xCAT 2 on AIX

Booting AIX diskless nodes

Version: 2.1

Date: 10/31/2008

1.0 Deploying AIX diskless nodes using xCAT

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.
- All relevant base AIX services are configured and running. This includes (but is not limited to) bootp, tftp, NFS, and hostname resolution.
- xCAT and prerequisite software has been installed on the management node.
- 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.)

1.1 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 xCAT *osimage* definition contains the names of the NIM resources as well as other information about how the node should be deployed.

You can use the xCAT **mknimimage** command to automate this process.

The **mknimimage** command 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 been completed. See the **mknimimage** man page for additional details.

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. " *61image_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. This assumes that the "*61cosi_resolv_conf"* resources has already been created using NIM commands.

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

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

1.2 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.

See the section called "Updating a NIM SPOT resource" later in this document for details on how to update a SPOT resource.

1.3 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.

net	The network address.
mask	The network mask.
gateway	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=9.114.113.224 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.

1.4 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
```

1.5 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.

1.6 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.

1.7 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.) 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.

For example, 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

1.8 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.

1.9 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.

1.10 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.

1.11 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

1.12 Verify the deployment

- As soon as the **rnetboot** command returns 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.

  To configure conserver:

  Set the "cons" attribute of the node definitions to "hmc".

  chdef -t node -o aixnodes cons=hmc

  Run the xCAT command.

  makeconservercf

  Kill the conserver daemon if it is running, and restart it:

  conserver &

  (You may need to add /opt/freeware/bin and /opt/freeware/sbin to your PATH first).

  To start a console:

  rcons node01

- 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 lpars, 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. Try the "short" approach first:

  nim -F -o reset node01

  nim -o dkls_init node01

  rnetboot -f node01
- If that doesn't work, you may need to delete the entire client definition from NIM and recreate it:

  nim -F -o reset node01

  nim -o deallocate -a root=root -a paging=paging -a dump=dump -a spot=61cosi node01

  nim -o remove node01

  mkdsklsnode -i 61cosi node01

  rnetboot -f node01

2.0 Updating a NIM SPOT resource

There are three basic processes you can use to update a SPOT:

1. Install additional **installp** file sets or **rpm** packages.
2. Add or modify specific files, (such as /etc/inittab).
3. Use the **chroot** command to run commands in the root file system contained in the SPOT.

Note: <u>You should not attempt to update a SPOT resource that is currently allocated to a node.</u> If you need to update an allocated SPOT either you can shut down the nodes and deallocate the SPOT resource first or you can make a copy of the SPOT and update that.

2.1 Install additonal software

You may need to install additional software into the SPOT before it is used to boot the nodes.

You can use the AIX **chcosi** command to install both **installp** file sets and **rpm** packages in a SPOT resource. (If the SPOT you want to update is currently

allocated the **chcosi** command will automatically attempt to create a copy of the SPOT to update.)

Before running the **chcosi** command you must add the new filesets and/or RPMs to the *lpp_source* resource used to create the SPOT. If we assume the *lpp_source* location for *61cosi* is */install/nim/lpp_source/61cosi_lpp*. The **installp** packages would go in: */install/nim/lpp_source/61cosi_lpp/installp/ppc* and the RPM packages would go in: */install/nim/lpp_source/61cosi_lpp/RPMS/ppc*.

The AIX **chcosi** command supports installing, updating, rejecting, removing, and committing the **installp** packages in the common image. See the **chcosi** man page details.

For example, to install and commit the optional OpenSSH packages from the AIX Expansion Pack you could issue the following command.

chcosi –i –c –s 61cosi_lpp –f openssh.base openssh.license openssh.man.en_US openssh.msg.en_US openssh.msg.EN_US 61cosi

Any additional software that is needed can be installed in a similar manner.

> **Note**: When installing software into a SPOT the pre and post install scripts for a particular software package will not run any code that will impact your running system, (like restarting daemons etc.). The script will check to see if it's installing into a SPOT and it will not run that code.

RPM packages may also be installed in the diskless image. You can use the **chcosi** command to install an RPM as follows.

> *chcosi –i –s 61cosi_lpp –f R:mypack..aix5.3.ppc.rpm 61cosi*

2.2 Add or modify files

You can also update files in the SPOT/COSI manually. The root file system for the diskless node will be created by copying the "*inst_root"* directory contained in the SPOT. In the SPOT we created for this example the "*inst_root*" directory would be:

> */install/nim/spot/61cosi/usr/lpp/bos/inst_root/*

For example, if you need to update the */etc/inittab* file that will be used on the diskless nodes you could edit:

> */install/nim/spot/61cosi/usr/lpp/bos/inst_root/etc/inittab*

You can also copy specific files into the *inst_root* directory so they will be available when the nodes boot. For example, you could copy a script called *myscript* to /*install/nim/spot/61cosi/usr/lpp/bos/inst_root/opt/foo/myscript* and then add an entry to */etc/inittab* so that it would be run when the node boots.

All the diskless nodes that are booted using this SPOT will get a copy of *inst_root* as the initial root directory.

Note: There are several files that you may want to consider updating in the SPOT *inst_root* directory. For example:

- /etc/hosts

- /etc/resolv.conf
- /etc/password
- /etc/profile
- etc.

2.3 Use chroot to run commands in the SPOT root directory

You can run commands in the root environment of the SPOT.

To do this use must go to the "inst_root" directory in the SPOT and use the chroot command to run the other commands. Since the "inst_root" directory does not include /usr you will have to mount it from elswhere in the SPOT.

For example, assume the location of the "inst_root" directory is "/install/nim/spot/61cosi/usr/lpp/bos/inst_root". You could run a command "rmitab 'tty002'" as follows

cd /install/nim/spot/61cosi/usr/lpp/bos/inst_root

mount /install/nim/spot/61cosi/usr ./usr

chroot . /usr/sbin/rmitab " tty002"

umount /usr

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 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 dkls_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 Updating AIX diskless nodes using xCAT

This section describes how AIX diskless nodes can be updated using xCAT and AIX/NIM commands. It covers the switching of the node to a completely different image or to an updated version of the current image. It is not meant to be an exhaustive presentation of all options that are available to xCAT/AIX system administrators.

To update an AIX diskless node with new or additional software you must modify the NIM SPOT resource (operating system image) that the node is running.

Since you cannot modify a SPOT while a node is using it, you must either stop the nodes to update the image or create a new image for the nodes to use.

Stopping the nodes to do the updates means the nodes will be unusable for some period of time and there will be no easy way to back out an update if necessary. For these reasons the procedure described in this "How-To" will focus on creating a new image and rebooting the nodes with that image. The new image could be a

completely new operating system image or it could be a copy of the the existing image that you can update as needed.

4.1 Create a new image

4.1.1 Create a new image from different source

In this case we create a new xCAT *osimage* definition with a new set of resources by running the xCAT **mknimimage** command with the source for the new resources. This is the same way you created the original xCAT *osimage* definition for the node.

When you run the command you must provide a source for the installable images. This can be the location of the source code or the name of another 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 definition as well as the xCAT *osimage* definition.

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

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

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

(This operation could take a while to complete!)

The command will create new NIM lpp_source and SPOT resources. It will also create dump, paging, and root resources if needed. A new xCAT *osimage* definition will also be created, (called "*61cosi"),* which will contain the names of these resources.

You could also use the name of an existing NIM lpp_source resource as the source of a new *osimage* definition. For example, you could use a resource created for a previous *osimage* called *61cosi_lpp* to create a whole new *osimage* called *61cosi_updt* as follows.

mknimimage -t diskless -s 61cosi_lpp 61cosi_updt

The **mknimimage** command will display the contents of the new *osimage* definition when it completes.

This new image can now be updated and used to boot the node.

4.1.2 Copy an existing image

You can use the **mknimimage** command to create a copy of an image. For example, if the name of the currently running image is *61cosi* and you want make a copy of it to update, you could run the following command.

mknimimage -t diskless -i 61cosi 61cosi_updt

If a "-i " value is provided then all the resources from the xCAT *osimage* definition (*61cosi*) will be used in the new *osimage* definition except the SPOT resource. The

new SPOT resource will be copied from the one specified in the original definition and renamed using the new *osimage* name provided (*61cosi_updt)*. A new xCAT *osimage* definition will also be created, called "*61cosi_updt",* which will contain the names of these resources.

This new image can now be updated and used to boot the node.

4.2 Update the SPOT(COSI) (optional)

Once the SPOT is created you can update it in several ways.

- Install additional **installp** or **rpm** packages.
- Add or modify specific files, (such as /etc/inittab).
- Use the **chroot** command to run commands in the root file system contained in the SPOT/COSI.

The details for updating a SPOT are covered in the "Updating a NIM SPOT resource" section.

4.3 Verify the new image (optional)

To display the xCAT image definition run the xCAT **lsdef** command.

lsdef -t osimage -l -o 61cosi

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 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

