructions in the InfoCenter link located at the following Web address:

http://publib.boulder.ibm.com/infocenter/wasinfo/v5r1//index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tins_epinst.html

2. Install the base version of WBISF

Next, install WBISF v5.1 (the base version). We secure copy (using the `scp` command) the C814RML.taz file from our code server to the install image. Expand the contents of tar file, and run the installation wizard (using the `linux390/install` command in the expanded directory). In the installation wizard, select the following two options:

- **Add to the existing copy of WebSphere Application Server v5.1**
- **Typical Installation**

Note: During installation, the wizard prompts for CD 2 of WBISF. Expand the contents of C814SML.taz file to the temporary WAS_update directory provided by the wizard.

Toward the end of installation, the wizard prompts for CD 1. Select option **Next** to allow installation to proceed. At completion, the wizard warns that a higher level of WebSphere Application Server than that required by WBISF is installed. Select **Finish** to ignore this.

3. Apply refresh/fixes to bring WBISF to level v5.1.1.2.

Next, apply fixes to bring WBISF to level v5.1.1.2. This is a two step process:

a. **Apply fixpack v5.1.1.**

Download the wbisf51_fp1_linux390.tar.gz file from the WBISF support page:

http://www-1.ibm.com/support/docview.wss?rs=1079&context=SS2KS6&dc=D400&uid=swg24007486&loc=en_US&cs=utf-8&lang=en

Apply the fixpack using the `./updateWizard.sh` command. This brings WBISF to v5.1.1.

b. **Apply cumulative fix 2.**

Download the wbisf511_cf2_linux390.zip file from:

http://www-1.ibm.com/support/docview.wss?rs=1079&context=SS2KS6&dc=D400&uid=swg24009803&loc=en_US&cs=utf-8&lang=en

Apply the fixpack using the `./updateWizard.sh` command. This brings WBISF to level v5.1.1.2 (resolving the warning message received during the base WBISF installation).

4. Install WebSphere Portal v5.1.0.1.

Follow the instructions found in InfoCenter at the following Web address:

<http://publib.boulder.ibm.com/pvc/wp/510/ent/en/InfoCenter/index.html>

In InfoCenter, navigate **Installing** → **Installing on Windows/Unix** → **Existing WebSphere Application Server security disabled**. Install WebSphere Portal v5.1.0.1 using the following steps:

- a. Secure copy the C82XSML.zip file from the code server to the installation image.
- b. Expand the contents of the file, and run the installation wizard using the `install.sh` command.
- c. When prompted, select the **Custom Installation** option. This allows you to install over an existing WebSphere Application Server instance. At least two components are installed:
 - WebSphere Portal
 - WebSphere Personalization

Note: During the installation, the wizard prompts for additional CDs:

- ▶ CD2
This is found in the C82Y1ML.zip file and provides the second half of the WebSphere Portal installation.
- ▶ CD3
This is found in the C82Y2ML.zip file and provides the IBM® Workspace Web Content Management and Personalization installation.

Expand these files to the temporary directory and provide that path to the wizard when prompted.

When installation completes successfully, WebSphere Portal starts.

Installation verification

We verified the installation by accessing the home page for our install image at the following Web address:

<http://litwps01.ltic.pok.ibm.com:9081/wps/portal>

Logon using the id/password you defined during the wizard installation. You should see a welcome page similar to Figure 3.

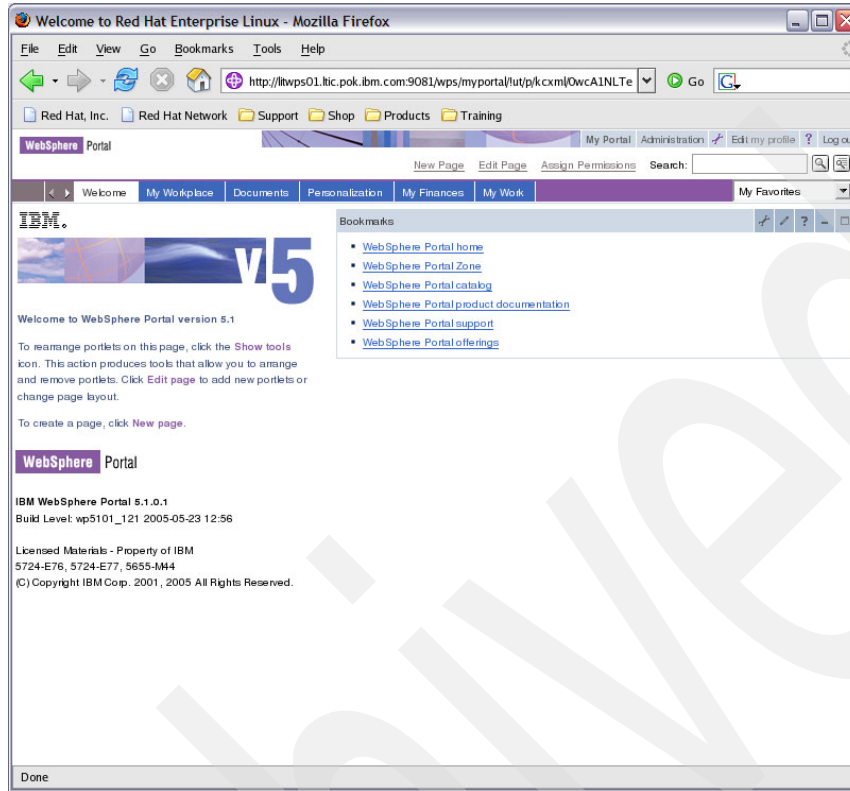


Figure 3 WebSphere Portal welcome page

Cloning

Now that standalone WebSphere Portal v5.1.0.1 server is working, it is time to clone it. We cloned it using the following steps:

1. Copy the WebSphere Portal image.

The base WebSphere Portal Linux™ image is copied to a second Linux image (modifying the necessary Linux system configuration).

2. Modify the WebSphere Portal configuration.

We synchronize the WebSphere Portal configuration on the target system with the host name of the clone.

The cloned stand-alone WebSphere Portal Servers will all have the same GUPID. This globally unique ID is maintained by WebSphere Portal Server in its database (currently in Cloudscape). For a collection of standalone WebSphere Portal Servers to deploy new Portal release content, there must be unique GUPID values for each server. However, for the approach we describe in this paper, the GUPID can remain the same for each member since we cluster the portal servers together.

Copy the WebSphere Portal image

There are many approaches for copying (a.k.a., cloning) a Linux image under z/VM. One approach is described in the June or December 2005 issues of the zSeries Platform Test Report for z/OS and Linux Virtual Servers found at the following Web address:

<http://www-03.ibm.com/servers/eserver/zseries/zos/integtst/>

We describe a simple manual procedure for performing this copying. Other approaches may be used as well. Modification of the WebSphere Portal configuration is independent of the cloning method used.

Note: The approach documented here utilizes Flashcopy2 support provided by an IBM Enterprise Storage Server (ESS) 800. If your DASD does not support flashcopy2, the z/VM DDR function may be used instead.

The base (where WebSphere Portal is installed) and target (the system to clone) configuration is shown in Table 3.

Table 3 Base and target system configuration

	Base system	Target system
VM user	litwps01	litwps02
Hostname	litwps01.ltic.pok.ibm.com	litwps02.ltic.pok.ibm.com
IP address	192.168.71.156	192.168.71.157
ECKD device	5412	5213
FCP device	156	257
LUN	5406000000000000	5517000000000000

The VM directory entry for the target system is shown in Figure 4.

```
USER LITWPS02 TEST4LIT 1536M 2048M GZ
  INCLUDE LINDFLT
  CPU 0
  CPU 1
  CPU 2
  IPL CMS PARM AUTOCR
  MACHINE ESA 3
  DEDICATE 0257 0257
  NICDEF 0700 TYPE QDIO LAN SYSTEM PRVV71
  MDISK 0191 3390 31 2 VM3017 RR ALL SOME FEW
  MDISK 0200 FB-512 V-DISK 2048000 MR
  MDISK 0201 3390 DEVNO 5213 MR ALL ALL AL
```

Figure 4 VM directory entry for target system

Our Linux file system on litwps01 was split across both ECKD and FCP SCSI disks. The ECKD disk can be copied extremely quickly by taking advantage of the ESS 800's Flashcopy2 support. Unfortunately, while the ESS supports Flashcopy of SCSI LUNs, zSeries does not. That means we needed to split up the copying process.

To copy the base WebSphere Portal image, we performed the following steps:

1. Prepare the base Linux image for copying.
2. Copy the SCSI LUN containing the /opt directory.
3. Flashcopy the ECKD disk.
4. Reset the base Linux image to its state prior to the copy.
5. Boot the cloned system and complete its configuration.

Prepare the base Linux image for copying

This step temporarily modifies litwps01's network configuration to match the target system's (litwps02) configuration. We did this so that after the copy litwps02 would have its own unique networking definition.

1. Edit /etc/sysconfig/network/ifcfg-eth0 on litwps01, and change the IP address to 192.168.71.157.
2. Edit /etc/hosts, and change host name and IP to that of litwps02.
3. Edit /etc/HOSTNAME, and change the host name to that of litwps02.

Copy the SCSI LUN containing the /opt directory

We temporarily remove /opt from the file system table on litwps01, just until the flashcopy step is done. Following is what our original fstab looked like:

```
/dev/dasdb1 / reiserfs acl,user_xattr 1 1
/dev/dasda1 swap swap pri=42 0 0
devpts /dev/pts devpts mode=0620,gid=5 0 0
proc /proc proc defaults 0 0
sysfs /sys sysfs noauto 0 0
/dev/sda1 /opt ext3 defaults 1 2
```

Following are the steps we used:

1. Copy /etc/fstab to /etc/orig.fstab (so we can restore it later).
2. Edit /etc/fstab removing /opt.

Next we define litwps02's SCSI LUN to litwps01's SCSI adapter, in order to perform the copy. Note that this LUN (551f) is new and contains no previously defined partitions.

1. Add LUN 551f to adapter 0156:

```
cd /sys/bus/ccw/devices/0.0.0156/0x5005076300c1afc4
echo 0x551f000000000000 > unit_add
```

2. Issue the **dmesg** command to determine the added device name. In our example, the command output is as follows:

```
SCSI device sdb: 19531264 512-byte hdwr sectors (10000 MB)
SCSI device sdb: drive cache: write back
sdb: unknown partition table
Attached scsi disk sdb at scsi0, channel 0, id 1, lun 1
Attached scsi generic sg1 at scsi0, channel 0, id 1, lun 1, type
0
```

The output indicates the LUN is named /dev/sdb.

3. Format the FCP SCSI disk:

```
fdisk /dev/sdb
```

Note: When prompted, provide the following responses:

- ▶ **n** to create a new partition.
- ▶ **p** to create the partition as a primary partition.
- ▶ **Enter** to select the first cylinder of the partition (the default value).
- ▶ **Enter** again to select the last cylinder of the partition (the default value).
- ▶ **w** to write the new partition table to disk.

4. Make a file system on the FCP SCSI disk:

```
mke2fs -j /dev/sdb1
```

5. Mount the FCP SCSI disk to litwps01 at mount point /mnt:

```
mount /dev/sdb1 /mnt
```

6. Copy litwps01's /opt directory to the newly mounted FCP SCSI disk:

```
cd /opt  
cp -pr * /mnt/
```

Note: This operation may take some time to complete.

7. Cleanup by removing litwps02's FCP SCSI disk from litwps01:

```
cd /  
umount /mnt  
cd /sys/bus/ccw/devices/0.0.0156/0x5005076300c1afc4  
echo 0x551f000000000000 > unit_remove
```

Flashcopy the ECKD disk

We performed the following from the MAINT user ID on our z/VM system:

1. Connect MAINT to litwps01's disk.
 - a. link litwps01 201 501 rr
 - i. If desired, the q v da command can be issued to confirm the disk is now linked.
 - b. att 5213 to *

```
DASD 5213 ATTACHED TO MAINT 5213 WITH DEVCTL
```
2. Flashcopy litwps01's disk to the 5213 disk for use by litwps02. At this point, 5213 is already attached to MAINT. Note that by default a VM user requires Class B privileges to issue the flashcopy command. We perform this operation from MAINT. If access to MAINT is limited on your system, then another option is to define a new class, such as Class Z, add the flashcopy command to that class, and then perform the flashcopy from a user with Class Z privileges.
 - a. flashcopy 501 0 end 5213 0 end

```
Command started: FLASHCOPY 501 0 END TO 5213 0 END  
Ready; T=0.01/0.01 11:26:08  
Command complete: FLASHCOPY 501 0 END TO 5213 0 END
```
3. Remove litwps01's disk from MAINT.
 - a. det 501
4. Remove litwps02's disk from MAINT.
 - a. det 5213

5. Next, we grant litwps02 access to our z/VM VSWITCH so it can access our virtual network when booted.
 - a. SET VSWITCH PRVV71 GRANT LITWPS02 VLAN 671

Reset the base Linux image to its state prior to the copy

Back on the litwps01 Linux image we restored files back to their original state.

1. Copy /etc/fstab.orig to /etc/fstab.
2. Edit /etc/sysconfig/network/ifcfg-eth0 to restore litwps01's IP address.
3. Edit /etc/hosts to restore litwps01's IP address and host name.
4. Edit /etc/HOSTNAME to restore litwps01's host name.

