

# XCAT 2 on AIX

## Booting AIX diskless nodes

Date: 5/15/2009

<a href="#">1.0 Overview</a>	<a href="#">1</a>
<a href="#">2.0 Deploying AIX diskless nodes using xCAT</a>	<a href="#">2</a>
<a href="#">2.1 Create a diskless image</a>	<a href="#">2</a>
<a href="#">2.2 Update the SPOT (optional)</a>	<a href="#">3</a>
<a href="#">2.2.1 Updating a SPOT with SSH software</a>	<a href="#">4</a>
<a href="#">2.3 Define xCAT networks</a>	<a href="#">5</a>
<a href="#">2.4 Create additional NIM network definitions (optional)</a>	<a href="#">5</a>
<a href="#">2.5 Define the HMC as an xCAT node</a>	<a href="#">7</a>
<a href="#">2.6 Discover the LPARs managed by the HMC</a>	<a href="#">7</a>
<a href="#">2.7 Define xCAT cluster nodes</a>	<a href="#">7</a>
<a href="#">2.8 Add IP addresses and hostnames to /etc/hosts</a>	<a href="#">8</a>
<a href="#">2.9 Create and define additional logical partitions (optional)</a>	<a href="#">8</a>
<a href="#">2.10 Gather MAC information for the boot adapters</a>	<a href="#">8</a>
<a href="#">2.11 Define xCAT groups (optional)</a>	<a href="#">9</a>
<a href="#">2.12 Set up post boot scripts (optional)</a>	<a href="#">10</a>
<a href="#">2.13 Verify the node definitions</a>	<a href="#">11</a>
<a href="#">2.14 Initialize the AIX/NIM diskless nodes</a>	<a href="#">11</a>
<a href="#">2.15 Verifying the node initialization before booting (optional)</a>	<a href="#">12</a>
<a href="#">2.16 Open a remote console (optional)</a>	<a href="#">12</a>
<a href="#">2.17 Initiate a network boot</a>	<a href="#">12</a>
<a href="#">2.18 Verify the deployment</a>	<a href="#">13</a>
<a href="#">3.0 Special Cases</a>	<a href="#">13</a>
<a href="#">3.1 Using other NIM resources</a>	<a href="#">13</a>
<a href="#">3.2 Booting a “dataless” node</a>	<a href="#">14</a>
<a href="#">3.3 Specifying additional values for the NIM node initialization</a>	<a href="#">14</a>
<a href="#">4.0 Cleanup</a>	<a href="#">15</a>
<a href="#">4.1 Removing NIM machine definitions</a>	<a href="#">15</a>
<a href="#">4.2 Removing NIM resources</a>	<a href="#">15</a>
<a href="#">5.0 Notes</a>	<a href="#">16</a>
<a href="#">5.1 Terminology</a>	<a href="#">16</a>
<a href="#">5.2 NIM diskless resources</a>	<a href="#">17</a>
<a href="#">5.3 NIM Commands</a>	<a href="#">18</a>
<a href="#">5.3.1 COSI commands</a>	<a href="#">18</a>
<a href="#">5.3.2 Thin server commands</a>	<a href="#">19</a>

## 1.0 Overview

This “How-To” describes how AIX diskless nodes can be deployed and updated using xCAT and AIX/NIM commands.

NIM (Network Installation Management) is an AIX tool that enables a cluster administrator to centrally manage the installation and configuration of AIX and optional software on machines within a networked environment. This document assumes you are somewhat familiar with NIM. For more information about NIM, see the IBM AIX Installation Guide and Reference.

(<http://www-03.ibm.com/servers/aix/library/index.html>)

The process described below is one basic set of steps that may be used to boot AIX diskless nodes and is not meant to be a comprehensive guide of all the available xCAT or NIM options.

**Before starting this process it is assumed you have completed the following.**

- An AIX system has been installed to use as an xCAT management node.
- xCAT and prerequisite software has been installed on the management node.
- The xCAT management node has been configured.
- One or more LPARs have already been created using the HMC interfaces.
- The cluster management network has been set up. (The Ethernet network that will be used to do the network installation of the cluster nodes.)

## 2.0 Deploying AIX diskless nodes using xCAT

### 2.1 Create a diskless image

In order to boot a diskless AIX node using xCAT and NIM you must create an xCAT *osimage* definition as well as several NIM resources.

