# xCAT 2 on AIX

# Using xCAT Service Nodes with AIX

# FIRST DRAFT

Version: 1.0

Date: 3/27/2009

# 1.0 Overview

**Note: The support described below requires xCAT version 2.2 or greater.**

In an xCAT cluster the single point of control is the xCAT *management node*. However, in order to provide sufficient scaling and performance for large clusters, it may also be necessary to have additional servers to help handle the deployment and management of the cluster nodes. In an xCAT cluster these additional servers are referred to as *service nodes*.

For an xCAT on AIX cluster the there is a primary NIM master which is typically on the management node. The service nodes are configured as additional NIM masters. All commands are run on the management node. The xCAT support automatically handles the setup of the low level service nodes and the distribution of the NIM resources. All installation resources for the cluster are managed from the primary NIM master. The NIM resources are automatically replicated on the low level masters when they are needed.

You can set up one or more service nodes in an xCAT cluster. The number you need will depend on many factors including the number of nodes in the cluster, the type of node deployment, the type of network etc. (As a general "rule of thumb" you should plan on having at least 1 service node per 128 cluster nodes.)

A service node may also be used to run user applications in most cases.

An xCAT service node must be installed with xCAT software as well as additional prerequisite software. See the section "Contents of xCATaixSN.bnd and xCATaixSSH.bnd" below for a list of the software that is required for AIX service nodes.

When using service nodes you must switch to a database that supports remote access. As a convenience, version 5.0 of the MySQL database is included in the xcat-mysql-<version>.tar.gz tar file that is available from the xCAT download site. This and other versions of MySQL are also available directly from the MySQL web site. (See http://dev.mysql.com/downloads/mysql/5.1.html#aix)

This document describes the process for setting up AIX service nodes and using them to deploy cluster nodes. The cluster nodes will be System p logical partitions.

AIX service nodes must be diskfull systems. Diskless xCAT service nodes are not currently supported for AIX.

In the process described below the service nodes will be deployed using a standard AIX/NIM "rte" network installation. If you are using multiple service nodes you may want to consider creating a "golden" **mksysb** image that you can use as a common image for all the service nodes. See the xCAT document named "Cloning AIX nodes (using an AIX mksysb image)" for more information on using **mksysb** images. (http://xcat.svn.sourceforge.net/svnroot/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT2onAIXmksysb.pdf)

In this document it is assumed that the cluster nodes will be diskless. The cluster nodes will be deployed using a common diskless image. It is also possible to deploy the cluster nodes using "rte" or "mksysb" type installs.

Before starting this process it is assumed you have configured an xCAT management node by following the process described in the "xCAT2onAIX" overview document. (http://xcat.svn.sourceforge.net/svnroot/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT2onAIX.pdf)

The process described below also assumes that System p LPARs for the cluster have already been defined using the standard HMC interfaces. At a minimum you need to define the partitions to use for your diskfull service nodes and at least one diskless partition (per CEC). The additional diskless partitions can be defined later in the process using the xCAT **mkvm** command if desired.

When defining the LPARs there are a couple things to keep in mind that may help with the rest of the cluster deployment.

1. When configuring an LPAR it would be advisable to use the host name you will want to use for the node as the LPAR name. When you run the xCAT **rscan** command in the process described below the default host name used for the node definitions will be the LPAR names.
2. It would also be advisable to use an LPAR naming convention that is representative of the cluster organization. For example, you might want to use "sn" in the names of the service node LPARs or "cn" in the cluster node LPARS or some other naming convention that helps keep track of a nodes characteristics etc.

# 2.0 Additional configuration of the management node

## *2.1 Cluster network configuration notes*

TBD

- The cluster network topology, naming conventions etc. should be carefully planned before beginning the cluster node deployment.
- XCAT requires an Ethernet network for installing and managing cluster nodes.
- Cluster nodes may be on different subnets.
- The cluster nodes must all have unique short host names to use in the xCAT node definitions.
- All cluster nodes must use the same domain name.
- The management node interfaces that will be used to manage the nodes should be configured before starting the xCAT deployment process.
- XCAT network definitions will have to be created for each unique subnet used in the cluster. (This is a step in the process described below.)
-

## *2.2 Choose the shell to use in the cluster (optional)*

By default the xCAT support will automatically set up **rsh** on all AIX cluster nodes. If you wish to use **ssh** you should modify the cluster site definition. To use **ssh** you would have to set the "useSSHonAIX=yes". You can also specify a path for the **ssh** and **scp** commands by setting the "**rsh**" and "**rcp**" . If not set the default path would be "/usr/bin/ssh" and "/usr/bin/scp".