Boot the cloned system and complete its configuration

1. We boot up litwps02 and then complete its configuration.
2. First we need to define our copied FCP SCSI disk to Linux. We use Yast2 for this purpose, adding adapter 257, lun = 551f000000000000, WWPN = 0x5005076300c1afc4.
3. Next we exit Yast2. At this point, a lsscsi command shows the disk defined as follows:

```
[0:0:1:0]   disk    IBM      2105800      .115 /dev/sda
```

4. We then mounted that disk at /opt

```
mount /dev/sda1 /opt
```

At this point, a df command shows we had the following configuration:

```
litwps02:~ # df
Filesystem            1K-blocks      Used Available Use% Mounted on
/dev/dasdb1            2403888      1125996   1277892   47% /
tmpfs                  770124         0     770124    0% /dev/shm
/dev/sda1              9611476      3647480   5475756   40% /opt
```

5. Next we update /etc/fstab, adding /opt mount.
6. Finally, we run mkinitrd and zipl:

```
Litwps02:/boot # mkinitrd
Root device:      /dev/dasdb1 (mounted on / as reiserfs)
Module list:      reiserfs dasd_fba_mod dasd_eckd_mod sd_mod zfcplib
Kernel image:    /boot/image-2.6.5-7.97-s390x
Initrd image:    /boot/initrd-2.6.5-7.97-s390x
ln: ~/var/tmp/mkinitrd.Lt1681/mnt/bin/sed': File exists
Shared libs:      lib64/ld-2.3.3.so lib64/libc.so.6 lib64/libselinux.so.1
Modules:          kernel/fs/reiserfs/reiserfs.ko
kernel/drivers/s390/block/dasd_mod.ko kernel/drivers/s390/block/dasd_fba_mod.ko
kernel/drivers/s390/block/dasd_eckd_mod.ko kernel/drivers/scsi/scsi_mod.ko
kernel/drivers/scsi/sd_mod.ko kernel/drivers/s390/cio/qdio.ko
kernel/drivers/s390/scsi/zfcplib.ko
DASDs:            0.0.0200(ECKD) 0.0.0201(ECKD)
zfcplib HBAs:     0.0.0257
zfcplib disks:
                  0.0.0257:0x5005076300c1afc4:0x551f000000000000
initrd updated, zipl needs to update the IPL record before IPL!
Litwps02:/boot # zipl
Using config file '/etc/zipl.conf'
```

```
Building bootmap '/boot/zipl/bootmap'  
Adding IPL section 'ipl' (default)  
Preparing boot device: dasdb (0201).  
Done.
```

7. We then reboot litwps02. At this point, we complete the copying process and have a newly operable Linux image complete with an installed WebSphere Portal server. However, that instance of WebSphere Portal needs some configuration changes before we can start and use it. That is the tricky part and is described in the next section.

Note: You can use the scripts from Appendix A and B to clone the Portal server V5.1 environment with WebSphere Application Server V6.

Modify the WebSphere Portal configuration

We exercised two different options in making changes to cloned WebSphere Portal instances. These options are documented separately. We are not recommending one option over the other as they both deliver the same results. The choice is entirely yours.

Following is the modification procedure summary (with both options):

1. Make changes to host name, cell name, and node name for the WebSphere Portal Server:
 - Option A
 - a. Download "WebSphere Configuration Management Scripts for V5.1.1" from the Online WebSphere Technical Library.
 - b. Select download.
 - c. Unzip the exportImport.zip file to the cloned WebSphere Portal instance.
 - d. Add the new XML code segment (see Appendix A) to the WebSphere Configuration Management script(exportImport.xml) V5.1.1
 - Option B
 - a. Use the custom bash scripts documented in Appendix B.
2. Make changes for the Message Broker and Queue Managers.
 - a. Change the node name reference in the JMS Server.
3. Verify that WebSphere Application Server and Portal Server are cloned successfully.
 - a. Start server1 and Portal server.
 - b. Start WebSphere admin or access the snoop servlet.
 - c. Start WebSphere Portal Server to access the Portal welcome page.

Challenges for changes to WebSphere Application Server and WebSphere Portal servers

Both the WebSphere Application Server and WebSphere Portal are very host name specific. The host name is statically stored in many of its configuration files. It is cumbersome to locate and change every instance. There is a technical paper outlining the procedure in changing the host name in WebSphere Application Server. You can find the paper at the following Web address:

<http://www-1.ibm.com/support/docview.wss?uid=swg27005391&aid=1>

That paper describes the exact files you need to change in WebSphere Application Server. The paper also provides a link to a set of scripts that accomplish most of the tasks in making the configuration changes.

<http://www-106.ibm.com/developerworks/websphere/library/samples/SampleScripts.html>

It does not describe what you need to change in WebSphere Portal; however, the idea remains the same. The following approach extends the methodology described in the WebSphere Application Server paper to cover changes required for all the files needed in the entire WebSphere Portal suite (including WebSphere Application Server). First we give an overview of what is involved, then we explain the changes we made to their set of scripts that help automate the process.

Overview of the procedure

WebSphere Application Server v5.x stores its configuration in XML files in a subdirectory under the main product installation root directory. For example:

```
/opt/WebSphere/AppServer/config/cells/litwps01/nodes/litwps01/servers/WebSphere_Portal/server.xml
/opt/WebSphere/AppServer/config/cells/litwps01/security.xml
/opt/WebSphere/AppServer/config/cells/litwps01/cell.xml
```

WebSphere **Portal Server** v5.x stores its configuration in various files in a subdirectory under its main production installation root directory. For example:

```
/opt/WebSphere/PortalServer/config/work/was/removeServerAttributes.jacl
/opt/WebSphere/PortalServer/config/work/was/wp_portal.properties
/opt/WebSphere/PortalServer/wcm/config/connect.cfg
```

There are a large number of files and directories that you need to modify. You can either manually search and modify the files and directories yourself or use our automated approach. The sample set of scripts that WebSphere provides allows us to change the host name, cell name, node name, and server name in an efficient and timely manner.

Option A: Automating the procedure using WebSphere's approach

We download the 'WebSphere configuration management scripts for v5.1.1' from the Online WebSphere Technical Library Web site - filename: exportImport.zip

<http://www-128.ibm.com/developerworks/websphere/library/samples/SampleScripts.html#download>

Description:

The zip file contains examples to change, export, and import existing configuration data. The following example tasks are provided:

- change host name
- change cell name
- change node name
- change server name
- export/import the whole repository config directory
- export/import individual application server
- export/import enterprise application
- export/import JDBC Provider with associate Data Sources

Setup:

1. Download the exportImport.zip file to the cloned WebSphere Portal image (in our case, litwps02).

2. Unzip exportImport.zip. The files expand to a subdirectory named exportImport.
\$ > unzip exportImport.zip
3. Copy all the files in ../exportImport/* to <WAS_HOME>/bin directory.

The scripts are written to make changes for WebSphere Application Server. We modify the scripts so that changes are made to WebSphere Portal Server as well. The main script to modify is exportImport.xml. We add our own XML content to exportImport.xml.

Note: We added between the following existing XML segments:

- ▶ After “Start changing nodeName Process”
- ▶ Before “Start changing serverName Process”

4. View Appendix A for XML code segment to add.

With the changes made, we now have a script that can update WebSphere Application Server and WebSphere Portal servers with the new host name information. We use the Readme.html provided by the exportImport.zip file to help guide us through the first stages of modifying the host, cell, and node name in WebSphere Application Server.

Change the host name:

Syntax:

```
ws_ant[.sh] -buildfile xxx [-logfile xxx] -DoldHostName=xxx
-DnewHostName=xxx changeHostName
```

We issued:

```
$WAS_HOME/bin > ./ws_ant -buildfile exportImport.xml -logfile changeHostName.log
-DoldHostName=litwps01 -DnewHostName=litwps02 changeHostName
```

Change the cell name:

Syntax:

```
ws_ant[.sh] -buildfile xxx [-logfile xxx] -DoldCellName=xxx
-DnewCellName=xxx changeCellName
```

We issued:

```
$WAS_HOME/bin > ./ws_ant -buildfile exportImport.xml -logfile changeCellName.log
-DoldCellName=litwps01 -DnewCellName=litwps02 changeCellName
```

Change the node name:

An important consideration related to modifying the node name is that WebSphere Embedded Messaging uses the node name to identify the message broker and queue manager processes.

1. In order to change the identity of these processes, destroy and recreate their definition.
2. Run the deletemq utility to remove the message broker and queue managers using the old nodename.

```
$WAS_HOME/bin > ./deletemq.sh litwps01 litwps01 server1
Using: litwps01 litwps01 server1
Issuing: env wempsdeletebroker WAS_litwps01_server1 -w
Purchased processor allowance not set (use setmqcap).
WebSphere MQ queue manager 'WAS_litwps01_server1' started.
BIP8071I: Successful command completion.
rc=0
Issuing: env endmqm -p WAS_litwps01_server1
WebSphere MQ queue manager 'WAS_litwps01_server1' ending.
WebSphere MQ queue manager 'WAS_litwps01_server1' ended.
```

```

rc=0
Issuing: env endmqlsr -w -m WAS_litwps01_server1
      No WebSphere MQ listeners found for the given Queue Manager
rc=0
Issuing: env dltmqm WAS_litwps01_server1
      WebSphere MQ queue manager 'WAS_litwps01_server1' deleted.

```

3. Issue the **dspmq** command to see that the definitions are deleted.
4. Run **createmq** to recreate definitions using the new nodename.

```

$WAS_HOME/bin > # ./createmq.sh /opt/WebSphere/AppServer/ litwps02 litwps02
server1
Using: /opt/WebSphere/AppServer/ litwps02 litwps02 server1
Issuing: env crtmqm -c "WebSphere Application Server queue manager - do not
delete" -lf 512 -ls 60 -u SYSTEM.DEAD.LETTER.QUEUE -oc amqwasoa -om amqwasoa
-os 32 -od port=0,ExitReason=16 WAS_litwps02_server1
      WebSphere MQ queue manager created.
      Creating or replacing default objects for WAS_litwps02_server1.
      Default objects statistics : 31 created. 0 replaced. 0 failed.
      Completing setup.
      Setup completed.
rc=0
Issuing: env strmqm -od port=0,ExitReason=16 WAS_litwps02_server1
      Purchased processor allowance not set (use setmqcap).
      WebSphere MQ queue manager 'WAS_litwps02_server1' started.
rc=0
Issuing: env runmqsc -e WAS_litwps02_server1
Using input from: /opt/WebSphere/AppServer//bin/createmq.mqsc
5724-B41 (C) Copyright IBM Corp. 1994, 2002. ALL RIGHTS RESERVED.
Starting MQSC for queue manager WAS_litwps02_server1.
AMQ8005: WebSphere MQ queue manager changed.
AMQ8006: WebSphere MQ queue created.
AMQ8006: WebSphere MQ queue created.
AMQ8006: WebSphere MQ queue created.
AMQ8014: WebSphere MQ channel created.
AMQ8015: WebSphere MQ channel deleted.
AMQ8015: WebSphere MQ channel deleted.
AMQ8015: WebSphere MQ channel deleted.
AMQ8015: WebSphere MQ channel deleted.
AMQ8015: WebSphere MQ channel deleted.
AMQ8015: WebSphere MQ channel deleted.
AMQ8015: WebSphere MQ channel deleted.
AMQ8015: WebSphere MQ channel deleted.
AMQ8015: WebSphere MQ channel deleted.
AMQ8015: WebSphere MQ channel deleted.
AMQ8015: WebSphere MQ channel deleted.
AMQ8015: WebSphere MQ channel deleted.
AMQ8015: WebSphere MQ channel deleted.
AMQ8015: WebSphere MQ channel deleted.
AMQ8007: WebSphere MQ queue deleted.
AMQ8007: WebSphere MQ queue deleted.
AMQ8007: WebSphere MQ queue deleted.
AMQ8007: WebSphere MQ queue deleted.
AMQ8007: WebSphere MQ queue deleted.
AMQ8007: WebSphere MQ queue deleted.
AMQ8007: WebSphere MQ queue deleted.
AMQ8007: WebSphere MQ queue deleted.
AMQ8007: WebSphere MQ queue deleted.
AMQ8007: WebSphere MQ queue deleted.
AMQ8007: WebSphere MQ queue deleted.
AMQ8007: WebSphere MQ queue deleted.
AMQ8006: WebSphere MQ queue created.
AMQ8006: WebSphere MQ queue created.
AMQ8006: WebSphere MQ queue created.
AMQ8006: WebSphere MQ queue created.
AMQ8006: WebSphere MQ queue created.

