xCAT 2 on AIX
Installing AIX standalone nodes
(using standard NIM rte method)

Date: 4/20/2009

1.0 Overview

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.
- The cluster network is configured. (The Ethernet network that will be used to perform the network boot of the nodes.)
- xCAT and prerequisite software has been installed and configured on the management node.
- Any logical partitions that will be used have already been created using the HMC interfaces.

2.0 Installing AIX nodes (using standard NIM rte method)

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

Note: If you wish to install and configure NIM manually you can run the AIX **nim_master_setup** command (Ex. "*nim_master_setup -a mk_resource=no -a device=<source directory>*").

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

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

/usr/bin/chuser fsize=-1 root

When you run the **mknimimage** 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 "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*, 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 610image 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. " *610image_lpp_source*"), except for the SPOT name. The default name for the SPOT resource will be the same as the *osimage* name.

Once the initial *osimage* definition is created you can change it by using the **chdef** command. For example, you may need to create additional NIM resources to use when installing the nodes, such as *script* or *installp_bundle* resources.

To add an *installp_bundle* resource to the *osimage* definition created in the previous example you could run the **chdef** command as follows.

chdef -t osimage -o 610image installp_bundle=mybundlename

The xCAT *osimage* definition can be listed using the **lsdef** 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 master 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.

2.2 Add SSH and requisite software (optional)

If you wish to use SSH as you remote shell then the SSH and it's prerequisite software must be installed on the nodes.

You will need the *openssl* and *openssh* software that was installed on the management node as well as several additional packages that are available in the dep-aix-<version>.tar.gz tar file.

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 610image_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-<version>.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 (xCATaixSSL.bnd and xCATaixSSH.bnd) to a location where they can be defined as a NIM resource, for example "/install/nim/installp_bundle".

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

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

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

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.).

These bundle files will be included in the underlying NIM commands that are called and NIM will include this additional software when installing the nodes.

2.3 Define xCAT networks

Create an xCAT 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.

2.4 Create additional NIM network definitions (optional)

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

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

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

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

Step 1

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

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

Step 2

Create a new interface entry for the NIM "master" definition. Assume that the next available interface index is "2" and the hostname of the NIM master is "xcataixmn".

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

Step 3

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

nim -o change -a routing2='master_net 10.0.0.247 8.124.37.24' clstr_net

Step 4

Verify the definitions by running the following commands.

lsnim -l master

lsnim -l master_net

lsnim -l clstr_net

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

2.5 Define the HMC as an xCAT node

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

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

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

2.6 Discover the LPARs managed by the HMC

This step assumes that the 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.

2.7 Define xCAT cluster nodes

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

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

cat mystanzafile | mkdef –z

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

2.8 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" (plus) 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 in the xCAT database.

2.9 Set up customization scripts (optional)

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

This support includes:

- The running of a set of default 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: The customization scripts are run during the boot process (out of /etc/inittab).

2.9.1 Add NTP setup script (optional)

To have xCAT automatically set up ntp on the cluster nodes you must add the **setupntp** script to the list of postscripts that are run on the nodes.

To do this you can either modify the "postscripts" attribute for each node individually or you can just modify the definition of a group that all the nodes belong to.

For example, if all your nodes belong to the group "compute" then you could add **setupntp** to the group definition by running the following command.

chdef -p -t group -o compute postscripts=setupntp

2.9.2 Add secondary adapter configuration script

It is possible to have additional adapter interfaces automatically configured when the nodes are booted. XCAT provides sample configuration scripts for both Ethernet and IB adapters. These scripts can be used as-is or they can be modified to suit you particular environment. The Ethernet sample is /install/postscript/configeth. When you have the configuration script that you want you can add it to the "postscripts" attribute as mentioned above. Make sure your script is in the /install/postscripts directory and that it is executable.

If you wish to configure IB interfaces please refer to: "xCAT 2 InfiniBand Support" **https://xcat.svn.sourceforge.net/svnroot/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT2IBsupport.odt**

2.10 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. In this case the first adapter that can be successfully used to ping the server will be written to the database.

Before running **getmacs** you must first run the **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.

2.11 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 you could run the following command.

xcat2nim -t node aixnodes

To create a NIM group definition called "*aixnodes*" 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

2.12 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 **nimnodeset** 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.

2.13 Open a remote console (optional)

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

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

To configure conserver run:

makeconservercf

To start a console:

rcons node01

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

2.14 Initiate a network boot

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

rnetboot aixnodes

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

chdef -t site -o clustersite ppctimeout=180

2.15 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, 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.

3.0 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.

3.1 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.

3.2 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.