For example:

***chdef -t site useSSHonAIX=yes***

You will also have to make sure that the **openssl** an **openssh** software is installed on your nodes. This will be explained in more detail below.

## *2.3 Configuring name resolution (optional)*

Name resolution is required by xCAT. You can use a simple /etc/hosts mechanism or you can optionally set up a DNS name server.

### 2.3.1 Add cluster nodes to the /etc/hosts file

There are many ways to get entries for all the cluster nodes in the /etc/hosts file.

These include:

- Manually adding the entries.
- Running a custom script that uses some cluster naming convention to automate the adding of the node entries. (User-provided.)
- Using the xCAT **makehosts** command after the XCAT node definitions have been created.

If you are dealing with a large number of nodes this task can be quite tedious. There is another option available in the xCAT support. This process uses a regular expression to automatically determine the IP addresses and hostnames for a set of nodes. To use this method you must decide on an appropriate naming conventions and IP address ranges. This process may seem a bit complicated but once you get things set up it can save time and add structure to your cluster.

If you choose to use this process you will have to come back to this section after you have created the xCAT node definitions later in this process. You should read through this now so that you can be prepared later.

As an example, suppose we decide on a node naming convention that includes the hardware frame number, the CEC number and the partition number. (Say "clstrf01c01p01" etc.) Also, lets say that the IP addresses would look something like "100.1.1.1" where the second number is the frame number, the third is the CEC number and the forth is the partition number.

With this example we can define a regular expression that, given a node name, could be used to derive a corresponding IP address and long hostname.

To have this regular expression applied to each node you can make use of the xCAT node group support. Let's say that all your cluster nodes belong to the group "compute". I can add the following values to the "compute" group definition.

***chdef -t group -o compute ip='|clstrf(\d+)c(\d+)p(\d+)|10.($1+0).($2+0).($3+0)|' hostnames='|(.*)|($1).cluster.com|'***

This basically says that for any node in the "compute" group the "ip" can be dervied by the regular expression *'|c906f(\d+)c(\d+)p(\d+)|10.($1+0).($2+0).($3+0)|'* , and the hostname can be derived from the expression *|(.*)|($1).cluster.com|'*.

So let's say that you have defined all your nodes using the xCAT support such as **rscan** or **mkvm** using the naming convention mentioned above. Now you could display the node definition as follows:

*lsdef -l clstrf01c02p03*

Since this node belongs to the "compute" group, when I display the definition it will use the regular expressions to derive the "ip" and "hostnames" values.

The output might look something like the following:

*Object name: clstrf01c02p03*
*cons=hmc*
*groups=lpar,all,compute*
*hcp=clstrhmc01*

*hostnames=clstrf01c02p03.cluster.com*
*id=1*
*ip=10.1.2.3*
*mac=001a64f9c009*
*mgt=hmc*
*nodetype=lpar,osi*
*os=AIX*
*parent=clstrf1fsp01-9125-F2A-SN024C332*
*postscripts=myscript*
*profile=MYimg*

Now that all the nodes have an "ip" and "hostnames" value you can run the xCAT **makehosts** command to update /etc/hosts.

***makehosts compute -l***

### 2.3.2 Set up a DNS nameserver

To set up the management node as the DNS name server you must set the "domain", "nameservers" and "forwarders" attributes in the xCAT "site" definition.

For example, if the cluster domain is "mycluster.com", the name of the management node is "mn20" and the site DNS servers are "9.14.8.1,9.14.8.2" then you would run the following command.

***chdef -t site domain= mycluster.com nameservers= mn20 forwarders= 9.14.8.1,9.14.8.2***

Edit "/etc/resolv.conf" to contain the cluster domain and nameserver. For example:

```
search mycluster.com
nameserver mn20
```

Create xCAT network definitions for all the the cluster networks. (Your network and mask value need to be defined for **makedns** to be able to set up the correct ip range for the management node to serve.)

Run **makedns** to create the /etc/named.conf file and populate the /var/named directory with resolution files.