```

```

AMQ8006: WebSphere MQ queue created.
AMQ8006: WebSphere MQ queue created.
AMQ8006: WebSphere MQ queue created.
29 MQSC commands read.
No commands have a syntax error.
All valid MQSC commands were processed.

rc=0
Issuing: env wempsdeletebroker WAS_litwps02_server1 -w
rc=0
Issuing: env wempscreatebroker WAS_litwps02_server1 -i dummy -a dummy -q
WAS_litwps02_server1 -n "" -u "" -p "" -b -j
AMQ8110: WebSphere MQ queue manager already exists.
WebSphere MQ queue manager running.

rc=0
Issuing: env wempsservice WAS_litwps02_server1 -r
BrokerUUID=FFFFFFFF-0000-0000-0000-000000000000
rc=0
Issuing: env endmqm -i WAS_litwps02_server1
WebSphere MQ queue manager 'WAS_litwps02_server1' ending.
WebSphere MQ queue manager 'WAS_litwps02_server1' ended.

rc=0

```

5. Issue the **dspmqs** command again to confirm that the definitions were recreated.
6. Change node name on the WebSphere Application Server environment

Syntax:

```

ws_ant[.sh] -buildfile xxx [-logfile xxx] -DoldNodeName=xxx
-DnewNodeName=xxx changeNodeName

```

We issued:

```

$WAS_HOME/bin > ./ws_ant -buildfile exportImport.xml -logfile changeNodeName.log
-DoldNodeName=litwps01 -DnewNodeName=litwps02 changeNodeName

```

Change node name of the WebSphere Portal Server environment

Syntax:

```

ws_ant[.sh] -buildfile xxx [-logfile xxx] -DoldNodeName=xxx
-DnewNodeName=xxx changePortalName

```

We issued:

```

$WAS_HOME/bin > ./ws_ant -buildfile exportImport.xml -logfile changePortalName.log
-DoldNodeName=litwps01 -DnewNodeName=litwps02 changePortalName

```

7. Once the configuration changes are made through the scripts, and the logs show no errors, we verify if WebSphere Application Server and WebSphere Portal servers started cleanly. We started server1 and WebSphere_Portal servers.

```

$ WAS_HOME/bin > ./startServer.sh server1
$ WAS_HOME/bin > ./startServer.sh WebSphere_Portal

```

8. Verify if WebSphere Application Server works by accessing the snoop servlet at the following Web address:

<http://litwps02.ltic.pok.ibm.com:9080/snoop>

9. Verify if WebSphere Portal works by accessing the WebSphere Portal Welcome page at the following Web address:

<http://litwps02.ltic.pok.ibm.com:9081/wps/portal>

Option B: Automating the Procedure using our customized bash script

An alternative to using WebSphere's automated approach with XML files is creating a bash script that basically accomplishes the same tasks. We wrote and tested a home grown bash shell script to make all the changes in /opt/WebSphere. The script has the logic to check if the file is binary or not—we recommend that you not make changes to binary files.

1. Run the script i from /opt/WebSphere.
2. View Appendix B for bash script code.
3. Take the contents and add it to a file.
4. Save the script file to /opt/WebSphere and make sure the file is executable.
5. Execute the script (we called our searchreplace.sh) passing it two arguments—the oldhostname to search for and the newhostname you want to change to.

Syntax:

```
./searchreplace.sh oldhostname newhostname
```

We issued:

```
$ /opt/WebSphere > ./searchreplace.sh litwps01 litwps02
```

All the necessary changes are automatically made from the script. Start WebSphere Application Server and WebSphere Portal as you normally would and verify if it works.

We ran the script against litwps03, and verified WebSphere Portal works as described previously for Option A.

WebSphere Portal cluster setup

We now have three working WebSphere Portal v5.1.0.1 standalone servers from the cloning procedure above. We can cluster them together using WebSphere Application Server Network Deployment v5.1.1.4. We created a brand new SLES 9 64 bit guest with a host name of litwps04.ltic.pok.ibm.com. Visit the following Web site for special instructions on installing WebSphere Application Server V5.1 on SLES9:

http://www-1.ibm.com/support/docview.wss?rs=180&context=SSEQTP&q1=s390x&uid=swg21182138&loc=en_US&cs=utf-8&lang=en

Note: We recommend that you migrate CloudScape to DB2 (transfer data) before setting up the cluster. This is because CloudScape is not supported in a clustered environment. See the section, “Migrate CloudScape to DB2 on Linux for zSeries or DB2 on z/OS” on page 25. In this Redpaper, we focus on cloning the portal server. Our portal servers are installed locally under the same z/VM; therefore, we can set up the cluster before migrating to DB2. However, if you plan to install portal servers on different z/VMs for High Availability, migrate CloudScape to DB2 before you set up the cluster.

We install and apply refreshes/fixes to get to WebSphere Application Server ND v5.1.1.4. Please use the following InfoCenter link to install and apply all refresh/fix packs:

http://publib.boulder.ibm.com/infocenter/wasinfo/v5r1//index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/tins_epinst.html

For Portal environment running with WebSphere Application Server V6, before you run ./addNode.sh, perform the following steps:

1. Change the timeout request for SOAP client from 180 to 6000 on your Portal server and the Deployment Manager
/opt/IBM/WebSphere/AppServer/profiles/<profile name>/properties
Example: opt/IBM/WebSphere/AppServer/profiles/AppSrv01/properties # vi soap.client.props search for "com.ibm.SOAP.requestTimeout=180" and change 180 to 6000.
2. Update the "HTTP inbound channel" Read/Write timeout value from 60 to 180 seconds for your Portal servers and Deployment Manager. At your dmgr admin console, select Servers → Application Servers → (your WebSphere Portal) → Web Container settings → Web container transport chains → HTTP INbound channel System Administration → Deployment Manager → Web container transport chains → HTTP Inbound channel.
3. Update the JVM max heapsize to 768MB for your Portal servers and Deployment Manager.
4. Stop server1.
5. Stop WebSphere_Portal.
6. Stop Deployment Manager.
7. If you receive an error message such as "too many open files", then issue "-ulimit -n8192", and reset it to "ulimit -n" after you successfully complete the ./addNode.sh.

With WebSphere Application Server ND v5.1.1.4 installed, we can now federate one of the WebSphere Portal nodes; however, we encountered the following error when trying to run addNode.sh:

```
litwps01:/opt/WebSphere/AppServer/bin # ./addNode.sh litwps04.ltic.pok.ibm.com
8879 -includeapps
ADMU0116I: Tool information is being logged in file
/opt/WebSphere/AppServer/logs/addNode.log
ADMU0001I: Begin federation of node litwps01 with Deployment Manager at
litwps04.ltic.pok.ibm.com:8879.
ADMU0009I: Successfully connected to Deployment Manager Server:
litwps04.ltic.pok.ibm.com:8879
ADMU0111E: Program exiting with error:
com.ibm.websphere.management.exception.AdminException: ADMU2092E:
The node and Deployment Manager must have the same product
extensions, but they do not match. The node product extension is
PME and the Deployment Manager product extension is BASE.
ADMU0211I: Error details may be seen in the file:
/opt/WebSphere/AppServer/logs/addNode.log
ADMU1211I: To obtain a full trace of the failure, use the -trace option.
```

We learned that all nodes, including WebSphere Application Server Network Deployment Manager, must contain the same exact components and extensions. This means we must have WBISF v5.1.1.2 installed on our WebSphere Application Server ND server—litwps04.ltic.pok.ibm.com.

We install and apply WBISF v5.1.1.2 to litwps04.ltic.pok.ibm.com using the same instructions as outlined in the previous section.

We use the following WebSphere Portal InfoCenter as a reference:

<http://publib.boulder.ibm.com/pvc/wp/510/ent/en/InfoCenter/index.html>

(Clustering → Installing A WebSphere Portal Cluster → Setting up the cluster)

Now, we can try to federate the nodes in the cluster again:

1. Federate the 1st node (litwps01) using the -includeapps option.

```
litwps01:/opt/WebSphere/AppServer/bin # ./addNode.sh litwps04.ltic.pok.ibm.com
8879 -includeapps
```

The output from this command is very long. It includes every WebSphere Portal application to be uploaded to the target repository on WebSphere Application Server ND.

2. After the **addNode.sh** command completes on the 1st node, the default cell name defined in the WebSphere Portal `wpconfig.properties` file must be updated to account for the change in cell name. The file is located in the following directory:

```
/opt/WebSphere/PortalServer/config/wpconfig.properties
```

- a. Ensure that the following properties are uncommented and specify appropriate values:
 - i. **CellName** property: Specify the name of the cell to which the WebSphere Portal node belongs. The cell name can be identified by the `was_config_root/cells/cell_name` directory on the node, where `cell_name` indicates the cell to which the node belongs.

Following is the change we made to our `wpconfig.properties` file for the new cell name:

```
# CellName: The name of the WebSphere Application Server Cell
CellName=litwps04Network
```

3. Update the deployment manager configuration with the new WebSphere Portal configuration. Run the following command from `/opt/WebSphere/PortalServer/config`.

```
./WPSconfig.sh post-portal-node-federation-configuration
```

4. Federate the remaining 2nd and 3rd nodes (`litwps02` and `litwps03`) without `-includeapps`.
 - It is important not to use the `-includeapps` option. The applications are not transferred as they are already in the deployment manager's master configuration.
 - Applications stored in the master configuration are still only assigned to `litwps01`. In order to share them with `litwps02/litwps03` and any additional nodes, a cluster must be created.

```
litwps02:/opt/WebSphere/AppServer/bin # ./addNode.sh
litwps04.ltic.pok.ibm.com 8879
litwps03:/opt/WebSphere/AppServer/bin # ./addNode.sh
litwps04.ltic.pok.ibm.com 8879
```

5. After successfully federating the 2nd and 3rd node, update the default cell name in `wpconfig.properties` with the same cell name we defined for `litwps01`. Note that for these secondary nodes, their **Primary Node** property must be set to `false`.

```
/opt/WebSphere/PortalServer/config/wpconfig.properties
```

- a. Ensure that the following properties are uncommented, and specify appropriate values:
 - **CellName** property: Specify the name of the cell to which the WebSphere Portal node belongs. The cell name can be identified by the `was_config_root/cells/cell_name` directory on the node, where `cell_name` indicates the cell to which the node belongs.
 - **PrimaryNode** property: Because this node is a secondary node, set this property to `false`.

Following is an example of the change we made to our `wpconfig.properties` file:

```
CellName=litwps04Network
PrimaryNode=false
```

6. Update the deployment manager configuration with the new WebSphere Portal configurations.

7. Execute the following command for both litwps02 and litwps03 from /opt/WebSphere/PortalServer/config.
./WPSconfig.sh post-portal-node-federation-configuration
 8. Create the cluster and added the nodes:
 - a. With the Deployment Manager started, access the admin console at the following Web address
<http://litwps04.ltic.pok.ibm.com:9090/admin>
 - b. Click **Servers** → **Clusters** → **New**.
 - c. Enter the following cluster information:
 - i. Clustername = WPS_Cluster.
 - ii. Check the box Prefer local enabled.
 - iii. Select the option **Select an existing server** to add to this cluster, and then choose server **WebSphere_Portal** on litwps01 from the list.
 - iv. Click **Next**.
 - d. Create additional cluster members.
 - i. Define the name for the new cluster member:
WebSphere_Portal2
WebSphere_Portal3
- Note:** Would not allow the use WebSphere Portal since it already exists.
- ii. Select the appropriate node.
litwps02
litwps03
 - iii. Select the appropriate HTTP port setting, depending on the cluster topology you are using:
Horizontal: Unchecked the box **Generate Unique HTTP Ports**.
 - iv. Click **Apply**.
 - v. Click **Next** to view the summary.
 - e. To create the new cluster, click **Finish**.
 - f. To save your changes, click **Save**.
 - g. Click **Save** again to confirm the changes and to update the deployment manager configuration.

The cluster members should appear in the screen as shown in Figure 5 on page 24.

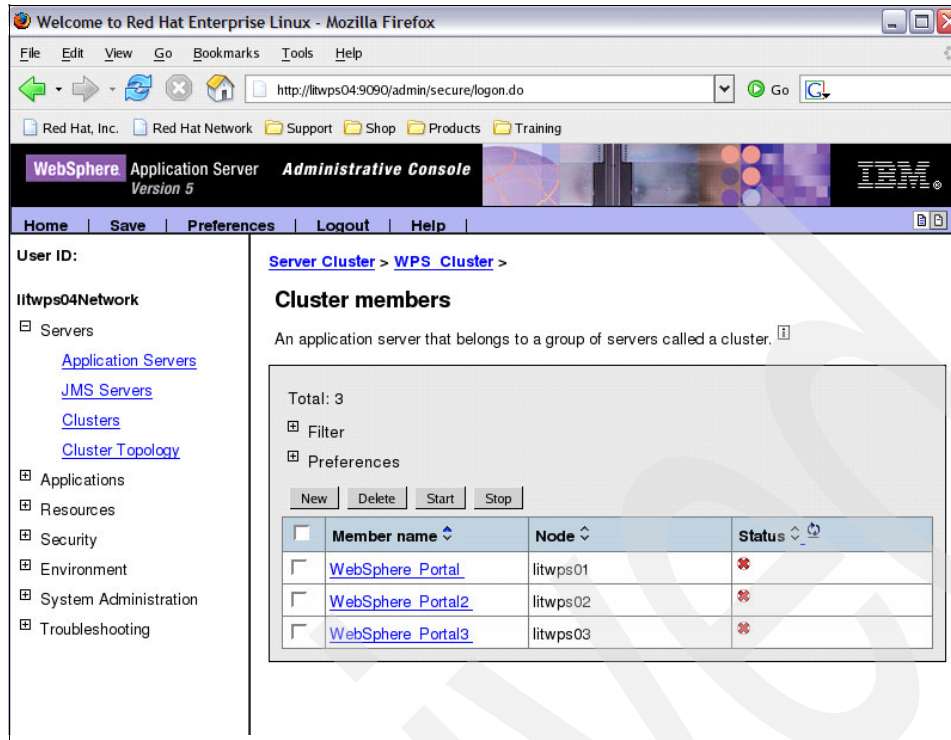


Figure 5 WebSphere Portal cluster members

- To enable portlet deployment in the cluster, we modify the `DeploymentService.properties` file on each WebSphere Portal member node, and set the `wps.appserver.name` property to the name of the cluster we defined.

```
/opt/WebSphere/PortalServer
/shared/app/config/services/DeploymentService.properties
```

Note: Each node in the cluster should have the same synchronization settings to ensure consistency between WebSphere Portal server configurations. By default, automatic synchronization occurs between every node when a change is made to the configuration of the cell, which is checked for updates once per minute.

