xCAT2 Top Doc

04/05/2010, 10:36:12 AM

Table of Contents

1.0 Overview of xCAT

xCAT (Extreme Cluster Administration Tool) is a toolkit that provides support for the deployment and administration of large cluster environments.

Earlier versions of xCAT have been used to deploy and manage many high end Linux clusters since 1999. The new xCAT version 2.X is a complete rewrite of xCAT that includes many architectural changes and functional enhancements.

xCAT is a scalable distributed computing management and provisioning tool that provides a unified interface for hardware control, discovery, and OS diskful/diskless deployment. Now xCAT 2 is open source on the Source Forge Website , so you can use it with confidence and participate in making it even better.

1.1 xCAT Architecture

xCAT 2 is a complete rewrite of xCAT 1.3 that includes many architectural changes and functional enhancements. All commands are client/server, authenticated, logged and policy driven. XCAT 2 supports roll base authentication. The clients can be run on any OS with Perl, including Windows. All communication is SSL encrypted. The code has been completely rewritten in Perl, and table data is now stored in a relational database and with the plug-in architecture you can chose your database from SQLite, MySQL, PostgreSQL with more options coming.

In the xCAT client/server application, flow between the client and server is controlled by the xCAT daemon (xcatd) on the Management Node. When xcatd receives the command which has been packaged as (XML), it determines whether the sender has authority to execute the command by

evaluating the ACL's in the policy table. The daemon also receives status and inventory information from the nodes as they are deployed. See xCAT 2 Architecture for more details.

xCAT 2 was designed to scale for extremely large clusters. See xCAT on the world's fastest know supercomputer. With the Hierarchical support, a single management node may have any number of stateless or statefull service nodes to increase the provisioning throughput and management of the largest clusters. All cluster services such as LDAP, DNS, DHCP, NTP, Syslog, etc... can be automatically configured throughout the cluster. Outbound cluster management commands such as rpower, xdsh, xdcp, etc...utilize this hierarchy for scalable systems management.

Stateless, Statelite and Stateful Choices

Stateless nodes are an important concept in xCAT 2. A stateless node is defined as one that has no "state" (configuration changes, software updates, etc.) stored permanently on it. This is extremely useful in a cluster for the following reasons:

- All nodes will have a much greater likelihood of staying consistent. And if the administrator does suspect that a node is out of sync with the rest of the cluster, they can simply reboot it and know that it is back in its original, pristine state.
- If a node experiences a hardware problem, the hardware can be pulled from the rack and replaced with new hardware and the node booted again and it will come up with the same state as before.
- In a provisioning environment, new nodes can be provisioned or moved without the worry of them losing state.

For Linux clusters, xCAT 2 provides the choice of either stateless or stateful nodes. A stateful node is one that has the OS installed on its local hard disk and therefore, changes to the node (configuration changes, software updates, etc.) can be made over time and those changes will persist.

Stateless nodes in xCAT 2 are implemented by not putting the OS on the local disk of the node. There are 3 choices for stateless:

1. **RAM-root** – The entire OS image is contained in a RAM file system that is sent to the node when it boots. Typical size for a minimal compute node for Linux is 75-160 MB of memory.
2. **Compressed RAM-root** – The OS image is in a compressed tar file. Individual files are extracted and cached when read. File writes are done to the cached copy. Typical size for a minimal compute node for Linux is 30-64 MB of memory.
3. **NFS Hybrid** – This is more accurately called NFS-root with copy-on-write. A minimal boot kernel is sent to the node, which readonly NFS mounts the OS image from the server. Files read are cached in memory. File writes are done to the cached copy. Typical size for a minimal compute node for Linux is 5 MB of memory.

Statelite (xCAT 2.4 and later) provides an efficient and flexible diskless solution ,because most of the OS image is NFS mounted read-only, but a configurable list of directories and files can be read-write. The read-write files can either be persistent across reboots, or volatile (restoring to pristine state after reboot). There are advantages and disadvantages over Stateless, which are outlined in the xCAT Statelite Cookbook.

Defining Hostnames in xCAT

xCAT hostnames should not contain mixed case. Name resolution (DNS) will resolve all names to lower case, so all hostnames defined in xCAT should be lowercase.

The preferred method of setting up xCAT is to use short hostnames (without the domain) for all nodes named in the database. If you must use long hostnames, then presently you are required to create an xCAT "alias" for the node by assigning the node named by long hostname to a node group that is named as the short hostname.

For example an entry in the nodelist table would look like the following:
"testnode.cluster.net","testnode,compute",,,,,

1.2 xCAT Features

Features provided by xCAT for AIX or Linux clusters include the following:

- Deploying diskless and diskfull nodes.
- Node group support.
- Node discovery.
- Operating system image management.
- Support for user-provided customization scripts. xCAT supports the automatic running of user-provided customization scripts on the nodes when they are deployed (installed or booted).
- xCAT data store in plug-in relational database (SQLite, MySQL,Postgresl, TBD)
- Hardware control commands for discovering hardware, gathering MAC addresses, VPD, and environments, power control, initiating a network boot, and LPAR creation/deletion.
- Hierarchical support to allow large system to distribute the management of the cluster to service nodes.
- Remote console support.
- Parallel remote shell and remote copy commands.
- rsync support.
- Monitoring plug-in infrastructure (RMC, Ganglia)
- Notification infrastructure which lets users monitor xCAT database table changes.
- Predefined conditions, responses and sensors.
- Software and firmware inventory
- InfiniteBand configuration
- Xen support
- Windows support
- GUI for initial cluster setup
- Allow continuous operation during cluster software updates using plug-in job scheduler (LoadLeveler, Moab).
- Automatic setup for syslog, remote shell, DNS, DHCP, and ntp for both the xCAT management node and the cluster nodes.
- Documentation and "**man**" pages.

1.3 xCAT license

xCAT 2 Open Source License: Eclipse Public License

1.4 xCAT support

Support for xCAT may now be purchased. Other support is offered through the xCAT public mailing list. You can also open a new feature request or a bug report using Tracker on Source Forge. Check out the Support section on the xCAT Source Forge Home page.

1.5 Syslog and Auditing

Syslog

xCAT automatically sets up syslog during the install on the MN and the nodes. It use local4 to register it syslog entries and it logs to the /var/log/messages file.

The syslog postscript accomplishes the setup. The syslog script is also called during initial install by xcatconfig on the MN. The xCAT daemon will detect, if the syslog daemon is not running and restart it when it starts.

The setup of the service nodes and compute nodes is to send all syslog messages to the Management Node (MN) to the /var/log/messages file.

xCAT messages all start with xCAT: They can easily be extracted from the messages file by grepping for xCAT: in the /var/log/messages file on the MN.