The **mknimimage** will handle all the NIM setup as well as the creation of the xCAT *osimage* definition. It will not attempt to reinstall or reconfigure NIM if that process has already been completed. See the **mknimimage man** page for additional details.

**Note:** If you wish to install and configure NIM manually you can run the AIX **nim\_master\_setup** command (Ex. "*nim\_master\_setup -a mk\_resource=no -a device=<source directory>*").

By default, the **mknimimage** command will create the NIM resources in subdirectories of `/install`. Some of the NIM resources are quite large (1-2G) so it may be necessary to increase the files size limit.

For example, to set the file size limit to "unlimited" for the user "root" you could run the following command.

```
/usr/bin/chuser fsize=-1 root
```

There are several NIM resources that must be created in order to deploy a diskless node. The main resource is the NIM SPOT (Shared Product Object Tree). An AIX diskless image is essentially a SPOT. It provides a `/usr` file system for diskless nodes and a root directory whose contents will be used for the initial diskless nodes root directory. It also provides network boot support. The **mknimimage** command also creates default NIM *lpp\_source*, *root*, *dump*, and *paging* resources.

When you run the command you must provide a source for the installable images. This is typically the AIX product media or the name of an existing NIM *lpp\_source*

resource. You must also provide a name for the image you wish to create. This name will be used for the NIM SPOT resource that is created as well as the name of the xCAT *osimage* definition. The naming convention for the other NIM resources that are created is the *osimage* name followed by the NIM resource type, (ex. “*61cosi\_lpp\_source*”).

For example, to create a diskless image called “*61cosi*” using the AIX product CDs you could issue the following command.

```
mknimimage -t diskless -s /dev/cd0 61cosi
```

(Note that this operation could take a while to complete!)

The command will display a summary of what was created when it completes.

The NIM resources will be created in a subdirectory of */install/nim* by default. You can use the “-l” option to specify a different location.

You can also specify alternate or additional resources on the command line using the “attr=value” option, (“<nim resource type>=<resource name>”). For example, if you want to include a “*resolv\_conf*” resource named “*61cosi\_resolv\_conf*” you could run the command as follows.

```
mknimimage -t diskless -s /dev/cd0 61cosi resolv_conf=61cosi_resolv_conf
```

**Any additional NIM resources specified on the command line must be previously created using NIM interfaces.** (Which means NIM must already have been configured previously. )

**Note:** Another alternative is to run **mknimimage** without the additional resources and then simply add them to the xCAT *osimage* definition later. You can add or change the *osimage* definition at any time. When you initialize and install the nodes xCAT will use whatever resources are specified in the *osimage* definition.

The xCAT *osimage* definition can be listed using the **lsdef** command, modified using the **chdef** command and removed using the **rmnimimage** command. See the **man** pages for details.

To get details for the NIM resource definitions use the AIX **lsnim** command. For example, if the name of your SPOT resource is “*61cosi*” then you could get the details by running:

```
lsnim -l 61cosi
```

To see the actual contents of a NIM resource use “*nim -o showres <resource name>*”. For example, to get a list of the software installed in your SPOT you could run:

```
nim -o showres 61cosi
```

## **2.2 Update the SPOT (optional)**

The SPOT created in the previous step should be considered the basic minimal diskless AIX operating system image. It does not contain all the software that would normally be installed as part of AIX, if you were installing a standalone system

from the AIX product media. (The “nim -o showres ...” command mentioned above will display what software is contained in the SPOT.)

You must install any additional software you need and make any customizations to the image before you boot the nodes.

It would also be good to add other configuration files such as /etc/passwd etc. These files will be then be available to every node that boots with this image.

For more information on updating a diskless image see the document called “Updating AIX cluster nodes” (<https://xcat.svn.sourceforge.net/svnroot/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT2onAIXUpdates.odt>)

See the section titled “Updating diskless nodes”

### 2.2.1 Updating a SPOT with SSH software

If you wish to use `ssh` as your remote shell you will have to install `openssl` and `openssh` along with several additional requisite software packages.

To help facilitate this task xCAT ships two AIX `installp_bundle` files. They are called `xCATaixSSL.bnd` and `xCATaixSSH.bnd`, and they are included in the xCAT tar file.