For example, following is the change we made in `DeploymentService.properties`:

```
#wps.appserver.name=WebSphere_Portal1
wps.appserver.name=WPS_Cluster
```

- Update the configuration on each WebSphere Portal node with appropriate cluster member information.
- Edit the WebSphere Portal `wpconfig.properties` file for each WebSphere Portal instance:


```
/opt/WebSphere/PortalServer/config/wpconfig.properties
```
- Ensure that the following properties are uncommented and specify appropriate values:

ServerName property: Update this property to match the cluster member name used to identify the node to the deployment manager. To determine the cluster member name, click **Servers** → **Cluster Topology** in the administrative console for the deployment manager, and expand the cluster you are working with to view the cluster members.

- Modify `/opt/WebSphere/PortalServer/config/wpconfig.properties` file on `litwps01` and change `ServerName=WebSphere_Portal`

litwps02 and change ServerName=WebSphere_Portal2

litwps03 and change ServerName=WebSphere_Portal3

13. Save changes.

14. Resynchronize the nodes:

- a. In the administrative console for the deployment manager, click **Save** to store your administrative configuration.
- b. Select **System Administration** → **Nodes**, select the node from the list, and click **Full Resynchronize**.

15. Regenerate the Web server plug-in using the deployment manager administrative console.

- a. **WebSphere Application Server Version 5**: Select **Environment** → **Update Web Server Plug-in** in the deployment manager administrative console, and click **OK**.
- b. Edit the `nd_root/config/cells/plugin-cfg.xml` file, and change any directory structure occurrences specific to the Network Deployment machine to match the directory structure used on the Web server. For example, references to `install_dir/WebSphere/DeploymentManager` would be replaced with `install_dir/WebSphere/AppServer`.

16. Stop and start the Web server:

- a. Copy the updated plugin-cfg.xml file to your Web server.
- b. Restart your Web server.

17. Restart all nodes in the cluster.

- a. Issue `stopNode.sh` and `startNode.sh` on all the nodes (litwps01 - litwps03).

18. Start Cluster and verify cluster setup works.

- a. Go to your **WebSphere Application Server admin console** → **Servers** → **Clusters** → **Check WPS_Cluster** and select **Start**.
- b. When the cluster is started, there are two ways of verifying if WebSphere Portal Cluster has started successfully.
 - i. Access it through the Web server, and it will bring you to the WebSphere Portal page via round-robin
[http://\[webserver\]/wps/portal](http://[webserver]/wps/portal)
 - ii. Access it individually through the following node web port:
<http://litwps01.ltic.pok.ibm.com:9081/wps/portal>
<http://litwps02.ltic.pok.ibm.com:9081/wps/portal>
<http://litwps03.ltic.pok.ibm.com:9081/wps/portal>

Once this is verified, you have completed setting up your WebSphere Portal Cluster.

Migrate CloudScape to DB2 on Linux for zSeries or DB2 on z/OS

WebSphere Portal stores configuration and access controls such as user identities, credentials and permissions for accessing portal resources, and user data in a database. By default, WebSphere Portal installs and uses the Cloudscape database.

The use of Cloudscape database is recommended for PoC, development, and test environments only. Cloudscape is limited to local access as it does not support remote access; therefore, you cannot use it in a clustered Portal environment. For a production environment, we recommend that you use a more robust database such as DB2.

If you plan to use a database other than Cloudscape, you must install it separately from WebSphere Portal. After installing the database software, there are some additional procedures for you to perform on the database server before you can perform the WebSphere Portal database transfer.

We used two external databases in our test—DB2 UDB for z/Linux and DB2 for z/OS. In both cases, we used DB2 Connect™ as an intermediate server to access the back-end database server. We installed and configured DB2 Connect as a standalone server and not directly on the same image as our WebSphere Portal Server nodes. Later we discovered that the InfoCenter states that if you plan to use DB2 for z/OS, the DB2 Connect client must be installed on the WebSphere Portal Server.

<http://publib.boulder.ibm.com/pvc/wp/510/ent/en/InfoCenter/index.html>

We were not aware of this information until after our testing completed. We always run with an intermediate DB2 Connect Server for our WebSphere Application Server Cluster testing and were not aware of this statement for WebSphere Portal. So, with regard to DB2 Connect placement, it turns out that our WebSphere Portal environment was not a supported one. With that said, we did not find any problems while running with our specific configuration in our particular environment, which is described next.

The database product and level details we used include the following:

1. DB2 Runtime Client running on SLES 8 31bit SP3 (litwps01 - litwps03.ltic.pok.ibm.com, and litdcon2.ltic.pok.ibm.com)

```
db2inst1@litwps01:~> db2level
DB21085I Instance "db2inst1" uses "32" bits and DB2 code release "SQL08023"
with level identifier "03040106".
Informational tokens are "DB2 v8.1.0.96", "s050811", "MI00144", and FixPak
"10".
Product is installed at "/opt/IBM/db2/V8.1".
```

2. DB2 Connect EE running on SLES 8 31bit SP3 (litdcon2.ltic.pok.ibm.com)

```
db2inst1@litdcon2:~> db2level
DB21085I Instance "db2inst1" uses "32" bits and DB2 code release "SQL08023"
with level identifier "03040106".
Informational tokens are "DB2 v8.1.0.96", "s050811", "MI00144", and FixPak
"10".
Product is installed at "/opt/IBM/db2/V8.1".
```

3. DB2 UDB Server running on SLES 8 31bit SP3 (litdat01.ltic.pok.ibm.com)

```
DB21085I Instance "db2inst1" uses "64" bits and DB2 code release "SQL08023"
with level identifier "03040106".
Informational tokens are "DB2 v8.1.0.96", "s050811", "MI00145", and FixPak
"10".
Product is installed at "/opt/IBM/db2/V8.1".
```

4. DB2 for z/OS v8.1.5

We used the WebSphere Portal InfoCenter as a reference to the database setup:

http://publib.boulder.ibm.com/infocenter/wp51help/index.jsp?topic=/com.ibm.wp.ent.doc/wpf/inst_start.html

In the following section, you are creating four databases on DB2 UDB for z/Linux:

- ▶ wpsdb - WebSphere Portal Database
- ▶ jcrdb - WebSphere Portal Database for DB2 Content Manager Runtime Edition tables
- ▶ fdbkdb - Feedback database
- ▶ lmbd - Likeminds database

Transfer database information to DB2 on z/Linux

1. Run the following commands on the DB2 UDB server (litdat01) for each of the databases:

a. The WebSphere Portal database (represented as wpsdb):

```
db2 "CREATE database wpsdb using codeset UTF-8 territory us"
db2 "UPDATE database configuration for wpsdb using applheapsz 16384
app_ctl_heap_sz 8192"
db2 "UPDATE database configuration for wpsdb using stmtheap 60000"
db2 "UPDATE database configuration for wpsdb using locklist 400"
db2 "UPDATE database configuration for wpsdb using indexrec RESTART"
db2 "UPDATE database configuration for wpsdb using logfilsiz 1000"
db2 "UPDATE database configuration for wpsdb using logprimary 12"
db2 "UPDATE database configuration for wpsdb using logsecond 10"
```

```
db2set DB2_RR_TO_RS=yes
db2set DB2_EVALUNCOMMITTED=YES
db2set DB2_INLIST_TO_NLJN=YES
```

b. The Feedback database (represented as fdbkdb):

```
db2 "create database fdbkdb using codeset UTF-8 territory us collate using
identity"
db2 "update database configuration for fdbkdb using logprimary 4"
db2 "update database configuration for fdbkdb using logsecond 25"
db2 "update database configuration for fdbkdb using logfilsiz 4096"
db2 "update database configuration for fdbkdb using applheapsz 5120"
```

c. The LikeMinds database (represented as lmbd):

```
db2 "create database lmbd using codeset UTF-8 territory us"
```

On the DB2 server machine, enter the following commands, where:

- jcrdb is the name of the database used to store user data and objects
- icmadmin is the recommended database user for jcrdb - going to use db2inst1
- dbpassword is the password for jcrdb - going to use db2inst1

```
db2 "CREATE database jcrdb using codeset UTF-8 territory us"
db2 "UPDATE DBM CFG USING QUERY_HEAP_SZ 32768"
db2 "UPDATE DBM CFG USING UDF_MEM_SZ 7000"
db2 "UPDATE DBM CFG USING SHEAPTHRES 10000"
db2 "UPDATE DBM CFG USING MAXAGENTS 500"
db2 "UPDATE DBM CFG USING DFT_MON_TIMESTAMP OFF"
```

```
db2 "UPDATE DB CFG FOR jcrdb USING LOCKTIMEOUT 30"
db2 "UPDATE DB CFG FOR jcrdb USING LOCKLIST 1000"
db2 "UPDATE DB CFG FOR jcrdb USING STMTHEAP 16384"
db2 "UPDATE DB CFG FOR jcrdb USING AVG_APPLS 5"
db2 "UPDATE DB CFG FOR jcrdb USING SORTHEAP 256"
db2 "UPDATE DB CFG FOR jcrdb USING LOGPRIMARY 10"
db2 "UPDATE DB CFG FOR jcrdb USING LOGFILSIZ 1000"
db2 "UPDATE DB CFG FOR jcrdb USING LOGSECOND 20"
```

```

db2 "UPDATE DB CFG FOR jcrdb USING LOGBUFSZ 32"
db2 "UPDATE DB CFG FOR jcrdb USING MAXAPPLS 200"
db2 "UPDATE DB CFG FOR jcrdb USING APPLHEAPSZ 16384"
db2 "UPDATE DB CFG FOR jcrdb USING DFT_QUERYOPT 2"
db2 "UPDATE DB CFG FOR jcrdb USING DBHEAP 2400"
db2 "UPDATE DB CFG FOR jcrdb USING APP_CTL_HEAP_SZ 20000"

db2 "CONNECT TO jcrdb USER db2inst1 USING db2inst1"

db2 "CREATE BUFFERPOOL ICMLSFREQBP4 SIZE 1000 PAGESIZE 4 K"
db2 "CREATE BUFFERPOOL ICMLSVOLATILEBP4 SIZE 8000 PAGESIZE 4 K"
db2 "CREATE BUFFERPOOL ICMLSMAINBP32 SIZE 8000 PAGESIZE 32 K"
db2 "CREATE BUFFERPOOL CMBMAIN4 SIZE 1000 PAGESIZE 4 K"

db2 "CREATE BUFFERPOOL OBJECTPOOL SIZE 2000 PAGESIZE 32 K"
db2 "CREATE BUFFERPOOL OBJPARTSPOOL SIZE 200 PAGESIZE 32 K"
db2 "CREATE BUFFERPOOL SMSPOOL SIZE 500 PAGESIZE 4 K"
db2 "CREATE BUFFERPOOL PARTSPOOL SIZE 100 PAGESIZE 32 K"
db2 "CREATE BUFFERPOOL BLOBPOOL SIZE 1000 PAGESIZE 32 K"
db2 "CREATE BUFFERPOOL REPLICAPool SIZE 1000 PAGESIZE 32 K"
db2 "CREATE BUFFERPOOL TRACKINGPOOL SIZE 250 PAGESIZE 4 K"
db2 "CREATE BUFFERPOOL VALIDATEPOOL SIZE 500 PAGESIZE 32 K"

db2 "CREATE REGULAR TABLESPACE ICMLFQ32 PAGESIZE 32 K MANAGED BY SYSTEM
USING ('ICMLFQ32') BUFFERPOOL ICMLSMAINBP32"
db2 "CREATE REGULAR TABLESPACE ICMLNF32 PAGESIZE 32 K MANAGED BY SYSTEM
USING ('ICMLNF32') BUFFERPOOL ICMLSMAINBP32"
db2 "CREATE REGULAR TABLESPACE ICMVFQ04 PAGESIZE 4 K MANAGED BY SYSTEM USING
('ICMVQ04') BUFFERPOOL ICMLSVOLATILEBP4"
db2 "CREATE REGULAR TABLESPACE ICMSFQ04 PAGESIZE 4 K MANAGED BY SYSTEM USING
('ICMSFQ04') BUFFERPOOL ICMLSFREQBP4"
db2 "CREATE REGULAR TABLESPACE CMBINV04 PAGESIZE 4 K MANAGED BY SYSTEM USING
('CMBINV04') BUFFERPOOL CMBMAIN4"
db2 "CREATE SYSTEM TEMPORARY TABLESPACE ICMLSSYSTSPACE32 PAGESIZE 32 K
MANAGED BY SYSTEM USING ('icmlssystspace32') BUFFERPOOL ICMLSMAINBP32"
db2 "CREATE SYSTEM TEMPORARY TABLESPACE ICMLSSYSTSPACE4 PAGESIZE 4 K MANAGED
BY SYSTEM USING ('icmlssystspace4') BUFFERPOOL ICMLSVOLATILEBP4"

db2 "CREATE REGULAR TABLESPACE OBJECTS PAGESIZE 32 K MANAGED BY SYSTEM USING
('objects') BUFFERPOOL OBJECTPOOL"
db2 "CREATE REGULAR TABLESPACE OBJPARTS PAGESIZE 32 K MANAGED BY SYSTEM
USING ('objparts') BUFFERPOOL OBJPARTSPOOL"
db2 "CREATE REGULAR TABLESPACE SMS PAGESIZE 4 K MANAGED BY SYSTEM USING
('sms') BUFFERPOOL SMSPOOL"
db2 "CREATE REGULAR TABLESPACE BLOBS PAGESIZE 32 K MANAGED BY SYSTEM USING
('blobs') BUFFERPOOL BLOBPOOL"
db2 "CREATE REGULAR TABLESPACE REPLICAS PAGESIZE 32 K MANAGED BY SYSTEM
USING ('replicas') BUFFERPOOL REPLICAPool"
db2 "CREATE REGULAR TABLESPACE TRACKING PAGESIZE 4 K MANAGED BY SYSTEM USING
('tracking') BUFFERPOOL TRACKINGPOOL"
db2 "CREATE REGULAR TABLESPACE VALIDATEITM PAGESIZE 32 K MANAGED BY SYSTEM
USING ('validateitm') BUFFERPOOL VALIDATEPOOL"

db2 "DISCONNECT jcrdb"
db2 "TERMINATE"