Some of the important logging for auditing:

- The xcatd daemon, logs all commands run, and who runs them.

  A typical entry is as follows:

  Feb 23 09:27:30 xcat20RRmn xCAT: xCAT: Allowing tabdump for root from localhost.localdomain

  You can grep for the following string in /var/log/messages on the MN, to obtain those entries:

  ```
  fgrep "xCAT: Allowing" /var/log/messages
  ```

  You can also grep for all denials of requests:

  ```
  fgrep "xCAT: Denying" /var/log/messages
  ```

- The xcatd daemon, logs startup errors and database access errors.
- Errors from postscript running on nodes, and service nodes are logged in the syslog on the MN.

Additonal setup

You can change the syslog setup to suit your needs. See the man page for the syslog software you are using.

On AIX, xCAT uses syslog:

xCAT sets up the log to create up to 5 files of 1meg each and then rotate. See /etc/syslog.conf.

```
# xCAT settings
*.debug   /var/log/messages  rotate 1024K files 5
```

On SLES xCAT uses syslog_ng:

The setup of xCAT used in syslog_ng, is basically the defaults of the syslog_ng software on the MN. It only changes to allow syslogs to be forwarded to the MN from the nodes.
This setup allows local messages for (local1-7) to be logged to /var/log/localmessages and all messages to /var/log/messages. In many cases you will find that /var/log/localmessages contain only or primarily xCAT messages, so that is the file to check

When syslog_ng is installed, it sets itself up to use logrotate to rotate the generated logs. See man logrotate for more details.

On RedHat xCAT user rsyslog:

xCAT uses basic defaults for rsyslog. See http://linux.die.net/man/8/rsyslogd for more information.

When rsyslog is installed, it sets itself up to use logrotate to rotate the generated logs. See man logrotate for more details.

On a node or service node:

xCAT directs syslog to send all syslog messages to the Management Node.

For example on RedHat:
This entry in /etc/rsyslog.conf, directs syslog to forward all messages to the ip address, which is the ip of the MN.

```
# xCAT settings
*.* @7.113.44.250
```

auditlog

As of xCAT 2.4, xCAT will not only log all command xCAT commands run and who runs them to syslog, but also to a new auditlog table in the xCAT database. See the manpage for the auditlog table.

2.0 Installing an xCAT Management Node

2.1 Install an AIX Management Node

Refer to the document titled Creating an xCAT Management Node for details on how to install an xCAT on AIX management node.

2.2 Install a Linux Management Node

To install the xCAT Management Node (MN), the following steps are taken:

1. Install the MN with the OS. Install all rpms from the OS. You will probably be missing dependencies for xCAT later, if you do not.
2. Configure Cluster-Facing NICS
3. Configure NTP
4. Configure Hostname
5. Configure DNS or some hostname resolution method.
6. Setup basic /etc/hosts file
7. Configure Ethernet switches

***Convention: In order to distinguish the operation among different OS or distros, the mark likes* [OS +distro+release] will be used before an operation to specify the applicable range of the operation.**
[RH] – applicable for all releases of RHEL and Fedora
[RHEL] – applicable for distro RHEL
[FEDORA] – applicable for distro Fedora
[SLES] – applicable for distro SLES
[SLES11] – applicable for distro SLES11

Install the OS of Management Node

xCAT 2 management nodes are supported on the following Linux distributions:
[RHEL]: rhels5.2, rhels5.3
[FEDORA]: fedora8, fedora9
[SLES]: sles10, sles11

Setup Your Networks

xCAT install process will scan and populate certain settings from the running configuration. Having the networks configured ahead of time will aid in correct configuration. All the networks in the cluster must be defined in the xCAT networks table before start installing cluster nodes. When xCAT is installed, it will run makenetworks to create an entry in the network table for each of the networks the management node is on. Additional network configuration can be added to the xCAT networks table manually later if needed.

Install the prerequisite packages for Management Node

As mentioned earlier, it is recommended to install all the RPMs on the management node from the OS repository, in case not all the RPMs are installed, we need to make sure that dhcp, bind (not bind-chroot), expect, httpd, nfs-utils, vsftpd, and perl-XML-Parser were installed before installing the xCAT. Please be aware that the packages names may vary slightly between different operating systems, setting up the operating system repository for YUM/zypper is a good practice, when the operating system repository for YUM/zypper is setup correctly, these xCAT prerequisite packages will be installed automatically when installing the xCAT.

[RH] Disable the SELinux

The RedHat SELinux (Secure Enterprise Linux) must be disabled on the xCAT 2 management node, various xCAT 2 functions will be affected if the SELinux feature is enabled.
One method for disabling the SELinux is the following:
1.In the /etc/selinux/config file, set SELINUX=disabled.
2.Reboot the node.

[RH] Disable the Firewall

The management node provides many services to the cluster nodes, but the firewall on the management node can interfere with this. If your cluster is on a secure network, the easiest thing to do is to disable the firewall on the management node:

```
service iptables stop
chkconfig iptables off
```

If disabling the firewall completely isn't an option, configure iptables to allow the following services on the NIC that faces the cluster: DHCP, TFTP, NFS, HTTP, FTP, DNS.

Configure Cluster-Facing NICS

The cluster-facing NICs should be configured before installing xCAT 2 management node. There are a number of methods to configure the NICs such as updating Linux ifcfg-* files, using Linux ifconfig command or Linux Graphic User Interface. Here is an example of using ifcfg-* files to configure the NICs:

The location of the ifcfg-* files:
[RH]: `/etc/sysconfig/network-scripts/ifcfg-*`
[SLES]: `/etc/sysconfig/network/ifcfg-*`

An example of the content of ifcfg-* file:

```
DEVICE=eth1
ONBOOT=yes
BOOTPROTO=static
IPADDR=11.16.0.1
NETMASK=255.255.0.0
```

Prevent DHCP client from overwriting DNS configuration

If the management node will be used to install any service node or compute node, the DHCP service needs to be configured on the management mode through the cluster-facing network interfaces, to prevent DHCP client from overwriting DNS configuration, the line “PEERDNS=no” is needed in the ifcfg-* files relevant to any cluster-facing network interfaces .
[RH]: `/etc/sysconfig/network-scripts/ifcfg-*`
[SLES]: `/etc/sysconfig/network/ifcfg-*`

Configure Hostname

The xCAT 2 management node hostsname must be configured before setting up the xCAT 2 mangement node. If the management node hostname is already configured, this step can be skipped. The Linux command “hostname” or the Linux GUI can be used to configure the hostname. Editing the hostname configuration file /etc/sysconfig/network on RedHat systems or /etc/HOSTNAME on SuSE systems is another method.
[RH]: Edit the /etc/sysconfig/network and add entry 'HOSTNAME=(desired hostname)'.