The basic process is:

- Copy the required software to the `lpp_source` resource that you used to create your SPOT.
- Install the software in the SPOT.

#### Copy the software.

You will need the `openssl` and `openssh` packages that you installed on the management server earlier in this process. You will also need some prerequisite RPM packages that were included in the xCAT dependency tar file. The easiest way to copy the software to the correct locations is to use the “nim -o update ..” command. Just provide the directory that contains your software and the NIM `lpp_source` resource name. (ie. “61cosi\_lpp\_source”).

For example:

```
nim -o update -a packages=all -a source=/tmp/myimages  
61cosi_lpp_source
```

#### Install the software into the SPOT.

To install additional software in the SPOT you can use the “nim -o cust” command.

To use the bundle files with this command you must first define them as NIM resources.

Copy the bundle files ( `xCATaixSN.bnd` and `xCATaixSSH.bnd`) to a location where they can be defined as a NIM resource, for example “/install/nim/installp\_bundle”.

To define the NIM resources you can run the following commands.

```
nim -o define -t installp_bundle -a server=master -a location=  
/install/nim/installp_bundle/xCATaixSSL.bnd xCATaixSSL
```

```
nim -o define -t installp_bundle -a server=master -a location=  
/install/nim/installp_bundle/xCATaixSSH.bnd xCATaixSSH
```

**Note:** You cannot update a SPOT that is currently allocated. To check to see if the SPOT is allocated you could run the following command.

```
lsnim -l <spot name>
```

Run the “nim -o cust ...” command to install the software in the SPOT.

```
nim -o cust -a lpp_source= 61cosi_lpp_source -a installp_flags=agcXY -a  
installp_bundle=xCATaixSSL 61cosi
```

```
nim -o cust -a lpp_source= 61cosi_lpp_source -a installp_flags=agcXY -a  
installp_bundle=xCATaixSSH 61cosi
```

Refer to the NIM documentation for more information on how to use the NIM commands mentioned above.

## **2.3 Define xCAT networks**

Create a network definition for each network that contains cluster nodes. You will need a name for the network and values for the following attributes.

<b>net</b>	The network address.
<b>mask</b>	The network mask.
<b>gateway</b>	The network gateway.

This “How-To” assumes that all the cluster node management interfaces and the xCAT management node interface are on the same network. You can use the xCAT **mkdef** command to define the network.

For example:

```
mkdef -t network -o net1 net=9.114.113.0 mask=255.255.255.224  
gateway=9.114.113.254
```

**Note:** NIM also requires network definitions. When NIM was configured in an earlier step the default NIM master network definition was created. The NIM definition should match the one you create for xCAT. If multiple cluster subnets are needed then you will need an xCAT and NIM network definition for each one. A future xCAT enhancement will simplify this by automatically creating NIM network definitions based on the xCAT definitions.

## **2.4 Create additional NIM network definitions (optional)**

For the processs described in this document we are assuming that the xCAT management node and the LPARs are all on the same network.

However, depending on your specific situation, you may need to create additional NIM network and route definitions.

NIM network definitions represent the networks used in the NIM environment. When you configure NIM, the primary network associated with the NIM master is automatically defined. You need to define additional networks only if there are nodes that reside on other local area networks or subnets. If the physical network is changed in any way, the NIM network definitions need to be modified.

The following is an example of how to define a new NIM network using the NIM command line interface.

### Step 1

Create a NIM network definition. Assume the NIM name for the new network is “clstr\_net”, the network address is “10.0.0.0”, the network mask is “255.0.0.0”, and the default gateway is “10.0.0.247”.

```
nim -o define -t ent -a net_addr=10.0.0.0 -a snm=255.0.0.0 -a  
routing1='default 10.0.0.247' clstr_net
```

### Step 2

Create a new interface entry for the NIM “master” definition. Assume that the next available interface index is “2” and the hostname of the NIM master is “xcataixmn”. This must be the hostname of the management node interface that is connected to the “clstr\_net” network.

```
nim -o change -a if2='clstr_net xcataixmn 0' -a cable_type2=N/A master
```