```

2. On the DB2 UDB Server and Connect Server, check the services file. If it does not specify DB2 connection and interrupt service ports, specify the ports for your operating system. We have the following in our /etc/services:

```
db2c_db2inst1 50001/tcp # DB2 connection service port
db2i_db2inst1 50000/tcp # DB2 interrupt service port
```

3. On the DB2 UDB Server and Connect Server, set DB2COMM to TCP/IP by using the db2set command as follows:

```
db2set DB2COMM=TCPIP
```

4. Set up the correct service name by entering the following command on the DB2 UDB Server and Connect Server:

```
db2 "UPDATE DBM CFG USING svcename svce_name"
```

where svce_name is the connection service name that is specified in step 2.

```
db2 update dbm cfg using svcename db2c_db2inst1
```

5. Define a synchronous point manager (SPM) name for the DB2 Connect Server:

```
db2 update dbm cfg using SPM_NAME litdcon2
```

6. On the DB2 Connect Server, catalog the TCP/IP node with the IP address of the remote DB2 UDB server, as follows:

```
db2 catalog tcpip node litdat01 remote 192.168.71.104 server 50001
```

7. Create the following catalog entries on the DB2 Connect Server so that requests from the DB2 Connect Server are passed to the DB2 UDB Server. Catalog the WebSphere Portal, Feedback and Likeminds databases as followed on DB2 Connect Server:

```
db2 catalog database wpsdb as wpsdb at node litdat01
db2 catalog database fdbkdb as fdbkdb at node litdat01
db2 catalog database lmbd as lmbd at node litdat01
db2 catalog database jcrdb as jcrdb at node litdat01
```

8. On all DB2 Runtime Clients, use a text editor to open the /etc/services file. If the file does not specify the DB2 connection service port, add the following text to the file to specify the port for the remote DB2 instance:

Note: The connection service port on DB2 Client must match the connection service port on the DB2 Connect server. The ports should match by number but not necessarily by name.

```
db2c_db2inst1 50001/tcp # DB2 connection service port
```

9. On all DB2 Runtime Clients, catalog the TCP/IP node with the IP address of the remote database server, as follows:

```
db2 catalog tcpip node litdcon2 remote 192.168.71.136 server 50001
```

10. Create the following catalog entries on the DB2 Runtime Clients so that requests from the DB2 Runtime Client are passed to the DB2 Connect Server. Catalog the WebSphere Portal, Feedback, and Likeminds databases as followed on all DB2 Runtime Clients:

```
db2 catalog database wpsdb as wpsdb at node litdcon2
db2 catalog database fdbkdb as fdbkdb at node litdcon2
db2 catalog database lmbd as lmbd at node litdcon2
db2 catalog database jcrdb as jcrdb at node litdcon2
```

11. From your DB2 Runtime Clients, test your remote connection by issuing the following command in the DB2 command window:

```
db2 "connect to alias_name user username using password"
```

For example:

```
db2inst1@litwps01:~> db2 connect to wpsdb user db2inst1 using db2inst1
```

Database Connection Information

```
Database server      = DB2/LINUXZ64 8.2.2
SQL authorization ID = DB2INST1
Local database alias = WPSDB
```

The databases are now prepared for the data transfer.

a. Select the **Transfer data to another database** option as shown in Figure 6.

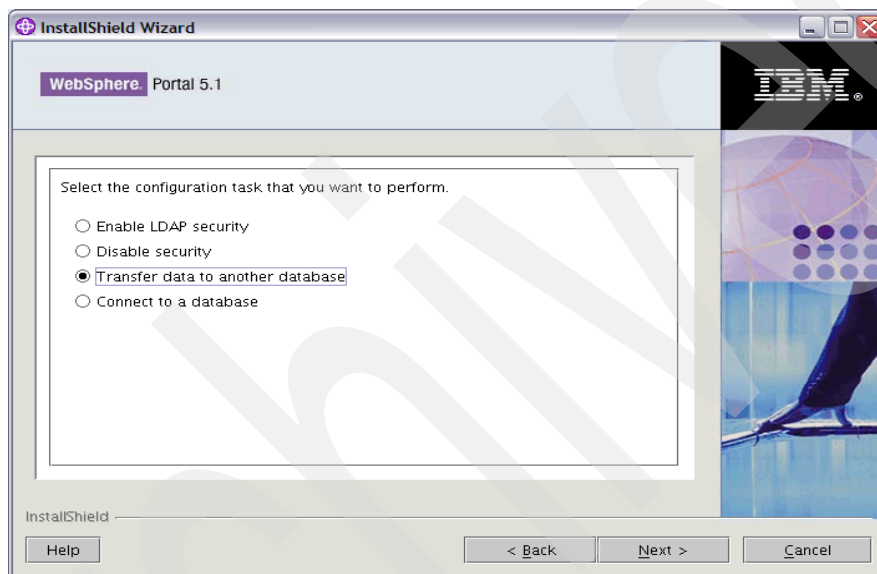


Figure 6 Select the **Transfer data to another database** option

b. Select the **IBM DB2 UDB** option in Figure 7.

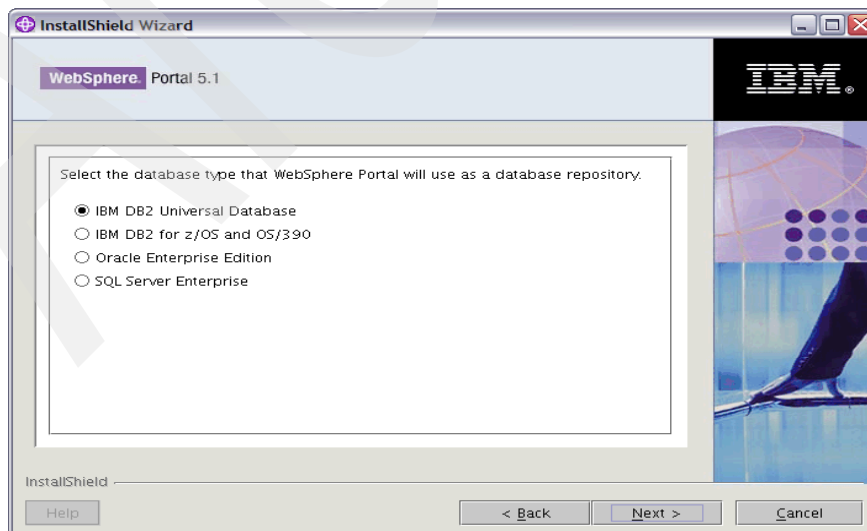


Figure 7 Select **IBM DB2 Universal Database™**

- c. Use the default DB2 properties file. See Figure 8.

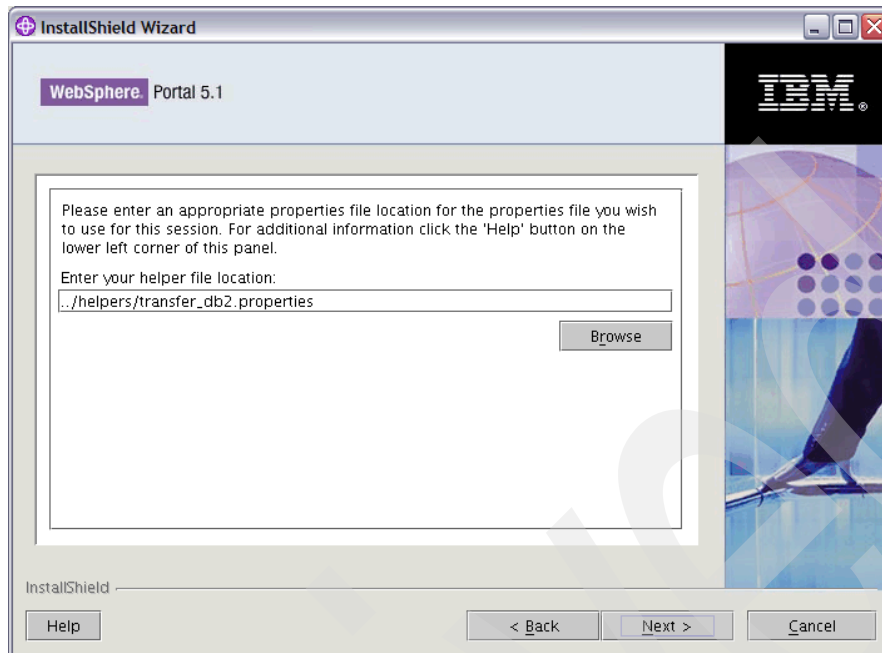


Figure 8 Use default DB2 properties

- d. We use JDBC™ Type 2(db2java.zip). Specify the full path in DbLibrary in Figure 9.

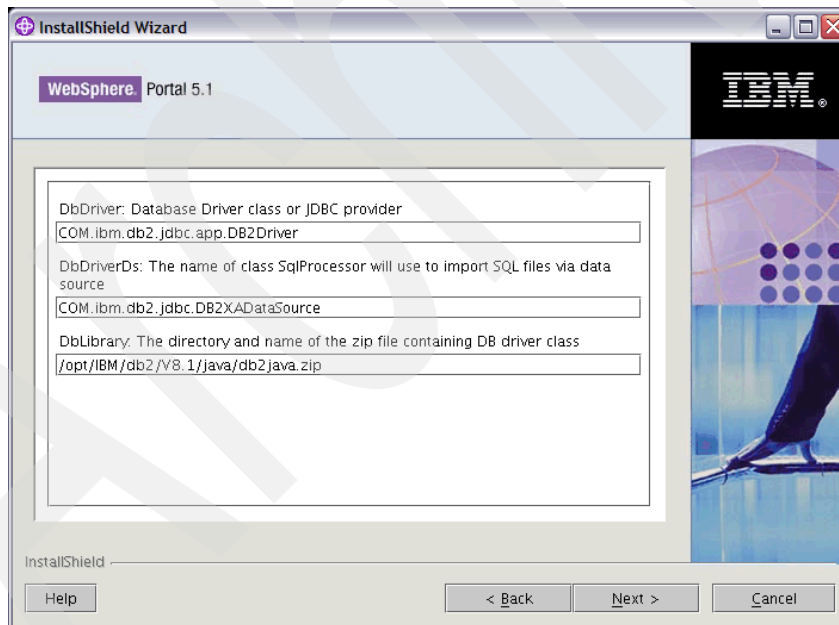


Figure 9 Specify full path to database driver

- e. Specify the options shown in Figure 10 on page 32 for the WebSphere Portal database.

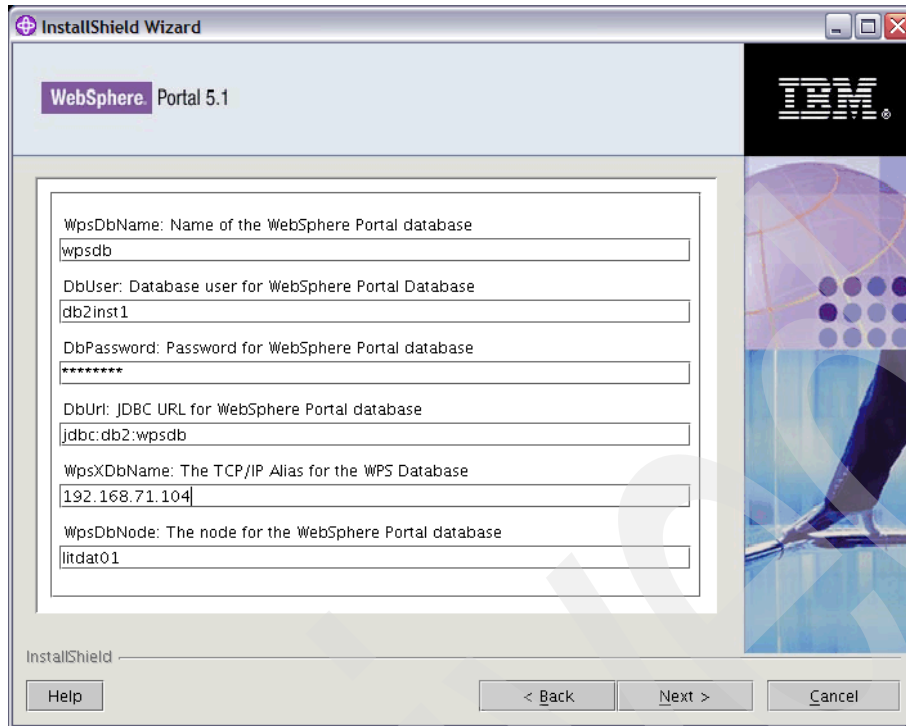


Figure 10 Provide WebSphere Portal database parameters

- f. Specify the f Member Manager database parameters as shown in Figure 11.