[SLES]: Edit the /etc/HOSTNAME and add the proper hostname.

Setup the TimeZone

When using management node to install service nodes or compute nodes, the timezone configuration on the management node will be inherited by the service nodes or compute nodes. So it is recommended to setup the correct timezone on the management node.

Configure DNS Resolution

/etc/resolv.conf should contain the DNS for management node itself.

```
search cluster
nameserver 11.16.0.1
```

Set up basic hosts file

Ensure the /etc/hosts has the entries of localhost and management node itself.

```
127.0.0.1          localhost.localdomain localhost
::1                localhost6.localdomain6 localhost6
11.16.9.1          xcat20mn.clusters.com xcat20mn
11.16.9.2           .....
```

Note: Name resolution is required by xCAT 2. You can use a simple /etc/hosts mechanism or you can optionally set up a DNS name server. You can refer to section 4.3 **Configuring name resolution (optional)** in xCAT2onAIX for more details on how to setup DNS resolution and /etc/hosts.

Restart Management Node (optional)

Though it is possible to restart the correct services for all settings except SELinux, the simplest step would be to reboot the management server at this point.

Configure Ethernet Switches (optional)

xCAT can use the ethernet switches for automatic hardware discovery and initial configuration, this is useful especially for large clusters. In general, this requires that the user in advance set up an ip address and basic snmp functionality. Allowing the snmp version 1 community string “public” read access will allow xCAT to communicate without further customization. It is also recommended that spanning tree be set to portfast or edge-port for faster boot performance. Please see the relevant switch documentation as to how to configure these items.

[RH] Remove the package tftp-server

xCATships the package atftp* package with its dependencies. This package conflicts with the tftp-server that is by default installed with the OS.
In order to install the xCAT successfully, you have to remove any tftp-server package that is installed first.

```
yum erase tftp-server.*
```

or if tftp-server depended by other packages:

```
rpm -e --nodeps tftp-server
```

[RH] Remove the package OpenIPMI-tools

The xCAT software ships its own ipmi-tools* package with its dependencies. This package conflicts with the OpenIPMIs-tool packages of the OS, which must be removed before xCAT is installed.

Download and Install xCAT 2 From an MN That Has Internet Access

XCAT can be downloaded and installed using YUM on RedHat systems or zypper on SuSE systems. To download and install xCAT2 on RedHat systems, go to the xCAT Download site in the section titled "**RPMs in directories, structured for YUM download** ". This allows you to get a repo file to put on your management node that will point YUM to the proper place to get the xCAT RPMs. Then follow the instructions in section "**Open Source Package Dependencies that xCAT Requires"** to download the correct repo file for the xcat-dep packages**.**

For SLES systems, just simply use the zypper command to add the xCAT download site into the zypper repository and then use the zypper command to download and install xCAT2.
Note: disable any firewalls that could prevent the download.

Note:We have chosen ppc64 as our architecture and sles10, modify to suit your needs.

2.2.1 [RH] Setup Yum

Download Repository Files into the /etc/yum.repos.d.

```
cd etc/yum.repos.d
wget http://xcat.sourceforge.net/yum/xcat-core/xCAT-core.repo
wget http://xcat.sourceforge.net/yum/xcat-dep/<release>/<arch>/xCAT-dep.repo
```

2.2.2 [SLES] Setup Zypper

[SLES11]:

```
zypper ar -t rpm-md http://xcat.sf.net/yum/xcat-core xCAT-core
zypper ar -t rpm-md http://xcat.sf.net/yum/xcat-dep/<release>/<arch> xCAT-dep
```

[SLES10.2+]:

```
zypper sa -t rpm-md http://xcat.sf.net/yum/xcat-core xCAT-core
zypper sa -t rpm-md http://xcat.sf.net/yum/xcat-dep/<release>/<arch> xCAT-dep
```

Download and Install xCAT 2 For an MN That Does Not Have Internet Access

Go to the xCAT Download site. Download the level of xCAT tarball you desired. Go to the xCAT Dependencies Download page and get the latest xCAT dependency tarball.

2.2.1 Download the tarballs

Download and copy the files to the Management Node (MN) and untar them:

```
mkdir /root/xcat2
cd /root/xcat2

tar jxvf core-rpms-snap.tar.bz2
tar jxvf xcat-dep-2*.tar.bz2

if you download xcat-core-2.x.tar.bz2 then

tar jxvf  xcat-core-2.x.tar.bz2
```

2.2.2 [RH] Setup YUM repositories for xCAT and Dependencies

On Red Hat related distros point YUM to the local repositories for xCAT and its dependencies:

```
cd /root/xcat2/xcat-dep/<release>/<arch>
./mklocalrepo.sh
cd /root/xcat2/xcat-core
./mklocalrepo.sh
```

2.2.3 [SLES] Setup Zypper repositories for xCAT and Dependencies

[SLES 10.2+]:

```
zypper sa file:///root/xcat2/xcat-dep/sles10/<arch> xCAT-dep
zypper sa file:///root/xcat2/xcat-core xcat-core
```

[SLES 11]:

```
zypper ar file:///root/xcat2/xcat-dep/sles11/<arch> xCAT-dep
zypper ar file:///root/xcat2/xcat-core  xcat-core
```

You can check status of zypper using command “zypper sl -d” and remove the zypper repository using command zypper sd on SLES 10.2+ or zypper rr on SLES 11.

Get the Requisite Packages From the Distro

xCAT depends on several packages that come from the Linux distro. Follow this section to create the repository of the OS of the Management Node.