***makedns***

Start DNS:

***startsrc -s named***

## *2.4 Check system services*

- **inted**

inetd includes services such as telnet, ftp, bootp, and others. Edit the /etc/inetd.conf file to turn on all services that are needed. Ftp and bootp are required for pSeries node installations. Stop and restart the **inetd** service after any changes:

*stopsrc -s inetd*
*startsrc -s inetd*

- **NFS**

  NFS is required for all NIM installs. Ensure the NFS daemons are running:

  *lssrc -g nfs*

  If any NFS services are inoperative, you can stop and restart the entire group of services:

  *stopsrc -g nfs*
  *startsrc -g nfs*

  There are other system services that NFS depends on such as inetd, portmap, biod, and others.

- **TFTP**

  To check if the TFTP daemonis running.

  *lssrc -a | grep tftpd*

  To stop and start tftp daeon.

  *stopsrc -s tftpd*
  *startsrc -s tftpd*

### *2.5 Switch to the MySQL database*

The xCAT support for Service Nodes requires a database with remote access capabilities. The MySQL database is provided for this purpose.

See the "xCAT 2.1 MySQL Setup" document for details on installing, configuring and migrating to MySQL.

http://xcat.svn.sourceforge.net/svnroot/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT2.SetupMySQL.pdf

## 3.0 Install the Service Nodes

### *3.1 Define the HMCs as xCAT nodes*

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***

## 3.2 Discover the LPARs managed by the HMCs

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***

The file will contain stanzas for all the LPARs that have been configured as well as some additional information that must also be defined in the xCAT database. It is not necessary to modify the non-LPAR stanzas in any way.

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.

Since we are using service nodes there are several values that must be set for the node definitions. **You can set these values later, after the nodes have been defined, or you can modify the stanzas to include the values now.**

- Add "service" to the "groups" attribute of all the service nodes.
- Add the name of the service node for a node to the node definition. (Ex. "servicenode=xcatSN01") If this is not set the default service node is the xCAT management node. This is the name of the service node as it is known by the management node.
- If the node is on a different subnet you must also set the "xcatmaster" attribute. This must be the name of the service node as it is known by the node.
- The "setupnameserver" attribute must be explicitly set to "yes" or "no". (Ex. "setupnameserver=no")

The updated stanzas might look something like the following.

```
cl2sn01:
    objtype=node
    nodetype=lpar,osi
    id=9
    hcp=hmc01
    pprofile=lpar9
    parent=Server-9117-MMA-SN10F6F3D
    groups=all,service
    setupnameserver=no
```

```
    mgt=hmc

cl2cn27:
    objtype=node
    nodetype=lpar,osi
    servicenode=cl2sn01
    id=7
    hcp=hmc01
    pprofile=lpar6
    parent=Server-9117-MMA-SN10F6F3D
    groups=all
    mgt=hmc

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
```

**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.

## *3.3 Define xCAT service 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. The stanza file will include any LPARs that were already created so it may include nodes other than just the service nodes.

***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 additional updates to the definitions, if needed.

Make sure the node attributes required for service nodes support are set as mentioned in the previous step.

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

- **TBD**

## 3.5 Create a Service Node 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 already 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 NIM "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 "610SNimage" and the required NIM resources you could run the following command.

***mknimimage -s /dev/cd0 610SNimage***

(Creating the NIM resources could take a while!)

By default the command will create NIM *lpp_source*, *spot*, and *bosinst_data* 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:

*mknimimage -s /dev/cd0 610SNimage resolv_conf=my_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.

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. " *610SNimage_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 "*610SNimage*" then you could get the details by running:

*lsnim -l 610SNimage*

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 610SNimage*

**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.

## *3.6 Add required service node software*

An xCAT AIX service node must also be installed with additional software.

To simplify this process xCAT includes the required xCAT and open source software in the core-aix-*.tar.gz and dep-aix*.tar.gz tar files. You will also need the *openssl* and *openssh* software that was installed on the management node.

The required software must be copied to the NIM lpp_source that is being used for this OS image. The easiest way to do this is to use the "nim -o update" command.

For example, assume all the required software has been copied and unwrapped in the /tmp/images directory.

To add all the packages to our lpp_source resource, you can run the following:

***nim -o update -a packages=all -a source=/tmp/images 610SNimage_lpp_source***

The NIM command will find the correct directories and update the lpp_source resource.

To get this additional software installed we need a way to tell NIM to include it. To facilitate this xCAT provides two AIX installp bundle files. The files are included in the core-aix-*.tar.gz file.