### Step 3

Create routing information so that NIM knows how to get from one network to the other. Assume the next available routing index is “2”, and the IP address of the NIM master on the “master\_net” network is “8.124.37.24”. Assume the IP address on the NIM master on the “clstr\_net” network is “10.0.0.241”. This command will set the route from “master\_net” to “clstr\_net” to be “10.0.0.241” and it will set the route from “clstr\_net” to “master\_net” to be “8.124.37.24”.

```
nim -o change -a routing2='master_net 10.0.0.241 8.124.37.24'  
clstr_net
```

### Step 4

Verify the definitions by running the following commands.

```
lsnim -l master
```

```
lsnim -l master_net
```

```
lsnim -l clstr_net
```

See the NIM documentation for details on creating additional network and route definitions. (*IBM AIX Installation Guide and Reference*. <http://www-03.ibm.com/servers/aix/library/index.html>)

## 2.5 Define the HMC as an xCAT node

The xCAT hardware control support requires that the hardware control point for the nodes also be defined as a cluster node.

The following command will create an xCAT node definition for an HMC with a host name of “*hmc01*”. The *groups*, *nodetype*, *mgt*, *username*, and *password* attributes **must** be set.

```
mkdef -t node -o hmc01 groups="all" nodetype=hmc mgt=hmc  
username=hscroot password=abc123
```

## 2.6 Discover the LPARs managed by the HMC

This step assumes that the LPARs were already created using the standard HMC interfaces.

Use the xCAT **rscan** command to gather the LPAR information. In this example we will use the “-z” option to create a stanza file that contains the information gathered by **rscan** as well as some default values that could be used for the node definitions.

To write the stanza format output of **rscan** to a file called “*mystanzafile*” run the following command.

```
rscan -z hmc01 > mystanzafile
```

This file can then be checked and modified as needed. For example you may need to add a different name for the node definition or add additional attributes and values etc.

**Note:** The stanza file will contain stanzas for objects other than the LPARs. This information must also be defined in the xCAT database. It is not necessary to modify the non-LPAR stanzas in any way.

## 2.7 Define xCAT cluster nodes

The information gathered by the **rscan** command can be used to create xCAT node definitions.

Since we have put all the node information in a stanza file we can now pass the contents of the file to the **mkdef** command to add the definitions to the database.

```
cat mystanzafile | mkdef -z
```

You can use the xCAT **lsdef** command to check the definitions (ex. “*lsdef -l node01*”). After the node has been defined you can use the **chdef** command to make any additional updates to the definitions, if needed.

## 2.8 Add IP addresses and hostnames to /etc/hosts

Make sure all node hostnames are added to /etc/hosts. Refer to the section titled “Add cluster nodes to the /etc/hosts file” in the following document for details. (<https://xcat.svn.sourceforge.net/svnroot/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT2onAIX.odt> )

## 2.9 Create and define additional logical partitions (optional)

You can use the xCAT **mkvm** command to create additional logical partitions for diskless nodes in some cases.

This command can be used to create new partitions based on an existing partition or it can replicate the partitions from a source CEC to a destination CEC.

The first form of the **mkvm** command creates new partition(s) with the same profile/resources as the partition specified on the command line. The starting numeric partition number and the *noderange* for the newly created partitions must also be provided. The LHEA port numbers and the HCA index numbers will be automatically increased if they are defined in the source partition.

The second form of this command duplicates all the partitions from the specified source CEC to the destination CEC. The source and destination CECs can be managed by different HMCs.

The nodes in the *noderange* must already be defined in the xCAT database. The “mgt” attribute of the node definitions must be set to “hmc”.

For example, to create a set of new nodes. (The “groups” attribute is required.)

```
mkdef -t node -o clstrn04-clstrn10 groups=all,aixnodes mgt=hmc
```

To create several new partitions based on the partition for node clstrn01.

```
mkvm -V clstrn01 -i 4 -n clstrn04-clstrn10
```

See the **mkvm** man page for more details.

## 2.10 Gather MAC information for the boot adapters.

Use the xCAT **getmacs** command to gather adapter information from the nodes. This command will return the MAC information for each Ethernet adapter available on the target node. The command can be used to either display the results or write the information directly to the database. If there are multiple adapters the first one will be written to the database and used as the install adapter for that node.



The command can also be used to do a ping test on the adapter interfaces to determine which ones could be used to perform the network boot. In this case the first adapter that can be successfully used to ping the server will be written to the database.