2.2.1 [FEDORA 8/9] Setup the respository

If your management node has access to the internet, you can simply create a file called **/etc/yum.repos.d/fedora-internet.repo** that contains:

```
[fedora-everything]
name=Fedora $releasever - $basearch
```

```
failovermethod=priority
#baseurl=http://download.fedora.redhat.com/pub/fedora/linux/releases/
$releasever/Everything/$basearch/os/
mirrorlist=http://mirrors.fedoraproject.org/mirrorlist?repo=fedora-
$releasever&arch=$basearch
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora file:///etc/pki/rpm-gpg/RPM-GPG-
KEY
```

If you management node does not have internet access, then download the necessary fedora RPMs, copy them to the MN, and create a local YUM repository:

Note: you can use this example for Fedora 9, just go to the Fedora 9 website by substituting "9" in for "8" in the below link and wget the equivalent Fedora 9 rpms.

```
cd /root/xcat2/xcat-dep/rh5/x86_64
export
BASEURL=http://download.fedora.redhat.com/pub/fedora/linux/releases/8/Everything/x
86_64/os/Packages/

wget $BASEURL/perl-Net-SNMP-5.2.0-1.fc8.1.noarch.rpm
wget $BASEURL/perl-XML-Simple-2.17-1.fc8.noarch.rpm
wget $BASEURL/perl-Crypt-DES-2.05-4.fc7.x86_64.rpm
wget $BASEURL/net-snmp-perl-5.4.1-4.fc8.x86_64.rpm
wget $BASEURL/ksh-20070628-1.1.fc8.x86_64.rpm
wget $BASEURL/perl-IO-Socket-INET6-2.51-2.fc8.1.noarch.rpm
wget $BASEURL/dhcp-3.0.6-10.fc8.x86_64.rpm
wget $BASEURL/syslinux-3.36-7.fc8.x86_64.rpm
wget $BASEURL/mtools-3.9.11-2.fc8.x86_64.rpm
wget $BASEURL/expect-5.43.0-9.fc8.x86_64.rpm
wget $BASEURL/perl-DBD-SQLite-1.12-2.fc8.1.x86_64.rpm
wget $BASEURL/perl-Expect-1.20-1.fc8.1.noarch.rpm
wget $BASEURL/perl-IO-Tty-1.07-2.fc8.1.x86_64.rpm
wget $BASEURL/scsi-target-utils-0.0-1.20070803snap.fc8.x86_64.rpm
wget $BASEURL/perl-Net-Telnet-3.03-5.1.noarch.rpm
wget $BASEURL/perl-TimeDate-1.16.6.fc8.noarch.rpm
wget $BASEURL/perl-DateTime-0.41-1.fc8.x86_64.rpm
wget $BASEURL/perl-DateTime-Set-0.25-4.fc7.noarch.rpm
wget $BASEURL/perl-MailTools-1.77-2.fc8.noarch.rpm
wget $BASEURL/perl-Set-Infinite-0.61-3.fc7.noarch.rpm
wget $BASEURL/perl-MIME-Lite-3.01-5.fc8.1.noarch.rpm
wget $BASEURL/perl-version-0.7203-2.fc8.x86_64.rpm
wget $BASEURL/perl-SOAP-Lite-0.68-5.fc8.noarch.rpm

createrepo .
```

Note: if using Fedora 9, you must have at least this version of net-snmp-perl: net-snmp-5.4.1-19.fc9

2.2.2 [RHEL] Setup repository

To make the necessary RHEL RPM prereqs available to the xCAT install process, mount the RHEL CD/DVD or ISO and then create a repo file in /etc/yum.repos.d that points to it.

If you have the RHEL CD(s) or DVD(s):

If the RHEL distro only has one CD or DVD:

1.Insert the RHEL CD or DVD into the CD or DVD drive.

2.Mount the disk into some directory, for example, /media/cdrom/.

3.Create a YUM repository file, for example, rhel-dvd.repo, under directory /etc/yum.repos.d. The YUM repository contents should look like:

```
[rhel-5.3]
name=RHEL 5.3 from DVD
baseurl=file:///media/cdrom/Server
enabled=1
gpgcheck=1
```

Note: To make the YUM repository work persistently after management node reboot, the CD/DVD drive mount needs to be added into the /etc/fstab, or add the mount command into Linux startup scripts.

If the RHEL distro has more than one CDs or DVDs:

1.Insert the RHEL CDs or DVDs into the CD or DVD drive one by one:

A. Mount the CD or DVD device

B. For the 1st CD or DVD only, cd to the mounted directory and run: rpm --import RPM-GPG-KEY-redhat-release

C.Copy the RPMs from the Server subdirectory of the CD or DVD disk to a directory on your hard disk (for example /rhels5.3). You can put the RPMs from all of the CDs or DVDs into the same directory.

2. Cd into the newly created RPM directory, install the createrepo RPM, and run: createrepo .

3.Create a YUM repository file, for example, rhel-cd.repo, in directory /etc/yum.repos.d. The YUM repository contents should look like:

```
[rhel-5.3]
name=RHEL 5.3 from directory
baseurl=file:///rhels5.3
enabled=1
gpgcheck=1
```

If you have the RHEL iso files:

If the RHEL distro only has one iso file:

1.Copy the iso file to any directory, such as /iso

```
mkdir /iso
cp RHEL5.2-Server-20080430.0-ppc-DVD.iso /iso/
```

2.Mount the iso file to a directory such as /iso/rhels5.2

```
cd /iso
mount -o loop RHEL5.2-Server-20080430.0-ppc-DVD.iso /iso/rhels5.2
```

3.Create a YUM repository file, for example, rhel-dvd.repo, under directory /etc/yum.repos.d. The YUM repository contents should look like:

```
[rhe-5-server]
name=RHEL 5 SERVER packages
baseurl=file:///iso/rhels5.2/Server
enabled=1
gpgcheck=1
```

Note: To make the YUM repository work persistently after management node reboot, the iso mount needs to be added into the /etc/fstab, or add the mount command into Linux startup scripts.

If the RHEL distro has more than one iso files

1.For each iso file:
A. Loopback mount the iso file
B. For the 1st iso file only, cd to the mounted directory and run: rpm --import RPM-GPG-KEY-redhat-release
C.Copy the RPMs from the Server subdirectory of the mounted directory to a directory on your hard disk (for example /rhels5.3). You can put the RPMs from all of the iso files into the same directory.
2. Cd into the newly created RPM directory, install the createrepo RPM, and run: createrepo .
3.Create a YUM repository file, for example, rhel-cd.repo, in directory /etc/yum.repos.d. The YUM repository contents should look like:

```
[rhel-5.3]
name=RHEL 5.3 from directory
baseurl=file:///rhels5.3
enabled=1
gpgcheck=1
```

For either case above:

```
change directory to where the RHEL CD image is and run
rpm --import RPM-GPG-KEY-redhat-release
```

2.2.3 [SLES] Setup repository

If you have a SLES ISO:

```
mkdir /iso
copy SLES11-DVD-ppc-GM-DVD1.iso  to /iso/
mkdir /iso/1
cd /iso
mount -o loop SLES11-DVD-ppc-GM-DVD1.iso 1

zypper ar file:///iso/1 sles11
```

Install xCAT and Dependencies on the Management Node

[RH]:

```
yum clean metadata
yum install xCAT
```

[SLES]:

```
zypper install xCAT
```

Install Yaboot on the Management Node for pLinux

Note: it is only necessary to install yaboot-xcat on releases earlier than 2.2.
If using yum/zypper it will automatically be installed for you in xCAT 2.2.
On pre-2.2 releases of xCAT, it is only necessary to manually install yaboot-xcat,

if you are planning to manage any ppc nodes.