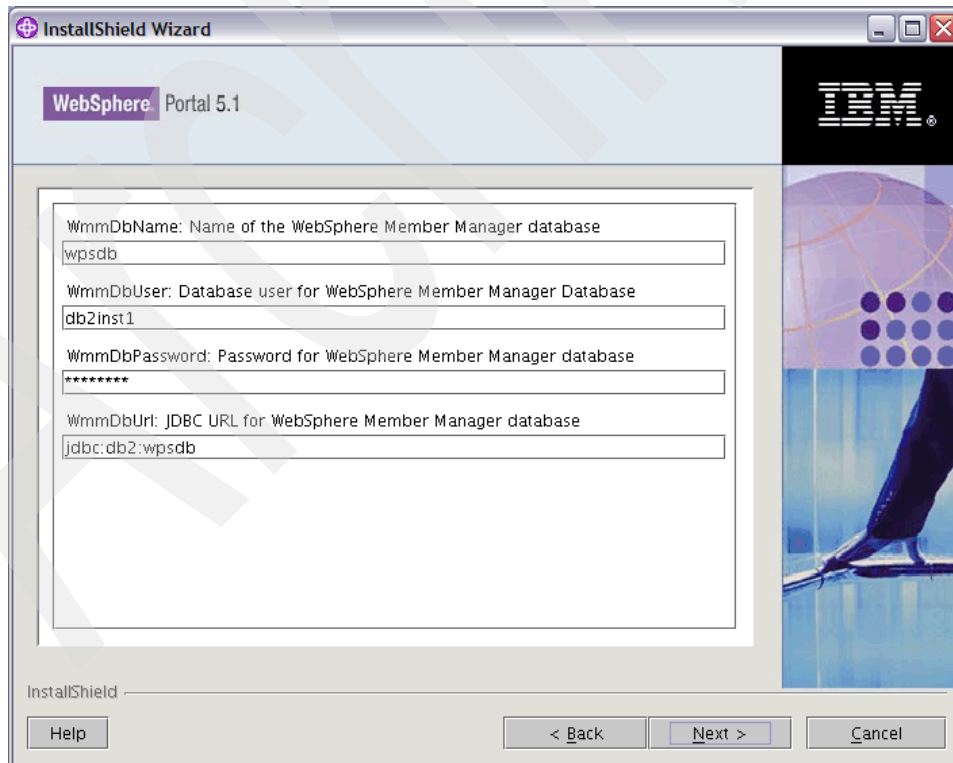


Figure 11 Provide the WebSphere Member Manager database parameters

g. Specify the Feedback database parameters as shown in Figure 12.

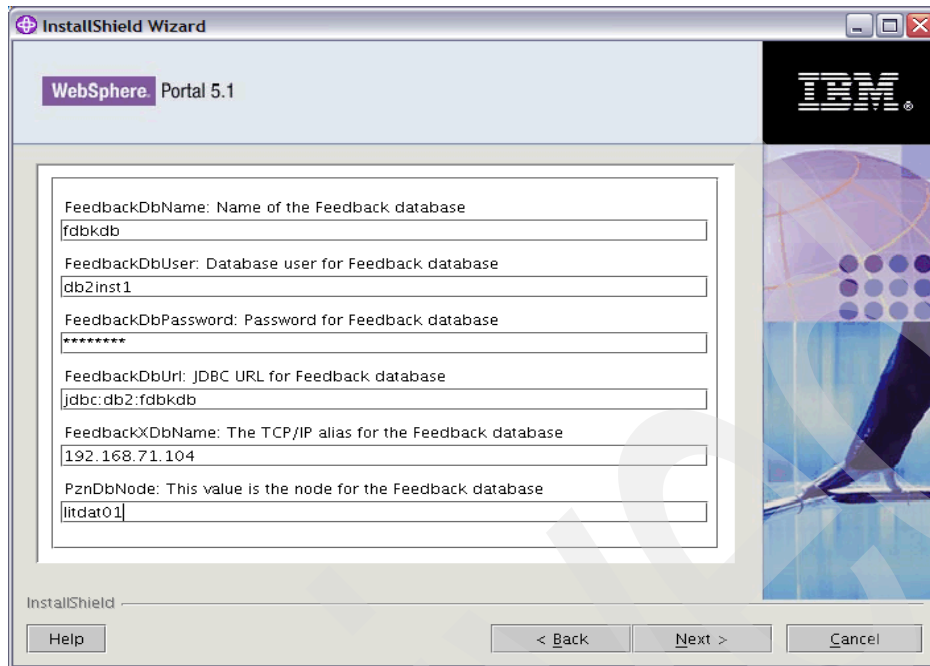


Figure 12 Provide Feedback database parameters

h. Specify the JCR database parameters as shown in Figure 13.

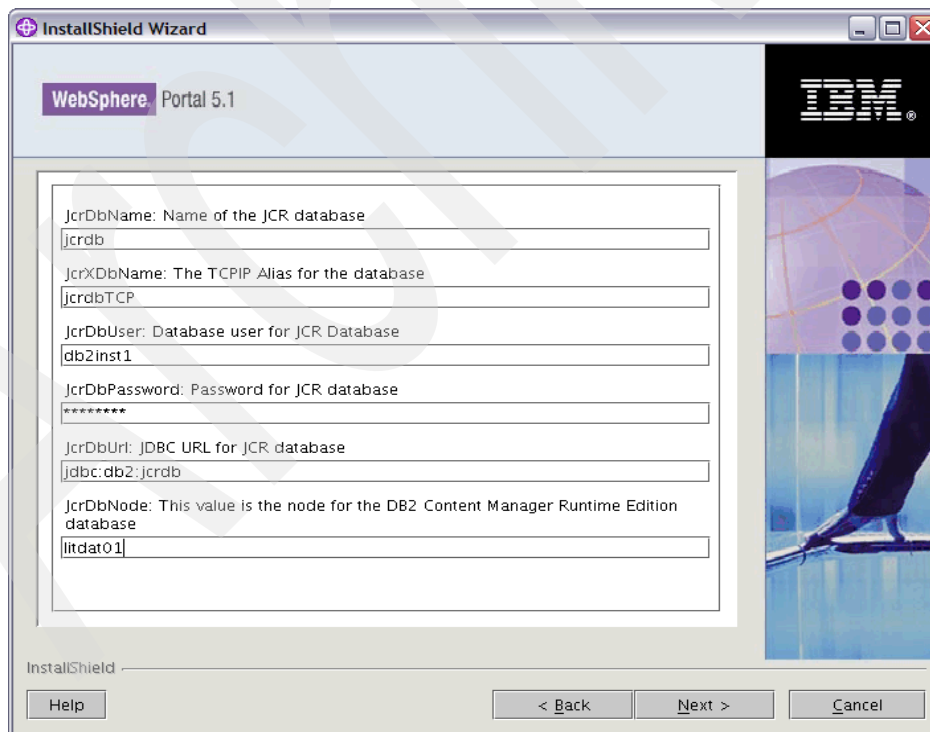


Figure 13 Provide the JCR database parameters

- i. Specify the Likeminds database parameters as shown in Figure 14.

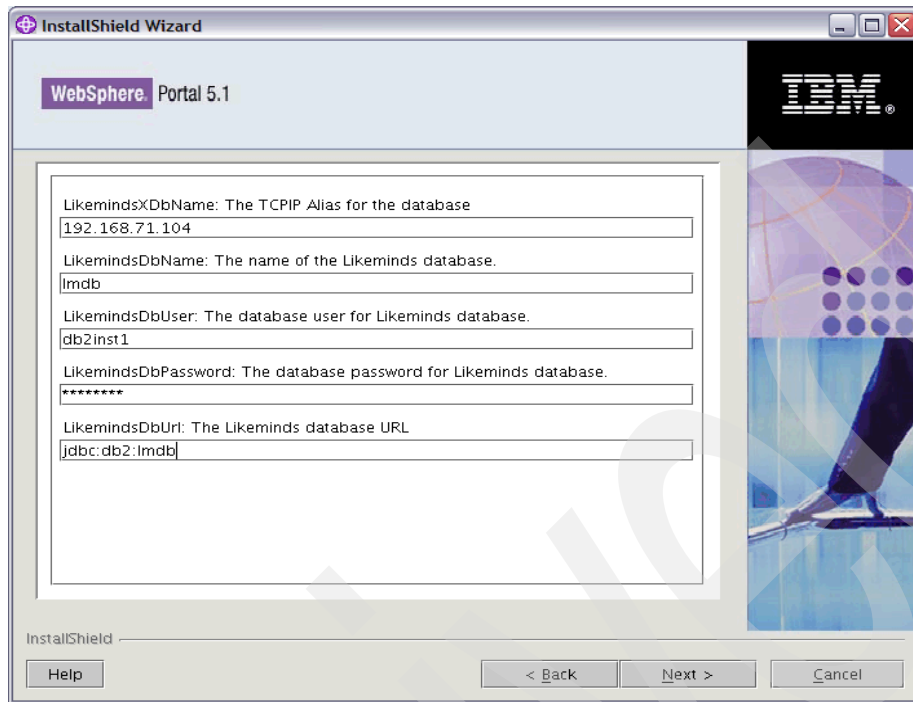


Figure 14 Provide the Likeminds database parameters

- j. After you enter all database information, click **Next** to begin transferring the data to another database. See Figure 15 on page 35.

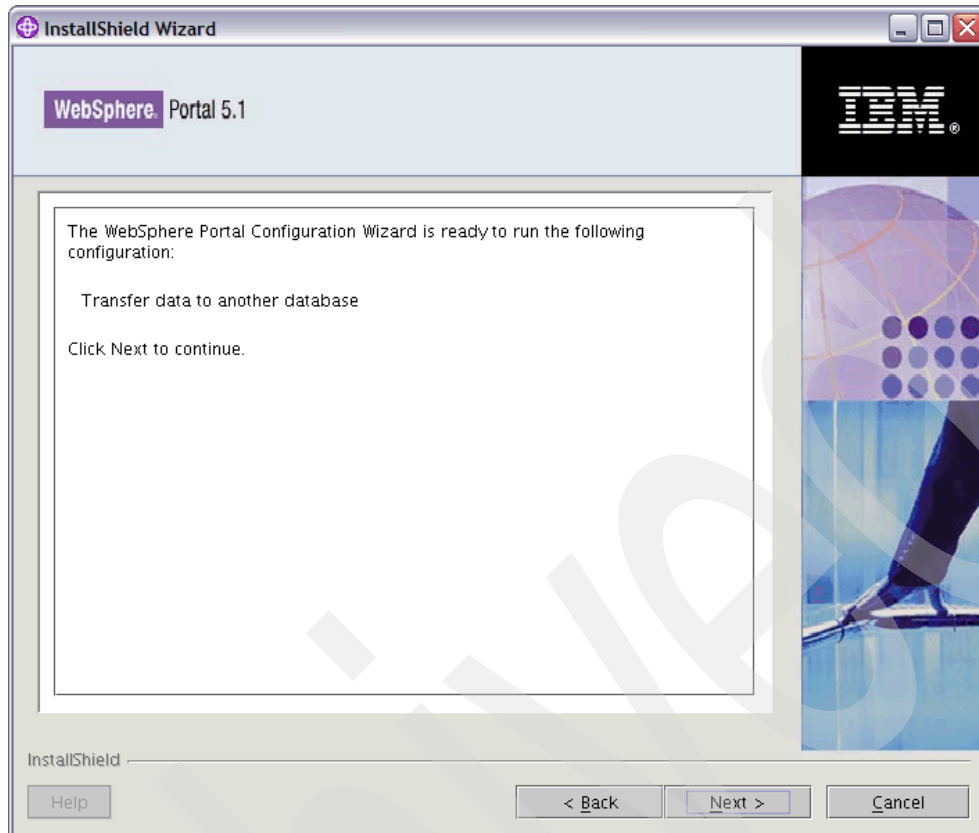


Figure 15 Click **Next** to transfer data

12. After the wizard completes successfully, we recommend that you run a db2 reorg check to improve performance. On the DB2 UDB Server, logon to each database and execute the following:

```
db2 connect to [database]
db2 reorgchk update statistics on table all
db2 terminate
```

13. Restart Nodes.

Note: The JDBC Data Source information was updated; however, we could not test the connection until after we recycled the nodes.

14. Test the new connections using the WebSphere Admin Console:

<http://litwps04.ltic.pok.ibm.com:9090/admin>

Resource → JDBC Providers (select view of litwps01) → wpsdbJDBC → Data Sources → select any or all data sources → Test Connection.

A message appears indicating that a successful connection occurred. In Custom Properties you can also verify that each datasource under litwps01 no longer references /opt/WebSphere/PortalServer/cloudscape/ as a value.

15. Repeat the procedure for the remaining nodes in the cluster—litwps02 and litwps03.

Important: DO NOT select **Transfer data to another database**. Choose **Connect to a database** in Figure 16 on page 36.

