

xCAT 2 on AIX

Installing AIX standalone nodes (using standard NIM rte method)

Version: 1.0

Date: 6/27/2008

<u>1.0 Installing AIX nodes (using standard NIM rte method).....</u>	<u>1</u>
<u> 1.1 Create an operating system image.....</u>	<u>2</u>
<u> 1.2 Create additional NIM network definitions (optional)</u>	<u>3</u>
<u> 1.3 Define xCAT networks.....</u>	<u>3</u>
<u> 1.4 Define the HMC as an xCAT node.....</u>	<u>4</u>
<u> 1.5 Discover the LPARs managed by the HMC.....</u>	<u>4</u>
<u> 1.6 Define xCAT cluster nodes.....</u>	<u>5</u>
<u> 1.7 Define xCAT groups (optional).....</u>	<u>5</u>
<u> 1.8 Set up customization scripts (optional).....</u>	<u>6</u>
<u> 1.9 Create NIM client & group definitions.....</u>	<u>6</u>
<u> 1.10 Initialize the AIX/NIM nodes.....</u>	<u>7</u>
<u> 1.11 Initiate a network boot.....</u>	<u>8</u>
<u> 1.12 Verify the deployment.....</u>	<u>8</u>
<u>2.0 Cleanup.....</u>	<u>9</u>
<u> 2.1 Removing NIM machine definitions.....</u>	<u>9</u>
<u> 2.2 Removing NIM resources</u>	<u>10</u>

1.0 Installing AIX nodes (using standard NIM rte method)

This “How-To” illustrates how AIX “standalone” machines may be installed using the NIM “rte” method.

The process uses xCAT features to automatically run the necessary NIM commands.

NIM 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 install an AIX standalone node using the NIM “rte” installation method and is not meant to be a comprehensive guide of all the available 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.
- The cluster network is set up.
- xCAT and prerequisite software has been installed on the management node.
- LPARs have already been created using the HMC interfaces.

1.1 Create an operating system image

Use the xCAT **mknimimage** command to create an xCAT *osimage* definition as well as the required NIM installation resources.

An xCAT *osimage* definition is used to keep track of a unique operating system image and how it will be deployed.

In order to use NIM to perform a remote network boot of a cluster node the NIM software must be installed, NIM must be configured, and some basic NIM resources must be created.

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

When you run the command you must provide a source directory for the required software. For the initial setup this is typically the product media (ex. /dev/cd0) but it could also be the name of an existing “lpp_source” resource.

In this example we need resources for installing a NIM “standalone” type machine using the NIM “rte” install method. (This type and method are the defaults for the **mknimimage** command but you can specify other values on the command line.)

For example, to create an *osimage* named “610image” and the required NIM resources you could run the following command.

```
mknimimage -s /dev/cd0 610image
```

(Creating the NIM resources could take a while!)

By default the command will create NIM *lpp_source*, *spot*, *bosinst_data*, and *resolv_conf* resources. 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 an *installp_bundle* resource named *mysshbndl* you could run the command as follows.

```
mknimimage -s /dev/cd0 610image installp_bundle= mysshbndl
```

When the command completes it will display the *osimage* definition which will contain the names of all the NIM resources that were created. The naming convention for the NIM resources that are created is the *osimage* name followed by the NIM resource type, (ex. “610image_lpp_source”), except for the SPOT name. The default name for the SPOT resource will be the same as the *osimage* name.

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.

In some cases you may also want to modify the contents of the NIM resources. For example, you may want to change the *bosinst_data* file or add to the *resolv_conf* file etc. For details concerning the NIM resources refer to the NIM documentation.

You can list NIM resource definitions using the AIX **lsnim** command. For example, if the name of your SPOT resource is "610image" then you could get the details by running:

```
lsnim -l 610image
```

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 610image
```

Note: The **mknimimage** command will take care of the NIM installation and configuration automatically, however, you can also do this using the standard AIX support. See the AIX documentation for details on using the **nim_master_setup** command or the SMIT "eznim" interface.

1.2 Create additional NIM network definitions (optional)

For this example we are assuming that the xCAT management node and the LPARs are all connected to 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 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. See the NIM documentation for details on creating additional network and route definitions.

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.

In our example we will assume 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: The xCAT definition should correspond to the NIM network definition. If multiple cluster subnets are needed then you will need an xCAT and NIM network definition for each one.

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 partitions are already created using the standard HMC interfaces.

Use the **rscan** command to gather the LPAR information. This command can be used to display the LPAR information in several formats and can also write the LPAR information directly to the xCAT database. 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.

Note: The stanza file will contain stanzas for things 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.

The updated stanza file might look something like the following.

```
Server-9117-MMA-SN10F6F3D:
objtype=node
nodetype=fsp
id=5
model=9118-575
serial=02013EB
hcp=hmc01
pprofile=
```

```
parent=Server-9458-10099201WM_A
groups=fsp,all
mgt=hmc
```

```
node01:
  objtype=node
  nodetype=lpar,osi
  id=9
  hcp=hmc01
  pprofile=lpar9
  parent=Server-9117-MMA-SN10F6F3D
  groups=all
  mgt=hmc
```

```
node02:
  objtype=node
  nodetype=lpar,osi
  id=7
  hcp=hmc01
  pprofile=lpar6
  parent=Server-9117-MMA-SN10F6F3D
  groups=all
  mgt=hmc
```

Note: The **rscan** command supports an option to automatically create node definitions in the xCAT database. To do this the LPAR name gathered by **rscan** is used as the node name and the command sets several default values. If you use the “-w” option make sure the LPAR name you defined will be the name you want used as your node name.

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 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 -t node -p -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.

1.8 Set up customization scripts (optional)

xCAT supports the running of user-provided customization scripts on the nodes when they are installed.

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.

1.9 Create NIM client & group definitions

You can use the xCAT **xcat2nim** command to automatically create NIM machine and group definitions based on the information contained in the xCAT node and group definitions. By doing this you synchronize the NIM and xCAT names so that you can use the same target names when running either an xCAT or NIM command.

To create NIM machine definitions you could run the following command.

```
xcat2nim -t node -o aixnodes
```

To create NIM group definitions you could run the following command.

```
xcat2nim -t group -o aixnodes
```

To check the NIM definitions you could use the NIM **lsnim** command or the xCAT **xcat2nim** command. For example, the following command will display the NIM definitions of the nodes: *node01*, *node02*, and *node03* (from data stored in the [NIM database](#)).

```
xcat2nim -t node -l -o node01-node03
```

1.10 Initialize the AIX/NIM nodes

You can use the xCAT **nimnodeset** command to initialize the AIX standalone nodes. This command uses information from the xCAT *osimage* definition and default values to run the appropriate NIM commands.

For example, to set up all the nodes in the group “*aixnodes*” to install using the *osimage* named “*610image*” you could issue the following command.

```
nimnodeset -i 610image aixnodes
```

To verify that you have allocated all the NIM resources that you need you can run the “**lsnim -l**” command. For example, to check node “*node01*” you could run the following command.

```
lsnim -l node01
```

The command will also set the “*profile*” attribute in the xCAT node definitions to “*610image* “. Once this attribute is set you can run the **nimnodeset** command without the “*-i*” option.

Note:

In preparation for the network boot, NIM configures **bootp**. Once the **nimnodeset** command completes 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
```

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 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 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*

2.0 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.

2.1 Removing NIM machine definitions

Use the xCAT **xcat2nim** 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.

```
xcat2nim -t node -r node01
```

The **xcat2nim** 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.

2.2 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 “610image” *osimage* definition along with all the associated NIM resources run the following command.

```
rmnimimage 610image
```

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