Before running **getmacs** you must first run the **makeconsvercf** command. You need to run **makeconsvercf** any time you add new nodes to the cluster.

***makeconsvercf***

Shut down all the nodes that you will be querying for MAC addresses.

***rpower aixnodes off***

To retrieve the MAC address for all the nodes in the group “*aixnodes*” and write the first adapter MAC to the xCAT database you could issue the following command.

***getmacs aixnodes***

To display all adapter information but not write anything to the database.

***getmacs -d aixnodes***

To retrieve the MAC address and do a ping test to determine which adapter MAC to use for the node you could issue the following command. (The ping operation may take a while to complete.)

***getmacs -d aixnodes -S 10.14.0.2 -G 10.14.0.2 -C 10.14.0.4***

The output would be similar to the following.

```
# Type Location Code MAC Address Full Path Name Ping Result Device Type
ent U9125.F2A.024C362-V6-C2-T1 fef9dfb7c602 /vdevice/l-lan@30000002 successful
virtual
ent U9125.F2A.024C362-V6-C3-T1 fef9dfb7c603 /vdevice/l-lan@30000003 unsuccessful virtual
```

From this result you can see that “*fef9dfb7c602*” should be used for this nodes MAC address.

For more information on using the **getmacs** command see the man page.

To add the MAC value to the node definition you can use the **chdef** command. For example:

***chdef -t node node01 mac=fef9dfb7c603***

## **2.11 Define xCAT groups (optional)**

There are two basic ways to create xCAT node groups. You can either set the “groups” attribute of the node definition or you can create a group directly.

You can set the “groups” attribute of the node definition when you are defining the node with the **mkdef** command or you can modify the attribute later using the **chdef** command. For example, if you want a set of nodes to be added to the group “aixnodes” you could run **chdef** as follows.

```
chdef -p -t node -o node01,node02,node03 groups=aixnodes
```

The “-p” option specifies that “aixnodes” be added to any existing value for the “groups” attribute.

The second option would be to create a new group definition directly using the **mkdef** command as follows.

```
mkdef -t group -o aixnodes members="node01,node02,node03"
```

These two options will result in exactly the same definitions and attribute values being created.

## **2.12 Set up post boot scripts (optional)**

xCAT supports the running of user-provided customization scripts on the nodes when they are deployed. For diskless nodes these scripts are run when the `/etc/inittab` file is processed during the node boot up.

To have your script run on the nodes:

1. Put a copy of your script in `/install/postscripts` on the xCAT management node. (Make sure it is executable.)
2. Set the “postscripts” attribute of the node or group definition to include the comma separated list of the scripts that you want to be executed on the nodes. For example, if you want to have your two scripts called “foo” and “bar” run on node “node01” you could use the **chdef** command as follows.

```
chdef -t node -o node01 postscripts=foo,bar
```

The order of the scripts in the list determines the order in which they will be run.

xCAT also runs some scripts to do default node configuration. You can see what scripts xCAT will run by looking at the “xcatdefaults” entry in the xCAT “postscripts” database table. (I.e. Run “`tabdump postscripts`”). You can change the default setting by using the xCAT **chtab** or **tabedit** command.

## 2.13 Verify the node definitions

Verify that the node definitions include the required information.

To get a listing of the node definitions you can use the **lsdef** command. For example, to display the definitions of all nodes in the group “aixnodes” you could run the following command.

```
lsdef -t node -l -o aixnodes
```

The output for one diskless node might look something like the following:

```
Object name: clstrn02  
cons=hmc  
groups=lpar,all  
hcp=clstrhmc01  
hostnames=clstrn02.mycluster.com  
id=2  
ip=10.1.3.2  
mac=001a64f9bfc9  
mgt=hmc  
nodetype=lpar,osi  
os=AIX  
parent=clstrf1fsp03-9125-F2A-SN024C352  
pprofile=compute
```

Most of these attributes should have been filled in automatically by xCAT.

**Note:** xCAT supports many different cluster environments and the attributes that may be required in a node definition will vary. For diskless nodes the node definition should include at least the attributes listed in the above example.

## 2.14 Initialize the AIX/NIM diskless nodes

You can set up NIM to support a diskless boot of nodes by using the xCAT **mkdsklsnode** command. This command uses information from the xCAT database and default values to run the appropriate NIM commands.