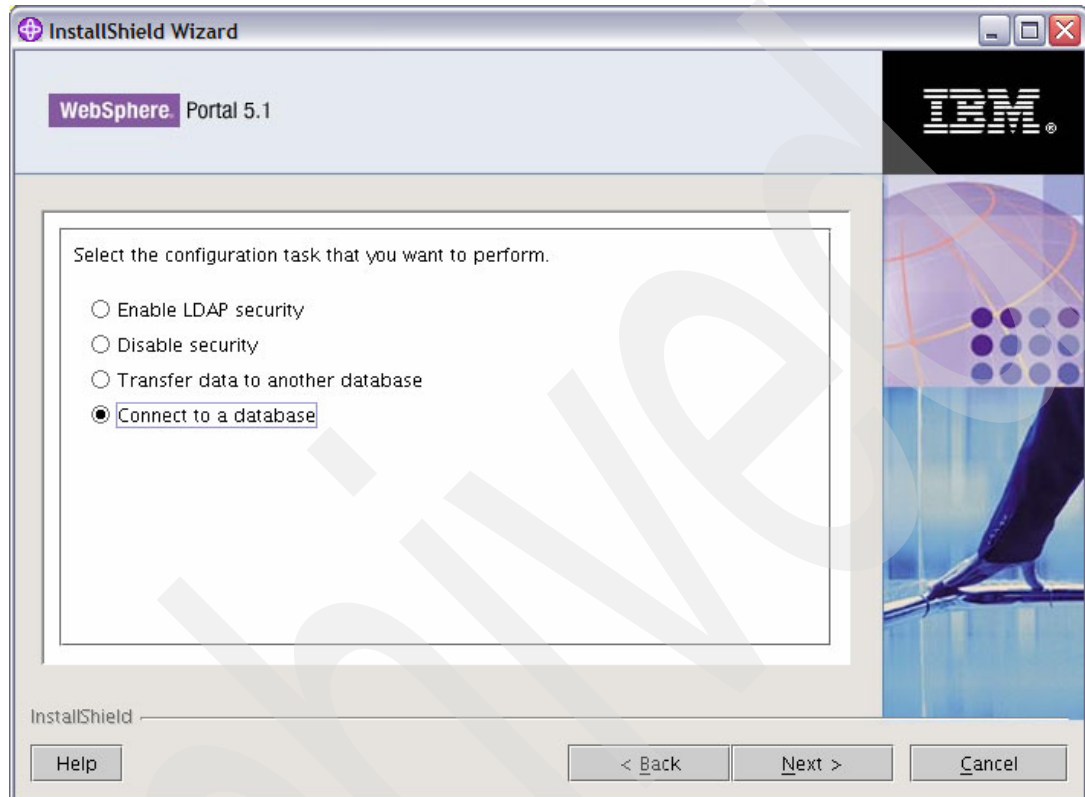


Figure 16 Select **Connect to a database**

The remaining steps are the same as the previous transfer with litwps01.

We completed transferring the database information from Cloudscape to DB2.

At this point, we have the database required by WebSphere Portal Server to support the administrative framework functionality.

Transfer database information to DB2 for z/OS

The instructions for transferring database information to DB2 for z/OS remain the same as the previous section instructions for transferring database information to DB2 on z/Linux except for the following additions and changes:

Create new databases on DB2 for z/OS. Preferably use different names to distinguish databases that belong to DB2 for z/Linux and DB2 for z/OS. Table 4 on page 37 shows how we used different naming conventions:

Table 4 DB2 for z/Linux names versus DB2 for z/OS names

	DB2 UDB	DB2 UDB for z/OS
WebSphere Portal database	wpsdb	wpsdbzos
WebSphere Portal database for DB2 Content Manager Runtime Edition tables	jcrdb	fdbkdbz
Feedback database	fdbkdb	lmbzos
Likeminds database	lmbd	jcrdbzos

For DB2 Connect

We added more catalog entries for DB2 Connect Server.

1. Catalog a node directory to DB2 for z/OS:

```
catalog tcpip node DBLNXT3 remote 192.168.25.36 server 446
```

2. Catalog dcs directories for the new database:

```
db2 catalog dcs db wpsdbzos as USIBMT6PETDB2
db2 catalog dcs db fdbkdbz as USIBMT6PETDB2
db2 catalog dcs db lmbzos as USIBMT6PETDB2
db2 catalog dcs db jcrdbzos as USIBMT6PETDB2
```

3. Catalog the new database aliases:

```
db2 catalog database wpsdbzos as wpsdbzos at node DBLNXT3 authentication DCS
db2 catalog database fdbkdbz as fdbkdbz at node DBLNXT3 authentication DCS
db2 catalog database lmbzos as lmbzos at node DBLNXT3 authentication DCS
db2 catalog database jcrdbzos as jcrdbzos at node DBLNXT3 authentication DCS
```

For DB2 Runtime Clients

We added more catalog entries for DB2 Runtime Clients.

```
db2 catalog database wpsdbzos as wpsdbzos at node litdcon2
db2 catalog database fdbkdbz as fdbkdbz at node litdcon2
db2 catalog database lmbzos as lmbzos at node litdcon2
db2 catalog database jcrdbzos as jcrdbzos at node litdcon2
```

Verify if the connection is from the DB2 Connect Server or the DB2 Runtime Client.

Following is an example of verifying connectivity from DB2 Connect Server to DB2 for z/OS:

```
db2inst1@litdcon2:~> db2 connect to jcrdbzos user db2ID using password
```

```
Database Connection Information
```

```
Database server          = DB2 OS/390 8.1.5
SQL authorization ID     = DB2LIT
Local database alias     = JCRDBZOS
```

Following is an example of verifying connectivity from DB2 Runtime Client to DB2 Connect Server:

```
db2inst1@litwps01:~> db2 connect to wpsdbzos user db2ID using password
Database Connection Information
Database server          = DB2 OS/390 8.1.5
SQL authorization ID    = DB2LIT
Local database alias    = WPSDBZOS
```

Now that you can remotely establish connections, some additional setup is required for the databases on z/OS.

1. Get additional commands, when creating the jcrdbzos database, from the following Web site:

http://publib.boulder.ibm.com/infocenter/wp51help/index.jsp?topic=/com.ibm.wp.ent.doc/wpf/setup_db2_zos_jcr.html

You can find a template of the commands at
`/opt/WebSphere/Portal/config/templates/db/db2_zos/jcr_sample.sql`

We worked closely with our DB2 administrator to define the required properties in DB2 such as the icmvolumes, icmvmcat, and bufferpools. The `jcr_sample.sql` is unique for your environment.

2. After you make your final changes to the `jcr_sample.sql` file, connect to the jcrdbzos database and pass it in:

```
db2 connect to jcrdbzos user db2ID using password
db2 < jcr_sample.sql
```

3. Connect to fdbkdbz database, and create the following tablespace:

```
db2 connect to fdbkdbz user db2ID using password
db2 CREATE TABLESPACE fdbkdbts IN fdbkdbz USING STOGROUP JCRICMSG PRIQTY 5000
SECQTY 500
```

4. Connect to lmbzos, and create the following tablespace:

```
db2 connect to lmbzos user db2ID using password
db2 CREATE TABLESPACE lmbdts IN lmbzos USING STOGROUP JCRICMSG PRIQTY 5000
SECQTY 500
```

We can now transfer the database contents to DB2 for z/OS by running the ConfigWizard on the first node—`litwps01`. The differences here between transferring the database contents from DB2 for z/Linux and DB2 for z/OS are as follows:

1. Select **IBM DB2 for z/OS and OS/390®** as shown in Figure 17 on page 39.

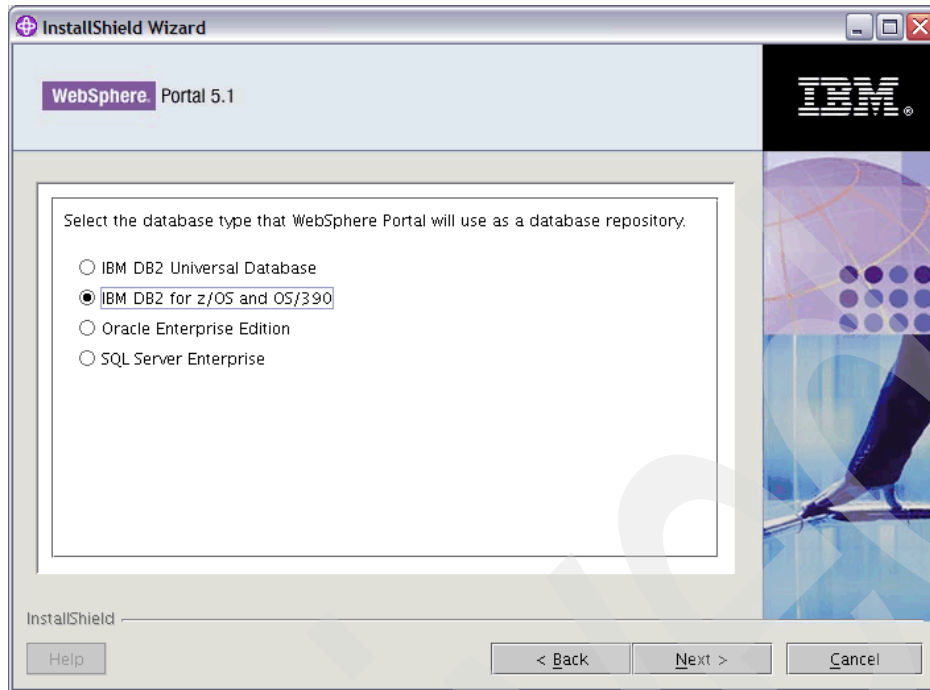


Figure 17 Select IBM DB2 for z/OS and OS/390

2. Select the DB2 z/OS properties file as shown in Figure 18.

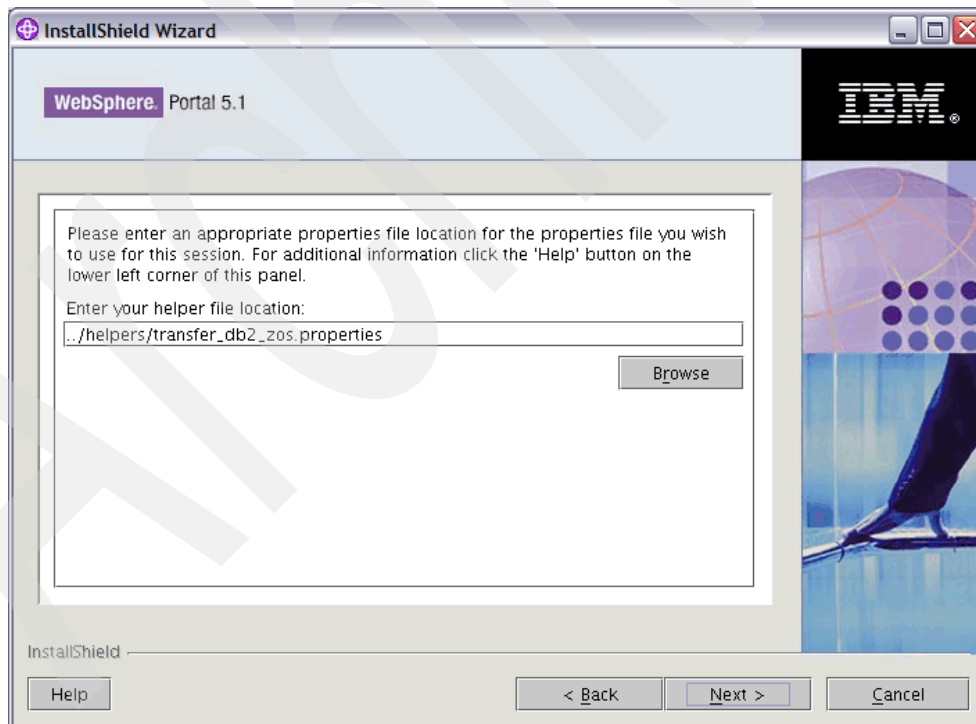


Figure 18 Select the DB2 z/OS properties file

- The remaining wizard panels remain unchanged; however, the JCR database requires that you complete additional fields as shown in Figure 19.