[RHEL]:
```
yum install yaboot-xcat.noarch
```

Test xCAT Installation

Add command to the path:
```
source /etc/profile.d/xcat.sh
```

Check to see the database is initialized:
```
tabdump site
```

The output should similar to the following:
```
key,value,comments,disable
"xcatdport","3001",,
"xcatiport","3002",,
"tftpdir","/tftpboot",,
"installdir","/install",,
```

Update xCAT Software after Installation

If you need to update the xCAT 2 rpms later:

- If the management node does not have access to the internet: download the new version of http://xcat.sf.net/yum/core-rpms-snap.tar.bz2 and untar it in the same place as before.
- If the management node has access to the internet, the yum/zypper command below will pull the updates directly from the xCAT site.

[RH]:
If you want to just update the xCAT rpms then run the below command. Note: this will not apply the changes that may have been made to the xCAT deps packages.

```
yum update '*xCAT*'
```

If you want to make all updates for xCAT, xCAT rpms and deps, run the following command. This command may also pick up additional OS updates.

```
yum update
```

[SLES]:

If you want to just update the xCAT rpms then run the below commands. Note: this will not apply the changes that may have been made to the xCAT deps packages.

```
zypper refresh
zypper update -t package '*xCAT*'
```

If you want to make all updates for xCAT, xCAT rpms and deps, run the following commands. These command may also pick up additional OS updates.

```
zypper refresh
zypper update
```

Note: If you have a service node stateless image in a hierachical configuration , don't forget to update the image with the new xCAT rpms to keep the service node at the same level as the management Node.

1.1

3.0 Deploying and Maintaining Cluster Nodes

Note: In xCAT 2.3 and bedyond, the node deployment progress can be monitored and reinitiated if necessary, see rnetboot and rpower manpages for more details.

3.1 Linux Nodes

xCAT supports deployment of diskfull and diskless Linux nodes in the cluster.

pLinux Cookbook

The xCAT2 pLinux Cookbook provides information on setting up diskfull and diskless pLinux clusters.

Linux BladeCenter How-To

Instructions for installation of a BladeCenter configuration are contained in the BladeCenter How-to.

IdataPlex How-To

An example of an iDataPlex configuration, and instructions for installation are contained in the iDataPlex How-to.

SLES 10.1 notes

Some helpful notes on installing SLES 10 SP1 are contained in the "SLES 10 SP 1 notes".

3.2 AIX Nodes

XCAT supports the deployment of AIX nodes. This includes the standard "rte" type installation, the cloning of nodes using mksysb images and the booting of diskless nodes. XCAT uses the base AIX support called NIM (Network Installation Management).

Installing AIX standalone nodes (using standard NIM rte method)

http://xcat.svn.sourceforge.net/viewvc/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT2onAIXinstall.pdf

Booting AIX diskless nodes

http://xcat.svn.sourceforge.net/viewvc/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT2onAIXDiskless.pdf

Cloning AIX nodes (using an AIX mksysb image)

http://xcat.svn.sourceforge.net/viewvc/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT2onAIXmksysb.pdf

AIX on JS blade installation

The document **xCAT2 AIX on JS blade – Howto** focuses on the step by step installation introduction for AIX on a JS blade center.

3.3 Updating the xCAT Cluster

You may need to update the nodes in the cluster (OS or applications) updating after the initial install. There are several ways to update depending on the scope of the update. You can use the updatenode function in xCAT to install additional software on the nodes. Other options are to use nimnodecust, rolling update, xdsh , update the diskless image and reboot etc.

Updating AIX cluster nodes

Use Updating AIX Cluster Nodes for information on updating your nodes in the cluster.

4.0 Node Discovery

One of the significant features of xCAT 2 is the node discovery approach. It ultimately performs the role of associating node MAC addresses with IP based on some physical cue (ethernet port or Bladecenter slot). It has the same goal as getmacs fulfilled historically, except it is node initiated, has more context to enable accommodation of more complex configurations, and automated.

The xCAT wiki section on Node Discovery contains information on the setup required to use the Discovery function.

5.0 Using Hierarchy

In large clusters it is desirable to have more than one node (the Management Node) handle the installation and management of the compute nodes. We call these additional nodes service nodes. You can have one or more service nodes set up to install & manage groups of compute nodes. With xCAT, you have the choice of either having each service node install a distinct set of compute nodes, or, if you are using Linux, having a pool of service nodes, any of which can respond to an installation request from a compute node. This document will cover the former case (distinct sets).

The service nodes need to communicate with the xCAT 2 database on the Management Node and run xCAT commands to install the nodes. The service node will be installed with the xCAT code and requires that the either MySQL or PostgreSQL Database be set up instead of the SQLite Default

database. These databases allows a remote client to be set up on the service node such that the service node can access (read/write) the database on the Management Node.

5.1 Setting up a hierarchical cluster

For information about adding hierarchy to your xCAT cluster, read xCAT2SetupHierarchy.

5.2 Setting up alternate databases

XCAT supports a plugable interface which allow you to choose the relational database you wish to use. The following are the currently supported databases, with SQLite being the default when xCAT is installed on the Management Node for the first time.

SQLite

XCAT will automatically perform the initial setup of an SQLite Database when the Management Node is first installed. This database is sufficient for small to moderate size systems (less than 1000 nodes for Linux, 300 for AIX) , if you are not using hierarchy (service nodes). SQLite cannot be used for hierarchy, because the service nodes require access to the database from the service node and this SQLite does not support remote access to the database. For hierarchy, you need to setup PostgreSQL or MySQL, see below.

PostgreSQL

Instructions for setting up a PostgreSQL database on Linux, are found in the xCAT2 Setup Posgresql documentation.

MySQL

Instructions for setting up a MySQL data base for xCAT on AIX or Linux are found in the xCAT2 MySQL setup documentation.

DB2 (TBD)

6.0 Synchronizing files from the MN to the nodes.

In xCAT2.3 and later support is provided to automatically sync fileSyncfiles How-Tos from the Management Node to the Service and compute nodes during and post installaion. The Syncfiles How-To describes how to do this.

7.0 Monitoring

There are two monitoring infrastructures in xCAT 2.0. The ***xCAT Monitoring Plug-in Infrastructure*** allows you to plug-in one or more third party monitoring software such as Ganglia, RMC, SNMP etc. to monitor the xCAT cluster. The ***xCAT Notification Infrastructure*** allows you to watch for the changes in xCAT database tables.