For example, to set up all the nodes in the group “aixnodes” to boot using the SPOT (COSI) named “61cosi” you could issue the following command.

```
mkdsklsnode -i 61cosi aixnodes
```

The command will define and initialize the NIM machines. It will also set the “profile” attribute in the xCAT node definitions to “61cosi”.

To verify that NIM has allocated the required resources for a node and that the node is ready for a network boot you can run the “**lsnim -l**” command. For example, to check node “*node01*” you could run the following command.

```
lsnim -l node01
```

**Note:**

The NIM initialization of multiple nodes is done sequentially and takes approximately three minutes per node to complete. If you are planning to initialize multiple nodes you should plan accordingly. The NIM development team is currently working on a solution for this scaling issue.

## **2.15 Verifying the node initialization before booting (optional)**

Once the **mkdsklnode** command completes you can check several things to see if it has been initialized correctly.

- The */etc/bootptab*, */etc/exports* files.
- The node info file in the */tftpboot* directory.
- The NIM node definition for the node. (“*lsnim -l node01*”)

## **2.16 Open a remote console (optional)**

You can open a remote console to monitor the boot progress using the xCAT **rcons** command. This command requires that you have **conserver** installed and configured.

**If you wish to monitor a network installation you must run rcons before initiating a network boot.**

To configure conserver run:

```
makeconservercf
```

To start a console:

```
rcons node01
```

**Note: You must always run makeconservercf after you define new cluster nodes.**

## **2.17 Initiate a network boot**

Initiate a remote network boot request using the xCAT **rnetboot** command. For example, to initiate a network boot of all nodes in the group “*aixnodes*” you could issue the following command.

```
rnetboot aixnodes
```

**Note:** If you receive timeout errors from the **rnetboot** command, you may need to increase the default 60-second timeout to a larger value by setting *ppctimeout* in the site table:

```
chdef -t site -o clustersite ppctimeout=180
```

## 2.18 Verify the deployment

- You can use the AIX **lsnim** command to see the state of the NIM installation for a particular node, by running the following command on the NIM master:

```
lsnim -l <clientname>
```

- Retry and troubleshooting tips:
  - For p6 lpar, it may be helpful to bring up the HMC web interface in a browser and watch the lpar status and reference codes as the node boots.
  - Verify network connections
  - If the **rnetboot** returns “unsuccessful” for a node, verify that bootp and tftp is configured and running properly.
    - View */etc/bootptab* to make sure an entry exists for the node.
    - Verify that the information in */tftpboot/<node>.info* is correct.
    - Stop and restart inetd:

```
stopsrc -s inetd
startsrc -s inetd
```
    - Stop and restart tftp:

```
stopsrc -s tftp
startsrc -s tftp
```
    -
  - Verify NFS is running properly and mounts can be performed with this NFS server:
    - View */etc/exports* for correct mount information.
    - Run the **showmount** and **exportfs** commands.
    - Stop and restart the NFS and related daemons:

```
stopsrc -g nfs
startsrc -g nfs
```
    - Attempt to mount a file system from another system on the network.
  - If the **rnetboot** operation is successful, but **lsnim** shows that the node is stuck at one of the netboot phases, you may need to redo your NIM definitions.

## 3.0 Special Cases

### 3.1 Using other NIM resources.

When you run the **mknimimage** command to create a new xCAT *osimage* definition it will create default NIM resources and add their names to the *osimage* definition. It is also possible to specify additional or different NIM resources to use for the *osimage*. To do this you can use the “attr=val [attr=val ...]” option. These “attribute equals value” pairs are used to specify NIM resource types and names to use when creating the the xCAT *osimage* definition. The “attr” must be a NIM resource type

and the “val” must be the name of a **previously defined** NIM resource of that type, (ie. “<nim\_resource\_type>=<resource\_name>”).

For example, to create a diskless image and include *tmp* and *home* resources you could issue the command as follows. This assumes that the *mytmp* and *myhome* NIM resources have already been created by using NIM commands directly.

```
mknimimage -t diskless -s /dev/cd0 611cosi tmp=mytmp home=myhome
```

These resources will be added to the xCAT *osimage* definition. When you initialize a node using this definition the **mkdsklsnode** command will include all the resources when running the “nim -o dcls\_init” operation.