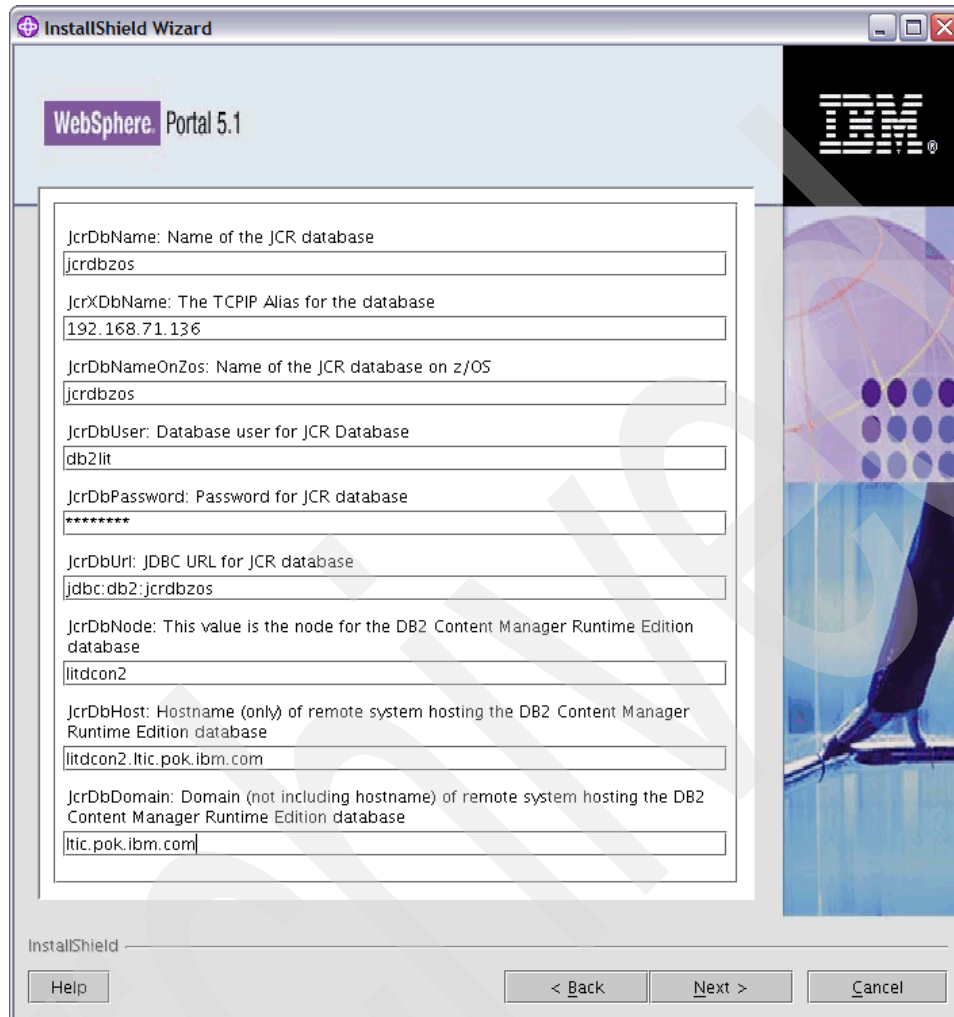


Figure 19 Provide more JCR database parameters

- Repeat the ConfigWizard procedure for the remaining nodes in the WebSphere cluster—litwps02 and litwps03.

Important: Remember to select the **Connect to a database** option for litwps02 and litwps03.

- Restart the nodes and Dmgr.
- Test the new connections using the WebSphere Admin Console.

<http://litwps04.ltic.pok.ibm.com:9090/admin>

Resource → JDBC Providers (select view of litwps01) → wpsdbDS → Data Sources → select any or all data sources → Test Connection.

- View the changes under the Custom Properties. The values were updated with the database names on DB2 for z/OS.

We successfully transferred the database information from Cloudscape to DB2 on z/OS.

Appendix A

```
<!-- ===== -->
<!-- ***** -->
<!-- Start changing WPName process -->
<!-- ***** -->

<target name="changePortalName" description="change WebSphere Portal Name"
depends="checkRelease,checkNewNode,checkOldNode">
  <property name="node.old" value="{oldNodeName}"/>
  <property name="node.new" value="{newNodeName}"/>
  <antcall target="changePortalConfig"/>
  <antcall target="changePortalConfigDir"/>
  <antcall target="changePortalinWAS"/>
  <antcall target="changePortalinWASConfigDir"/>
</target>

<target name="changePortalConfig" description="change the Portal name in
wpconfig.properties">
  <!-- Modify the Portal Name for wpconfig.properties under main config -->
  <echo message="Modifying wpconfig.properties from {node.old} to
${node.new}"/>
  <replace file="/opt/WebSphere/PortalServer/config/wpconfig.properties"
token="{node.old}" value="{node.new}"/>
  <!-- Modify the Portal Name for wp_portal.properties under Lotus Workplace
-->
  <echo message="Modifying wp_portal.properties from {node.old} to
${node.new}"/>
  <replace
file="/opt/WebSphere/PortalServer/config/work/was/wp_portal.properties"
token="{node.old}" value="{node.new}"/>
  <!-- Modify the Portal Name for wpconfig_ascii.properties under config -->
  <echo message="Modifying wpconfig_ascii.properties from {node.old} to
${node.new}"/>
  <replace file="/opt/WebSphere/PortalServer/config/wpconfig_ascii.properties"
token="{node.old}" value="{node.new}"/>
  <!-- Modify the Portal Name for index.hta under firstStep -->
  <echo message="Modifying index.hta from {node.old} to {node.new}"/>
  <replace file="/opt/WebSphere/PortalServer/firstStep/index.hta"
token="{node.old}" value="{node.new}"/>
</target>

<target name="changePortalConfigDir" description="change the Portal name in all
Lotus Workplace files">
  <!-- Modify all the Portal name occurrences under Lotus Workplace -->
  <echo message="Modifying all Lotus Workplace files from {node.old} to
${node.new}"/>
  <replace dir="/opt/WebSphere/PortalServer/config/work/was"
token="{node.old}" value="{node.new}">
    <include name="**/*.jac1"/>
  </replace>
  <!-- Modify all the Portal name occurrences in restoreportlet files-->
  <echo message="Modifying all restoreportlet files from {node.old} to
${node.new}"/>
```

```

        <replace dir="/opt/WebSphere/PortalServer/config/work/" token="${node.old}"
value="${node.new}">
            <include name="**/*.bat"/>
        </replace>
        <replace dir="/opt/WebSphere/PortalServer/config/work/" token="${node.old}"
value="${node.new}">
            <include name="**/*.sh"/>
        </replace>
        <!-- Modify all the Portal name occurrences under jcr -->
        <echo message="Modifying all jcr files from ${node.old} to ${node.new}"/>
        <replace dir="/opt/WebSphere/PortalServer/jcr/lib/com/ibm/icm"
token="${node.old}" value="${node.new}">
            <include name="**/*.properties"/>
        </replace>
        <replace dir="/opt/WebSphere/PortalServer/jcr/config/work"
token="${node.old}" value="${node.new}">
            <include name="**/*.params"/>
        </replace>
        <!-- Modify all the Portal name occurrences under wcm-->
        <echo message="Modifying all Web Content Manager files from ${node.old} to
${node.new}"/>
        <replace dir="/opt/WebSphere/PortalServer/wcm/config" token="${node.old}"
value="${node.new}">
            <include name="**/*.cfg"/>
        </replace>
        <replace dir="/opt/WebSphere/PortalServer/wcm/config" token="${node.old}"
value="${node.new}">
            <include name="**/*.properties"/>
        </replace>
    </target>

    <target name="changePortalinWAS" description="change the Portal name in all
WebSphere AppServer files">
        <!-- Modify all the Portal name occurrences under WebSphere Application
Server-->
        <echo message="Modifying all WebSphere Application Server files from
${node.old} to ${node.new}"/>
        <replace dir="/opt/WebSphere/AppServer/bin/wsinstance" token="${node.old}"
value="${node.new}">
            <include name="**/*.xml"/>
        </replace>
        <replace dir="/opt/WebSphere/AppServer/config/cells" token="${node.old}"
value="${node.new}">
            <include name="**/*.xml"/>
        </replace>
    </target>

    <target name="changePortalinWASConfigDir" description="change the Portal name in
all WebSphere AppServer directory structure">
        <!-- Modify all the Portal name occurrences in file system under WebSphere
Application Server -->
        <echo message="Modifying directory structure for WebSphere Application Server
files from ${cell.old} to ${cell.new}"/>

```

```

        <move
todir="${configRoot}/../bin/wsinstance/configdefaults/cells/${node.old}/nodes/${node.new}">
        <fileset
dir="${configRoot}/../bin/wsinstance/configdefaults/cells/${node.old}/nodes/${node.old}"/>
        </move>
        <move
todir="${configRoot}/../bin/wsinstance/configdefaults/cells/${node.new}">
        <fileset
dir="${configRoot}/../bin/wsinstance/configdefaults/cells/${node.old}"/>
        </move>

        <move
todir="${configRoot}/cells/${node.new}/applications/BPEContainer_${node.old}_server1.ear/deployments/BPEContainer_${node.new}_server1">
        <fileset
dir="${configRoot}/cells/${node.new}/applications/BPEContainer_${node.old}_server1.ear/deployments/BPEContainer_${node.old}_server1"/>
        </move>
        <move
tofile="${configRoot}/cells/${node.new}/applications/BPEContainer_${node.old}_server1.ear/BPEContainer_${node.new}_server1.ear"
file="${configRoot}/cells/${node.new}/applications/BPEContainer_${node.old}_server1.ear/BPEContainer_${node.old}_server1.ear"/>
        <move
todir="${configRoot}/cells/${node.new}/applications/BPEContainer_${node.new}_server1.ear">
        <fileset
dir="${configRoot}/cells/${node.new}/applications/BPEContainer_${node.old}_server1.ear"/>
        </move>

        <move
todir="${configRoot}/cells/${node.new}/applications/BPEWebClient_${node.old}_server1.ear/deployments/BPEWebClient_${node.new}_server1">
        <fileset
dir="${configRoot}/cells/${node.new}/applications/BPEWebClient_${node.old}_server1.ear/deployments/BPEWebClient_${node.old}_server1"/>
        </move>
        <move
tofile="${configRoot}/cells/${node.new}/applications/BPEWebClient_${node.old}_server1.ear/BPEWebClient_${node.new}_server1.ear"
file="${configRoot}/cells/${node.new}/applications/BPEWebClient_${node.old}_server1.ear/BPEWebClient_${node.old}_server1.ear"/>
        <move
todir="${configRoot}/cells/${node.new}/applications/BPEWebClient_${node.new}_server1.ear">
        <fileset
dir="${configRoot}/cells/${node.new}/applications/BPEWebClient_${node.old}_server1.ear"/>
        </move>

```

```

        <move
todir="${configRoot}/../installedApps/${node.new}/BPEContainer_${node.new}_server1
.ear">
        <fileset
dir="${configRoot}/../installedApps/${node.new}/BPEContainer_${node.old}_server1.e
ar"/>
        </move>
        <move
todir="${configRoot}/../installedApps/${node.new}/BPEWebClient_${node.new}_server1
.ear">
        <fileset
dir="${configRoot}/../installedApps/${node.new}/BPEWebClient_${node.old}_server1.e
ar"/>
        </move>
    </target>

<!-- =====>

```

Appendix B

```

#!/bin/bash
function help
# give help
#+=====+
{
echo "Usage: searchreplace oldhostname newhostname"
exit
}
#+=====+
main()
if [ $# != 2 ]; then
echo "Error: incorrect number of arguments"
help
fi
OLDHOST=$1
NEWHOST=$2
#+=====+
# find all the files from here on that contains the old host name that's not a
fgrep -R $OLDHOST * | grep -v Binary > outlist
#+=====+
# now replace those files that contain the old host name with the new host name
echo "Replacing files that contain $OLDHOST with $NEWHOST"
cat outlist | cut -d ":" -f 1 > outlist2
cat outlist2 | while read myline; do if [ -f "$myline" ]; then mv "$myline"
"$myline.bak"; sed "s/$OLDHOST/$NEWHOST/g" "$myline.bak" > "$myline"; rm
"$myline.bak"; fi; done
#+=====+
touch newdirs
# now find those directories that are the old host name
find ./ -name "$OLDHOST*" > olddirs
while [ "`cat olddirs | wc -l`" -gt 0 ]
do
cat olddirs | sed "s/$OLDHOST/$NEWHOST/g" > newdirs
exec 3<> olddirs

```

```
exec 4<> newdirs
# Change directories with old host name to new host name
read myline <&3; read myline2 <&4; echo "Moving $myline to $myline2"; mv "$myline"
"$myline2"
exec 4>&-
exec 3>&-
find ./ -name "*$OLDHOST*" > olddirs
done
#+=====+
# clean up
rm outlist
rm outlist2
rm olddirs
rm newdirs
```

The team that wrote this Redpaper

This Redpaper was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Scott Loveland is a Senior Technical Staff Member and chief architect in the IBM Customer Test organization in Poughkeepsie, New York. Scott founded the Poughkeepsie Test and Integration Center for Linux™ in 2002. Some of his recent publications include the white paper *High Availability Architectures for Linux on System z* (IBM), and the book *Software Testing Techniques: Finding the Defects that Matter* (Charles River Media).

Phillip Chan is a Staff Software Engineer and member of the zSeries Linux Virtual Server Platform Evaluation Test Team in the IBM Customer Test Organization in Poughkeepsie, New York. His areas of expertise include runtime configuration and troubleshooting. He also has several years of experience with DB2, WebSphere MQ, and WebSphere Application Server.

Theresa Tai is a Senior Software Engineer with IBM System z for the Systems and Technology Group in Poughkeepsie, New York. Before Theresa joined the New Technology Center in 1998, her experience included JES2 and JES3 product design, development, and service. In recent years Theresa focused on WebSphere Application Server, WebSphere Portal Server, and IBM JVM™ on zSeries platform. She also co-authored a number of IBM Redbooks™ to date on topics relating to WebSphere on zSeries.

Thanks to the following people for their contributions to this project:

Jin Xiong, Mark Ver
IBM Test and Integration Center for Linux on zSeries

Archived

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Send us your comments in one of the following ways:

- ▶ Use the online **Contact us** review redbook form found at:
ibm.com/redbooks
- ▶ Send your comments in an email to:
redbook@us.ibm.com
- ▶ Mail your comments to:
IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400 U.S.A.



Trademarks

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

Cloudscape™
DB2 Connect™
DB2 Universal Database™
DB2®
ECKD™
Enterprise Storage Server®
@server®

@server®
IBM®
Lotus®
OS/390®
Redbooks™
Redbooks (logo) ™
System z™

WebSphere®
Workplace™
z/OS®
z/VM®
zSeries®
z9™

The following terms are trademarks of other companies:

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

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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