How to enable and use the xCAT Monitoring infrastructure is documented in the xCAT 2.0 Monitoring How-to .

8.0 xCAT imaging

Steps for imaging with xCAT are documented in xCAT imaging with Partimage.

9.0 xCAT Statelite

Information on setting up a Linux Cluster with NFS root diskless nodes is contained in the xCAT Statelite Cookbook.

10.0 xCAT IB Support

The xCAT IB support document will explain how to configure xCAT for infiniband.

11.0 SystemP hardware discovery and HMC connection

How to perform system P hardware automation discovery and setup HMC connection are documented in System p Hardware Management.

12.0 Customize your installation methods

How to setup customized methods of installation are documented in

Boot Targets and Customized Installation methods.

13.0 Uninstalling xCAT on Linux

Steps for removing xCAT from your Management Node are documented in Uninstalling xCAT2.

14.0 Migrating from xCAT 1.3 to 2.x

If you are planning to migrate from xCAT 1.3 to 2.x, read the xCAT Quick Install Tutorial.

15.0 xCAT for CSM Admin

The Redbook "xCAT 2 Guide for the CSM System Administrator" describes the xCAT architecture, Quick deployment, and contains CSM to xCAT transition scenarios.

The Redbook "Configuring and Managing AIX Clusters with xCAT2" describes xCAT2 on AIX and how to transition AIX CSM to xCAT.

16.0 xCAT How-To(s)

There have been a variety of How-To's put out on the xCAT Wiki on various subjects and more are constantly beeing added. Check out the How-To's for valuable information.

16.1 xCAT on Windows

Directions for installing a Windows 2008 Enterprise Server (x86 / x86_64) node with xCAT 2.1 are being developed. The latest notes are available on the WIKI for xCAT on Windows.

16.2 xCAT and Xen

xCAT can be configured to work with the Xen hypervisor to install and manage virtual compute nodes. Installing and setting up Xen for xCAT is documented in the xCAT & Xen How-to.

16.3 Stateless GPFS

GPFS is a premier cluster filesystem. You can run it stateless on nodes.
How to setup a Stateless GPFS Cluster documents the process.

16.4 Maui

Installing and setting up MAUI for xCAT is documented in the xCAT How-to for MAUI.

16.5 Moab

Installing and setting up Moab for xCAT is documented in the Moab Adaptive Computing Installation How-to.

16.6 Torque

Installing and setting up Torque for xCAT is documented in the TBD

16.7 Ganglia

Installing and setting up Ganglia for xCAT is documented in the xCAT How-to for Ganglia.

16.8 LDAP

LDAP can be used for user Management in xCAT. Installing and setting up LDAP for xCAT is documented in the xCAT How-to for LDAP.

16.9 XCAT Developer Guide (TBD)

If you would like to develop code for the xCAT project, read the Developer's Guide for information on the process.

17.0 Known Bugs

https://sourceforge.net/tracker/?group_id=208749&atid=1006945

18.0 Feature requests

https://sourceforge.net/tracker/?group_id=208749&atid=1006948

19.0 References

- xCAT web site: http://xcat.sourceforge.net/
- xCAT wiki: WIKI main page

- xCAT mailing list: Joining the mailing list

20.0 Appendix A: xCAT Commands and Database Tables

Note: some of these commands run on Linux and AIX, some are targeted only for AIX or Linux.

Database support

- DB Tables– Complete list of xCAT database tables descriptions.
- chdef - Change xCAT data object definitions.
- chtab - Add, delete or update rows in the database tables.
- dumpxCATdb – dumps entire xCAT database.
- gettab – searches through tables with keys and return matching attributes.
- lsdef - used to display xCAT object definitions which are stored in the xCAT database.
- mkdef – used to create xCAT data object definitions.
- mkrrbc - Adds or deletes BladeCenter management module and switch node definitions in the xCAT cluster database.
- mkrrnodes – adds or deletes nodes in the xCAT cluster database. Allows creation/deletion of many nodes at once.
- nodeadd - Adds nodes to the xCAT cluster database.
- nodech - Changes nodes' attributes in the xCAT cluster database.
- nodels - lists the nodes, and their attributes, from the xCAT database.
- noderm - removes the nodes in the noderange from all database table.
- restorexCATdb – restore the xCAT database.
- rmdef - remove xCAT data object definitions.
- tabdump - isplay an xCAT database table in CSV format.
- tabedit - view an xCAT database table in an editor and make changes.
- tabgrep - list table names in which an entry for the given node appears.
- tabrestore - replaces the contents of an xCAT database table with the contents in a csv file.
- xcatstanzafile - Format of a stanza file that can be used with xCAT data object definition commands.

Hardware Control

- getmacs - Collects node MAC address.
- lshwconn – Display the connection status for FSP and BPA nodes (2.3)
- lsslp - Discovers selected networked services information within the same subnet.
- lsvm - Lists partition profile information for HMC- and IVM-managed nodes.
- mkhwconn - Sets up connections for FSP and BPA nodes to HMC nodes (2.3).
- nodestat - display the running status of a noderange
- rbeacon - Turns beacon on/off/blink or gives status of a node or noderange.
- rcons - remotely accesses the serial console of a node.
- renergy – remote energy management tools (2.3)
- replaycons - replay the console output for a node

- reventlog - retrieve or clear remote hardware event logs
- rflash - Performs Licensed Internal Code (LIC) update support for HMC-attached P5/P6
- rmhwconn – Remove the connections from the FSP and BPA nodes to the HMC nodes (2.3).
- rmigrate - Execute migration of a guest VM between hosts/hypervisors .
- rmvm - Removes HMC- and IVM-managed partitions.
- rnetboot - Cause the range of nodes to boot to network.
- rpower - remote power control of nodes
- rscan - Collects node information from one or more hardware control points.
- rsetboot - Sets the boot device to be used for BMC-based servers for the next boot only.
- rspconfig - configures various settings in the nodes' service processors.
- rspreset - resets the service processors associated with the specified nodes
- switchblade - reassign the BladeCenter media tray and/or KVM to the specified blade
- wcons – windowed remote console
- wkill – kill windowed remote consoles

Monitoring

- monadd - Registers a monitoring plug-in to the xCAT cluster.
- moncfg - Configures a 3rd party monitoring software to monitor the xCAT cluster.
- mondecfg - Deconfigures a 3rd party monitoring software from monitoring the xCAT cluster.
- monls - Lists monitoring plug-in modules that can be used to monitor the xCAT cluster.
- monrm - Unregisters a monitoring plug-in module from the xCAT cluster.
- monstart - Starts a plug-in module to monitor the xCAT cluster.
- monstop - Stops a monitoring plug-in module to monitor the xCAT cluster.
- regnotif - Registers a Perl module or a command that will get called when changes occur in the desired xCAT database tables.
- unregnotif - unregister a Perl module or a command that was watching for the changes of the desired xCAT database tables.