See the NIM documentation for more information on supported diskless resources.

### **3.2 Booting a “dataless” node.**

AIX NIM includes support for “dataless” systems as well as “diskless”. NIM defines a dataless machine as one that has some local disk space that could be used for paging space and optionally the /tmp and /home. If you wish to use dataless machines you can create an xCAT *osimage* definition for them with the **mknimimage** command. When creating the *osimage*, use the “-t” option to specify a type of “dataless”.

For example, to create an *osimage* definition for “dataless” nodes you could run the command as follows.

```
mknimimage -s /dev/cd0 -t dataless 53cosi
```

When the node is initialized to use this image the **mkdsklsnode** command will run the “nim -o dtls\_init ..” operation.

See the NIM documentation for more information on the NIM support for dataless systems.

### **3.3 Specifying additional values for the NIM node initialization.**

When you run the **mkdsklsnode** command to initialize diskless nodes the command will run the required NIM commands using some default values. If you wish to use different values you can specify them on the **mkdsklsnode** command line using the “attr=val [attr=val ...]” option. See the **mkdsklsnode man** page for the details of what attributes and values are supported.

For example, when **mkdsklsnode** defines the diskless node there are default values set for the “speed”(100) and “duplex”(full) network settings. If you wish to specify a different value for “speed” you could run the command as follows.

```
mkdsklsnode -i myosimage mynode speed=1000
```

## 4.0 Cleanup

The NIM definitions and resources that are created by xCAT commands are not automatically removed. It is therefore up to the system administrator to do some clean up of unused NIM definitions and resources from time to time. (The NIM lpp\_source and SPOT resources are quite large.) There are xCAT commands that can be used to assist in this process.

### 4.1 Removing NIM machine definitions

Use the xCAT **rmdsklsnode** command to remove all NIM machine definitions that were created for the specified xCAT nodes. This command will not remove the xCAT node definitions.

For example, to remove the NIM machine definition corresponding to the xCAT node named “node01” you could run the command as follows.

```
rmdsklsnode node01
```

The previous example assumes that the NIM machine definition is the same name as the xCAT node name. If you had used the “-n” option when you created the NIM machine definitions with **mkdsklsnode** then the NIM machine names would be a combination of the xCAT node name and the *osimage* name used to initialize the NIM machine. To remove these definitions you must provide the **rmdsklsnode** command with the name of the *osimage* that was used.

For example, to remove the NIM machine definition associated with the xCAT node named “node2” and the *osimage* named “61spot” you could run the following command.

```
rmdsklsnode -i 61spot node02
```

If the NIM machine is currently running or the machine definition was left in a bad state you can use the **rmdsklsnode** “-f” (force) option. This will stop the node and deallocate any resources it is using so the machine definition can be removed.

The **rmdsklsnode** command is intended to make it easier to clean up NIM machine definitions that were created by xCAT. You can also use the AIX **nim** command directly. See the AIX/NIM documentation for details.

### 4.2 Removing NIM resources

Use the xCAT **rmnimimage** command to remove all the NIM resources associated with a given xCAT *osimage* definition. The command will only remove a NIM resource if it is not allocated to a node. You should always clean up the NIM node definitions before attempting to remove the NIM resources. The command will also remove the xCAT *osimage* definition that is specified on the command line.

For example, to remove the “61spot” *osimage* definition along with all the associated NIM resources run the following command.

```
rmnimimage 61cosi
```

If necessary, you can also remove the NIM definitions directly by using NIM commands. See the AIX/NIM documentation for details.

## 5.0 Notes

### 5.1 Terminology

#### **image**

The term “image” is used extensively in this document. The precise meaning of an “image” will vary depending on the context in which the term is being used. In general you can think of an image as the basic operating system image as well as other resources etc. that are needed to boot a node. In most cases in this document we will be referring to an image as either an xCAT *osimage* definition or an AIX/NIM diskless image (called a SPOT or COSI).

**osimage** - This is an xCAT object that can be used to describe an operation system image. This definition can contain various types of information depending on what will be installed on the node and how it will be installed. The image definition is not node specific and can be used to deploy multiple nodes. It contains all the information that will be needed by the underlying xCAT and NIM support to deploy the node.