To use the bundle files you need to define them as NIM resources and add them to the xCAT *osimage* definition.

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/xCATaixSN.bnd xCATaixSN***

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

To add these bundle resources to your xCAT osimage definition run"

***chdef -t osimage -o 610SNimage installp_bundle="xCATaixSN,xCATaixSSH"***

**Note:** Make sure the xCATaixSN comes first! There is a temporary issue with the AIX openssh installp package that requires that it be done in a separate bundle file that comes after the xCATaixSN bundle (which contains the openssl dependency.).

## *3.7 Add cluster resolv.conf file (optional)*

The xCAT deployment code will automatically handle the creation of an /etc/resolv.conf file on all the cluster nodes. If you want xCAT to handle this you should make sure the "domain" and "nameservers" attributes of the "site" definition are set.

For example:

***chdef -t site -o clustersite domain=mycluster.com nameservers= 100.240.0.1***

## *3.8 Set cluster root password (optional)*

You can have xCAT create an initial root password for the cluster nodes when they are deployed. To do this you must modify the xCAT "passwd" table.

You can use the **tabedit** command to add and entry to this table. For example:

*tabedit passwd*

You will need an entry with a *"key"* set to *"system"*, a *"username"* set to *"root"* and the *"password"* attribute set to whatever attribute you want.

You can change the passwords on the nodes at any time using **xdsh** and the AIX **chpasswd** command.

For example:

*xdsh node01 'echo "root:mypw" | chpasswd -c'*

## 3.9 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=9.114.0.0 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.

## 3.10 Create additional NIM network definitions (optional)

TBD

For this example 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.

- TBD – add example for NIM network setup

*nim -o define -t ent -a net_addr=10.0.0.0 -a snm=255.0.0.0 -a routing2='default 10.0.0.241' clstr_net*

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

*nim -o change -a routing2='master_net 10.0.0.241 c906mgrs1.ppd.pok.ibm.com' clstr_net*

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. (*IBM AIX Installation Guide and Reference*. http://www-03.ibm.com/servers/aix/library/index.html)

## 3.11 Define xCAT "service" group

If you did not already create the xCAT "service" group when you defined your nodes then you can do it now using the **mkdef** or **chdef** command.

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 to create the group called "*service*" with the members *sn01* and *sn02* you could run **chdef** as follows.

*chdef –t node -p –o sn01,sn02 groups=service*

The "-p" option specifies that "service" 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 service members="sn01,sn02"*

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

## 3.12 Include customization scripts

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

This support includes:

- The running of a set of customization scripts that are required by xCAT.

  You can see what scripts xCAT will run by default 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. The scripts are contained in the /install/postscripts directory on the xCAT management node.

- The optional running of customization scripts provided by xCAT.

  There is a set of xCAT customization scripts provided in the /install/postscripts directory that can be used to perform optional tasks such as additional adapter configuration.

- The optional running of user-provided customization scripts.

  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 definition to include the comma separated list of the scripts that you want to be executed on the nodes. The order of the scripts in the list determines the order in which they will be run. 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 -p postscripts=foo,bar*

     (The "-p" means to add these to whatever is already set.)

**Note: T**he customization scripts are run during the boot process (out of /etc/inittab).

### 3.12.1 Add "servicnode" script for service nodes

You must add the "servicenode" script to the *postscripts* attribute of all the service node definitions. For example, to add the "servicenode" postscript to all nodes in the group called "service" you could run the following command.

***chdef -p -t node service postscripts=servicenode***

### 3.12.2 Add NTP setup script (optional

**- TBD**

## *3.13 Gather MAC information for the install 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.

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

**makeconservercf**

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.

## *3.14 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 database. 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 for your service nodes you could run the following command.

***xcat2nim -t node service***

To create NIM group definition for the group "service" you could run the following command.

***xcat2nim –t group –o service***

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 contained in the xCAT group called "service", (from data stored in the NIM database).

*xcat2nim -t node -l service*

## *3.15 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 "*service*" to install using the *osimage* named "*610SNimage*" you could issue the following command.

***nimnodeset -i 610SNimage service***

To verify that you have allocated all the NIM resources that you need you can run the "**lsnim –l**" command. For example, to check all the nodes in the group "service" you could run the following command.