Inventory

- rinv - remote hardware inventory.
- rvitals – retrieves remote hardware vitals information.
- sinv - Checks the software configuration of the nodes in the cluster.

Parallel Commands

- pcons - runs a command on the noderange using the out-of-band console.
- pping - parallel ping.
- ppping – parallel ping between nodes in a cluster.
- prsync – parallel rsync
- pscp – parallel remote copy (supports scp and not hierarchy)
- psh – parallel remote shell (supports ssh and not hierarchy)
- xdcp – concurrently copies files too and from multiple nodes. (scp/rcp and hierarchy)
- xdsh – concurrently runs commands on multiple nodes. (supports ssh/rsh and hierarchy)
- xdshbak- formats the output of the xdsh command.
- xcoll – Formats command output of the psh, xdsh, rinv command

Deployment

- copycds-cdrom - client side wrapper for copycds supporting physical drives.
- copycds - Copies Linux distributions and service levels from CDs to install directory.
- genimage - Generates a stateless image to be used for a diskless install.
- geninitrd - Regenerates the initrd for a stateless image to be used for a diskless install.
- liteimg - Modify statelite image
- mkdsklsnode - xCAT command to define and initialize AIX/NIM diskless machines.
- mknimimage - xCAT command to create AIX image definitions.
- mknb - creates a network boot root image for node discovery and flashing
- nimnodecust - xCAT command to customize AIX/NIM standalone machines.
- nimnodeset - xCAT command to initialize AIX/NIM standalone machines.
- nodeset - set the boot state for a noderange
- packimage - Packs the stateless image from the chroot file system.
- rbootseq - Persistently sets the order of boot devices for BladeCenter blades.
- rinstall - Begin installation on a noderange
- rmdsklsnode - Use this xCAT command to remove AIX/NIM diskless machine definitions.
- rmnimimage - xCAT command to remove an xCAT osimage definition and the associated NIM resources.
- setupiscsidev - creates a LUN for a node to boot up with, using iSCSI.
- updateSNimage – (No longer used) Adds the needed Service Node configuration files to the install image.
- updatenode - Reruns postsctipts or runs additional scripts on the nodes.
- winstall - Begin installation on a noderange and display in wcons
- xcat2nim - Use this command to create and manage AIX NIM definitions based on xCAT object definitions.

csm to xCAT migration tools

- csm2xcat - Migrates a CSM database to a xCAT database.
- cfm2xcat – (2.3+)Migrates a CSM cfmupdatenode set to the xdcp -F sync files setup in xCAT.
- groupfiles4dsh - Creates a directory of nodegroup files to be used with AIX dsh.

Others

- makedhcp - Creates new dhcp configuration files and updates live dhcp configuration using omapi.
- makedns - sets up domain name services (DNS) from the entries in /etc/hosts.
- makehosts - sets up /etc/hosts from the xCAT hosts table.
- makeconservercf - creates the conserver.cf configuration file and stops and starts conserver.
- makeknownhosts – creates a ssh known_hosts file from the input node range.
- makenetworks - populates the xCAT networks table, using network information from the local system
- mysqlsetup – automatically setup of the MySQL database and xCAT to use MySql.
- noderange – Supported syntax for compactly expressing a list of node names.

- pbstop - Monitors your cluster in a terminal window.
- xcatconfig – setups up MN during install. Can be used to reinitialize keys, credentials, site table after install.
- xcatstart - Starts the xCAT daemon (xcatd) on AIX.
- xcatstop - Stops the xCAT daemon (xcatd) on AIX.
- restartxcatd - restart the xCAT daemon (xcatd) on AIX. (xCAT2.4)
- xCATWorld – Sample client program for xCAT.
- xpbsnodes - PBS pbsnodes front-end for a noderange.
- Summary of xCAT Commands

21.0 Appendix B: xCAT node group support (static and dynamic)

A node group is essentially a named collection of cluster nodes that can be used as a simple way to target an action to a specific set of nodes. The node group names can be used in any xCAT command that targets a node range.

XCAT supports both *static* and *dynamic* groups. A *static* group is defined to contain a specific set of cluster nodes. A *dynamic* node group is one that has its members determined by specifying a selection criteria for node attributes. If a nodes attribute values match the selection criteria then it is dynamically included as a member of the group. The actual group membership will change over time as nodes have attributes set or unset. This provides flexible control over group membership by defining the attributes that define the group, rather than the specific node names that belong to the group. The selection criteria is a list of "attr<operator>val" pairs that can be used to determine the members of a group, (see below).

Note: Dynamic node group support is available in xCAT version 2.3 and later.

In xCAT, the definition of a *static* group has been extended to include additional attributes that would normally be assigned to individual nodes. When a node is part of a *static* group definition it can inherit the attributes assigned to the group. This feature can make it easier to define and manage cluster nodes in that you can generally assign nodes to the appropriate group and then just manage the group definition instead of multiple node definitions. This feature is not supported for *dynamic* groups.

To list all the attributes that may be set for a group definition you can run:

lsdef -t group -h

When a node is included in one or more *static* groups a particular node attribute could actually be stored in several different object definitions. It could be in the node definition itself or it could be in one or more *static* group definitions. The precedence for determining which value to use is to choose the attribute value specified in the node definition if it is provided. If not, then each *static* group that the node belongs to will be checked to see if the attribute is set. The first value that is found is the

value that is used. The static groups are checked in the order that they are specified in the "groups" attribute of the node definition.

NOTE: In a large cluster environment it is recommended to focus on group definitions as much as possible and avoid setting the attribute values in the individual node definition. (Of course some attribute values, such as a MAC addresses etc., are only appropriate for individual nodes.) Care must be taken to avoid confusion over which values will be inherited by the nodes.

Group definitions can be created using the **mkdef** command, changed using the **chdef** command, listed using the **lsdef** command and removed using the **rmdef** command.

Creating a static node group

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

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

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

or

nodech node01,node02,node03 groups=aixnodes

The "-p" (plus) option specifies that "*aixnodes*" be added to any existing value for the "groups" attribute. The "-p" (plus) option is not supported by **nodech** command.

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.

Creating a dynamic node group

The selection criteria for a dynamic node group is specified by providing a list of "attr<operator>val" pairs that can be used to determine the members of a group. The valid operators include: "==", "!=", "=~" and "!~". The "attr" field can be any node definition attribute returned by the **lsdef** command. The "val" field in selection criteria can be a simple sting or a regular expression. A regular expression can only be specified when using the "=~" or "!~" operators. See http://www.perl.com/doc/manual/html/pod/perlre.html for information on the format and syntax of regular expressions.