**COSI** - A Common Operating System Image is the name used by AIX/NIM to refer to a SPOT resource. From the NIM perspective this would be an AIX diskless image.

#### **diskless node**

The operating system is not stored on local disk. For AIX systems this means the file systems are mounted from a NIM server. NIM also supports the concept of a *dataless* system which has some limited disk space that can be used for certain file systems. See the “Special Cases” section below for information on using additional NIM features.

#### **diskfull node**

For AIX systems this means that the node has local disk storage that is used for the operating system. Diskfull AIX nodes are typically installed using the NIM **rte** or **mksysb** type install methods.



## 5.2 NIM diskless resources

The following list describes the required and optional resources that are managed by NIM to support diskless and dataless clients.

### **boot**

Defined as a network boot image for NIM clients. The **boot** resource is managed automatically by NIM and is never explicitly allocated or deallocated by users.

### **SPOT**

Defined as a directory structure that contains the AIX run-time files common to all machines. These files are referred to as the **usr** parts of the fileset. The **SPOT** resource is mounted as the **/usr** file system on diskless and dataless clients.

Contains the **root** parts of filesets. The **root** part of a fileset is the set of files that may be used to configure the software for a particular machine. These **root** files are stored in special directories in the **SPOT**, and they are used to populate the root directories of diskless and dataless clients during initialization.

The network boot images used to boot clients are constructed from software installed in the **SPOT**.

A **SPOT** resource is required for both diskless and dataless clients.

### **root**

Defined as a parent directory for client **"/** (**root**) directories. The client root directory in the **root** resource is mounted as the **"/** (**root**) file system on the client.

When the resources for a client are initialized, the client **root** directory is populated with configuration files. These configuration files are copied from the **SPOT** resource that has been allocated to the same machine.

A **root** resource is required for both diskless and dataless clients.

This resource is managed automatically by NIM.

### **dump**

Defined as a parent directory for client dump files. The client dump file in the **dump** resource is mounted as the dump device for the client.

A **dump** resource is required for both diskless and dataless clients.

This resource is managed automatically by NIM.

### **paging**

Defined as a parent directory for client paging files. The client paging file in the **paging** resource is mounted as the paging device for the client.

A **paging** resource is required for diskless clients and optional for dataless clients.

This resource is managed automatically by NIM.

### **home**

Defined as a parent directory for client **/home** directories. The client directory in the **home** resource is mounted as the **/home** file system on the client.

A **home** resource is optional for both diskless and dataless clients.

### **shared\_home**

Defined as a **/home** directory shared by clients. All clients that use a **shared\_home** resource will mount the same directory as the **/home** file system.

A **shared\_home** resource is optional for both diskless and dataless clients.

### **tmp**

Defined as a parent directory for client **/tmp** directories. The client directory in the **tmp** resource is mounted as the **/tmp** file system on the client.

A **tmp** resource is optional for both diskless and dataless clients.

### **resolv\_conf**

This resource is a file that contains nameserver IP addresses and a network domain name.

It is copied to the **/etc/resolv.conf** file in the client's root directory.

A **resolv\_conf** resource is optional for both diskless and dataless clients.

The AIX/NIM resources for diskless/dataless machines will remain allocated and the node will remain initialized until they are specifically unallocated and uninitialized.

## **5.3 NIM Commands**

### **5.3.1 COSI commands**

AIX provides commands that may be used to manage the SPOT (or COSI) resource. Refer to the AIX man pages for further details.

- **mkcosi** Create a COSI (SPOT) for a thin server (diskless or dataless client) to mount and use.
- **chcosi** Manages a Common Operating System Image (COSI).
- **cpcosi** Create a copy of a COSI (SPOT).
- **lscosi** List the properties of a COSI (SPOT).
- **rmcosi** Remove a COSI (SPOT) from the NIM environment.

### 5.3.2 Thin server commands

AIX provides several commands that can be used to manage diskless (also called thin server) nodes. See the AIX man pages for further details.

- **mkts** Create a thin server and all necessary resources.
- **lsts** List the status and software content of a thin server.
- **swts** Switch a thin server to a different COSI.
- **dbts** Perform a debug boot on a thin server.
- **rmts** Remove a thin server from the NIM environment.