*lsnim -l service*

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

## *3.16 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.**

## 3.17 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 "*service*" you could issue the following command.

***rnetboot service***

**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*

## 3.18 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 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:
    - Check /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*

## 3.19 Configure additional adapters on the service nodes (optional)

**TBD** - use configeth? - w/updatenode – or do manaually ???

## 3.20 Verify Service Node configuration

During the node boot up there are several xCAT post scripts that are run that will configure the node as an xCAT service node. It is advisable to check the service nodes to make sure they are configured correctly before proceeding.

There are several things that can be done to verify that the service nodes have been configured correctly.

1. Check if NIM has been installed and configured by running "lsnim -a" or some other basic NIM command on the service node.
2. Check to see if all the additional software has been installed. For example, Run "rpm -qa" to see if the xCAT and dependency software is installed.
3. Try running some xCAT commands such as **lsdef** to see if the xctd daemon is running and if data can be retrieved from the xCAT database on the management node.
4. If using SSH for your remote shell try to ssh to the service nodes from the management node.

# 4.0 Install the cluster nodes

## *4.1 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.

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

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*

## 4.2 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.

## 4.3 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.

## 4.4 Define the LPARs currently managed by the HMC

This step assumes that 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

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.

## *4.5 Create and define additional cluster nodes*

!!!! need process – **TBD**

- use mkvm to create additional diskless partitions – per cec
- mkvm -V c906f01c01p02 -i 4 -n c906f01c01p04-c906f01c01p29
-

Make sure "servicenode" is set to the name of the service node as known by the management node and "xcatmaster" is set to the name of the service node as known by the node.

## *4.6 Gather MAC information for the node 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*

## *4.7  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 -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.

## *4.8 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.

## *4.9 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.

## 4.10 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*

## 4.11 Verify the deployment

- 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*

  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*

# 5.0 Contents of xCATaixSN.bnd and xCATaixSSH.bnd

These are AIX installp bundle files. They can be used with AIX and NIM commands to simplify the process of installing or copying specific sets of software. These two bundle files are included in the xCAT for AIX tar file that is available from the xCAT web site. The xCATaixSN.bnd file contains a list of all the software that is needed on an AIX service node. The xCATaixSSH.bnd file contains a list of openssh software that would be needed if you choose to use SSH as your remote shell.

## *5.1 xCATaixSN.bnd*

```
# Software needed for the xCAT on AIX Service Nodes

# installp packages
I:openssl.base
I:openssl.license
I:bos.sysmgt.nim.master

# RPMs
R:bash-3.2-1.aix5.2.ppc.rpm
R:conserver-8.1.16-2.aix5.3.ppc.rpm
R:expect-5.42.1-3.aix5.1.ppc.rpm
R:fping-2.4b2_to-1.aix5.3.ppc.rpm
R:perl-DBI-1.55-1.aix5.3.ppc.rpm
R:perl-Digest-MD5-2.36-1.aix5.3.ppc.rpm
R:perl-Expect-1.21-1.aix5.3.ppc.rpm
R:perl-IO-Socket-SSL-1.06-1.aix5.3.ppc.rpm
R:perl-IO-Stty-.02-1.aix5.3.ppc.rpm
R:perl-IO-Tty-1.07-1.aix5.3.ppc.rpm
R:perl-Net_SSLeay.pm-1.30-1.aix5.3.ppc.rpm
R:perl-Net-Telnet-3.03-1.2.aix5.3.ppc.rpm
R:tcl-8.4.7-3.aix5.1.ppc.rpm
R:tk-8.4.7-3.aix5.1.ppc.rpm
R:perl-xCAT-2.2*
R:xCAT-client-2.2*
R:xCAT-server-2.2*
R:xCAT-rmc-2.2*
R:perl-DBD-SQLite-1.13-1.aix5.3.ppc.rpm
R:net-snmp-5.4.2.1-1.aix5.3.ppc.rpm
R:net-snmp-devel-5.4.2.1-1.aix5.3.ppc.rpm
R:net-snmp-perl-5.4.2.1-1.aix5.3.ppc.rpm
```

```
# optional
R:perl-DBD-mysql-4.007-1.aix5.3.ppc.rpm
R:xcat-mysql-5.0-1.aix5.3.ppc.rpm
```

## 5.2 xCATaixSSH.bnd

```
# bundle file for openssh - installp package from AIX
#       Note - openssl must be installed before openssh

I:openssh.base
I:openssh.license
I:openssh.man.en_US
I:openssh.msg.en_US
I:openssh.msg.EN_US
```