4.4 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:

3. Put a copy of your script in /install/postscripts on the xCAT management node. (Make sure it is executable.)
4. 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.

4.5 Initialize the 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.

There are three basic situations where you would need to initialize a NIM diskless machine.

1. When you are booting a NIM diskless machine for the first time.
2. When you want to switch a running node to a new image.
3. When you want to do the initialization for a new image while the node is still running. (This avoids having the node be down while the initialization step is completing.)

For the first situation you can do the initialization of the nodes using the xCAT **mkdsklsnode** command as follows.

mkdsklsnode -i 61cosi node01,node09

If the "-i" value is not provided then the code checks the node definitions for the value of the "profile" attribute. If the "-i" value is provided then the command will set the "profile" attribute in the node definitions.

In the second situation you want to switch the nodes to use a new or updated image. You can use the "-f" (force) option of the **mkdsklsnode** command to do this. With this option the **mkdsklsnode** command will stop the running node, deallocate the resources and do the NIM re-initialization with the new image. In this case the node would be unavailable during the initialization as well as the time for the node reboot.

Note: The NIM support for re-initialization take 3-4 minutes and is done sequentially.

For example, to switch the node named "*node29*" to a new image named "*611spot*" you could run the following command.

mkdsklsnode -f -i 611spot node29

The name of the image ("*611spot*") is the xCAT *osimage* name which is also the name of the SPOT resource that was created for this *osimage* definition.

In the last scenario we want to initialize an xCAT diskless node while the node continues running. To do this we need to create an alternate NIM machine definition for the same xCAT cluster node.

Creating alternate NIM machine definitions is possible because the NIM name for a machine definition does not have to be the hostname of the node. This allows you to have multiple NIM machine definitions for the same node. Since all the NIM initialization of the alternate machine definition can be done while the node is running, the downtime for the node is reduced to the time it takes to reboot.

For example, to initialize the xCAT node named "*node42*" to use the xCAT *osimage* named "*61cosi*" you could run the following command.

mkdsklsnode -n -i 61cosi node42

The naming convention for the new NIM machine name is "<xcat_node_name>_<image_name>", (Ex."*node42_61cosi*"). You could continue to create alternate machine definitions for each new image you wish to use for the node. The last NIM machine name that is initialized will determine what the node will use for the next boot.

> Debug tip: If you have forgotten which machine name you last initialized with NIM, and want to verify which image will actually be loaded on the next boot, NIM creates a /tftpboot/<*hostname*>.info file that contains mount information for the SPOT and other resources.

Using the "-n" option will save time but it will also leave you with multiple alternate NIM machine definitions for the same node. If you wish to do go back to the "normal" naming convention, (using the xCAT node name as the NIM machine name), you could run the **mkdsklsnode** command for the same node without the "-n" option. For example, in the previous example you got a NIM machine definition called "node42_61cosi". If you wish to switch back to a NIM machine name of "node42" for the next update you could run **mkdsklsnode** as follows. (You may need the "-f" (force) option if the "node42" definition already exists.)

mkdsklsnode -f -i 611cosi node42

4.6 Verify node readiness (optional)

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

In preparation for the network boot the NIM "dkls_init" operation configures *bootp*. At this point you can verify that the /etc/bootptab file has an entry for each node you wish to boot. Also, it is recommended that you stop and restart the inetd service to ensure the new *bootp* configuration is loaded:

stopsrc -s inetd

startsrc -s inetd

4.7 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

4.8 Verify the deployment

- As soon as the **rnetboot** command returns 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. To configure conserver, run:

  makeconservercf

  Kill the conserver daemon if it is running, and restart it:

  conserver &

  (You may need to add /opt/freeware/bin and /opt/freeware/sbin to your PATH first).

  To start a console:

  rcons node01

- 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 lpars, 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 filesystem 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. Try the "short" approach first:

  nim -F -o reset node01

  nim -o dkls_init node01

  rnetboot -f node01
- If that doesn't work, you may need to delete the entire client definition from NIM and recreate it:

  nim -F -o reset node01

  nim -o deallocate -a root=root -a paging=paging -a dump=dump -a spot=61cosi node01

  nim -o remove node01

  mkdsklsnode -i 61cosi node01

  rnetboot -f node01

5.0 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.

5.1 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.

5.2 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.

6.0 Notes

6.1 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.

6.2 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.

6.3 NIM Commands

6.3.1 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.

6.3.2 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.