Operator descriptions:

==	Select nodes where the attribute value is exactly this value.
!=	Select nodes where the attribute value is not this specific value.

=~	Select nodes where the attribute value matches this regular expression.
!~	Select nodes where the attribute value does not match this regular expression.

The selection criteria can be specified using one or more "-w attr<operator>val" options on the command line.

If the "val" field includes spaces or any other characters that will be parsed by shell then the "attr<operator>val" needs to be quoted.

For example, to create a *dynamic* node group called *mygroup*, where the hardware control point is *hmc01* and the partition profile is not set to *service*.

mkdef -t group -o mygroup -d -w hcp==hmc01 -w pprofile!=service

To create a *dynamic* node group called *pslesnodes*, where the operating system name includes *sles* and the architecture includes *ppc*.

mkdef -t group -o pslesnodes -w os=~sles[0-9]+ -w arch=~ppc

To create a dynamic node group called *nonpbladenodes* where the node hardware management method is not set to *blade* and the architecture does not include *ppc*

mkdef -t group -o nonpbladenodes -w mgt!=blade -w 'arch!~ppc'

22.0 Appendix C: xCAT Database Tables and Object Defs

The xCAT data that is used to manage a cluster is contained in a relational database. Different types of data are stored in different tables. You can manage this information directly using a set of table oriented commands provided by xCAT.

See the man page for xcatdb for a description of the xCAT tables.

Note:

For all of the node related tables (except the nodelist table), an extra feature is provided to allow one row in the table describe a lot of nodes: instead of a node name in the 1st column you can specify a node group. If all the values you are specifying in this row are the same for the whole node group, then you just specify them as you always do. If the values vary among the members of the node group with some regular pattern, you can specify a regular expression. See the xCAT database man page (use URL http://xcat.sourceforge.net/man5/xcatdb.5.html) for details.

When xCAT is searching a table for a value for a specific node, it will first look for a row that specifies the exact node name. If not found, it will then look for rows with a group that the node is a member of. It will search groups in the order that the groups are specified for that node in the nodelist table. For this reason, it will make most sense for you to list the groups in the nodelist table in order of most specific to most general.

To manage these tables directly, xCAT provides the **chtab, tabdump, tabrestore**, and **tabedit** commands.

22.1 Object definitions

In addition to managing the database tables directly, xCAT also supports the concept of data object definitions. Data objects are abstractions of the data that is stored in the xCAT database. This support provides a conceptually simpler implementation for managing cluster data, (especially data associated with a specific cluster node). It is also more consistent with other IBM systems management products. The attributes and values defined in the data object definitions will still be stored in the database tables defined for xCAT. These data object definitions should not limit experienced xCAT customers from managing the specific tables directly, if they so desire. A new set of commands is provided to support the object definitions. These commands will automatically handle the storage in and retrieval from the correct tables.

The following data object types are currently supported.

1) **site** - Cluster-wide information. All the data is stored in the *site* table.
2) **node** - Information for a specific cluster node. The data for a node is stored in multiple tables in the database. The commands that are provided to manage these definitions automatically figure out which attributes are stored in which table. It is therefore not necessary to keep track of a large number of table names and attribute locations.
3) **network** - A description of a unique network. This data is stored in the *networks* table.
4) **monitoring** - A description of a monitoring plugin. This data is stored in the *monitoring* table.
5) **notification** - Defines the Perl modules and commands that will get called for changes in certain xCAT database tables. The data is stored in the *notification* table.
6) **group** – Defines a set of nodes. A group definition can be used as the target set of nodes for a specific xCAT operation. It can also be used to define node attributes that are applied to all group members. The group data is stored in multiple tables in the database.
7) **osimage** – Defines a unique operating system image and related resources that are required for xCAT to deploy a cluster node.
8) **policy** - Controls who has authority to run specific xCAT operations.
9) **boottarget** - Target profiles with their accompanying kernel parameters.
10) **eventlog** - Stores events that occurred.

There are four basic xCAT commands that may be used to manage any of the data object definitions.

- **mkdef** – Make data object definitions.
- **chdef** - Change data object definitions.
- **lsdef** - List data object definitions.
- **rmdef** - Remove data object definitions.

In addition to the standard command line input and output the **mkdef**, **chdef**, and **lsdef** commands support the use of a stanza file format for the input and output of information. Input to a command can be read from a stanza file and the output of a command can be written to a stanza file. A stanza file contains one or more stanzas that provide information for individual object definitions. For example:

1. To create a set of definitions using information contained in a stanza file.
   cat mystanzafile | mkdef -z

2. To write all node definitions to a stanza file.
   lsdef –t node -l -z > nodestanzafile

The stanza file support also provides an easy way to backup and restore the cluster data.

For more information on the use of stanza files see the **xcatstanzafile man** page.

23.0 Appendix D: Customizing Your Nodes by Creating Your Own Postscripts

xCAT automatically runs a few postscripts that are delivered with xCAT to set up the nodes. You can also add your own postscripts to further customize the nodes. To add your own postscript, place it in /install/postscripts on the management node. Then add it to the postscripts table for the group of nodes you want it to be run on (or the "all" group if you want it run on all nodes):

```
chtab node=mygroup postscripts.postscripts=mypostscript
```

On each node, 1st the scripts listed in the xcatdefaults row of the table will be run and then the scripts for the group that this node belongs to. If the node is being installed, the postscripts will be run after the packages are installed, but before the node is rebooted. If the node is being diskless booted, the postscripts are run near the end of the boot process. Best practice is to write the script so that it can be used in either environment.

When your postscript is executed on the node, serveral variable will be set in the environment, which your script can use to control its actions:

- MASTER – the management node or service node that this node is booting from
- NODE – the hostname of this node
- OSVER, ARCH, PROFILE – this node's attributes from the nodetype table
- NODESETSTATE – the argument given to nodeset for this node
- NTYPE - "service" or "compute"
- all the site table attributes are exported. You can add your own site table attributes to have other data available to your postscript.

23.1 Suggestions for writing postscripts

- Some compute node profiles exclude perl to keep the image as small as possible. If this is your case, your postscripts should obviously be written in another shell language, e.g. bash, ksh.
- If a postscript is specific for an os, name your postscript mypostscript.osname, e.g setupssh.aix.
- Add "logger" statements to send errors back to the Management Node. XCAT sets up that all syslog messages will be routed to the Management Node. This will help debug.