

xCAT2 Top Doc

07/30/2010, AM 10:47:45

Table of Contents

1.0 Overview of xCAT	2
1.1 xCAT Architecture	3
1.2 Supported Operating Systems	3
1.3 Stateless, Stalelite and Stateful Choices	3
1.4 Defining Hostnames in xCAT	4
1.5 xCAT Features	4
1.6 xCAT license	5
1.7 xCAT support	5
2.0 Installing an xCAT Management Node	7
2.1 Install an AIX Management Node	7
2.2 Install a Linux Management Node	7
2.3 Download and Install xCAT 2 From an MN That Has Internet Access	10
2.4 Download and Install xCAT 2 For an MN That Does Not Have Internet Access	11
2.5 Get the Requisite Packages From the Distro	12
2.6 Install xCAT and Dependencies on the MN (yum/zypper)	15
2.7 Install/Update xCAT and Deps on the MN (without yum/zypper)	16
2.8 Install Yaboot on the MN for pLinux(pre2.3)	17
2.9 Update xCAT Software after Installation	18
3.0 Deploying and Maintaining Cluster Nodes	18
3.1 Linux Nodes	18
3.2 pLinux Cookbook	19
3.3 Linux BladeCenter How-To	19
3.4 IdataPlex How-To	19
3.5 SLES 10.1 notes	19
4.0 AIX Nodes	19
4.1 Installing AIX standalone nodes (using standard NIM rte method)	19
4.2 Booting AIX diskless nodes	19
4.3 Cloning AIX nodes (using an AIX mkysyb image)	19
4.4 AIX on JS blade installation	19
5.0 Updating the xCAT Cluster	19
5.1 Updating Linux	19
5.2 Updating AIX cluster nodes	20
6.0 Node Discovery	20
7.0 Using Hierarchy	20
7.1 Setting up a hierarchical cluster	20
8.0 Setting up alternate databases	21
8.1 SQLite	21
8.2 PostgreSQL	21
8.3 MySQL	21
8.4 DB2	21

9.0 Synchronizing files from the MN to the nodes	21
10.0 Monitoring	21
11.0 xCAT imaging	22
12.0 xCAT Statelite	22
13.0 xCAT IB Support	22
14.0 SystemP hardware discovery and HMC connection	22
15.0 xCAT IBM HPC Integration Support	22
16.0 Customize your installation methods	22
17.0 Uninstalling xCAT on Linux	23
18.0 Migrating from xCAT 1.3 to 2.x	23
19.0 xCAT for CSM Admin	23
20.0 xCAT How-To(s)	23
20.1 xCAT on Windows	23
20.2 xCAT and Xen	23
20.3 Stateless GPFS	23
20.4 Maui	24
20.5 Moab	24
20.6 Torque	24
20.7 Ganglia	24
20.8 LDAP	24
20.9 XCAT Developer Guide	24
21.0 Known Bugs	24
22.0 Feature requests	24
23.0 References	24
24.0 Appendix A: xCAT Commands and Database Tables	24
25.0 Appendix B: xCAT node group support (static and dynamic)	28
26.0 Appendix C: xCAT Database Tables and Object Defs	30
26.1 Object definitions	31
27.0 Appendix D: Customizing Your Nodes by Creating Your Own Scripts	32
27.1 Types of Post*scripts	32
27.2 Post*Script execution	33
27.3 Prescripts	36
27.4 Suggestions for writing scripts	37

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 xpower, xdsh, xdc, etc...utilize this hierarchy for scalable systems management.

1.2 Supported Operating Systems

xCAT 2 management nodes are supported on this list of [Hardware and OS Distos](#).

1.3 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](#).

1.4 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",,,,,

1.5 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.

1.6 xCAT license

xCAT 2 Open Source License: [Eclipse Public License](#)

1.7 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](#).

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 syslog 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](#).

With release 2.4.1, you can control the amount of auditing with the site.auditskipcmds attribute. This attribute allows you to enter a list of commands that will not be logged, or the variable “ALL” which means log no commands. See the [manpage for the site 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. But if you don't, and you use yum/zypper, it will install the dependencies (e.g. dhcp, bind, expect, httpd, nfs-utils, vsftpd, etc.) for xCAT later.
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
8. Create a separate file system for /install

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 a Linux Management Node

Install the OS of Management Node

xCAT 2 management nodes are supported on this list of [Hardware and OS Distos](#).

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.

[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 hostname must be configured before setting up the xCAT 2 management node. The hostname or its resolvable ip address will be used as the default master name in the xCAT site table, when installed. This name needs to be the one that will resolve to the cluster-facing NIC.

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.

Create a Separate File system for /install (optional)

It is not required, but recommended, that you create a separate file system for the /install directory on the Management Node. The size should be at least 30 meg to hold to allow space for several install images.

[RH] Remove the package tftp-server

If you are a Power5 user, then skip the removal of tftp-server in this section. For more information read [xCAT 2 Cookbook for Linux on IBM System P](#).

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 OpenIPMI-tools packages of the OS, which must be removed before xCAT is installed.

2.3 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. Pick the level of xCAT you wish to install. You can install the latest release, the latest snapshot of the release, or the latest development level (next release). The section points to the repo files that will point YUM to the proper place to get the xCAT RPMs and the dependency package on your Management Node.

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. See section [Disable the Firewall](#).

[RH] Setup Yum

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

To get the current official release:

```
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
```

for example:

```
wget http://xcat.sourceforge.net/yum/xcat-dep/rh5/ppc64/xCAT-dep.repo
```

[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
```

2.4 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.

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
```

[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
```

[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.

2.5 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.

[FEDORA] Setup the repository

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 your 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: The value of <releasever> in following link can be the release number of Fedora, it can be **8,9,12,13**.

```
cd /root/xcat2/xcat-dep/rh5/x86_64
```

```
export
BASEURL=http://download.fedora.redhat.com/pub/fedora/linux/releases/<releasever>/E
verything/x86_64/os/Packages/
```

[Fedora8]

```
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
```

[Fedora9]

You need to download rpm packages which are same with the Fedata8, but net-snmp-5.4.1-19.fc9 is needed for Fedora9.

[Fedora12]

```
wget $BASEURL/expect-5.43.0-19.fc12.i686.rpm
wget $BASEURL/net-snmp-perl-5.4.2.1-18.fc12.x86_64.rpm
wget $BASEURL/syslinux-3.75-4.fc12.x86_64.rpm
wget $BASEURL/dhcp-4.1.0p1-12.fc12.x86_64.rpm
wget $BASEURL/perl-XML-Simple-2.18-5.fc12.noarch.rpm
wget $BASEURL/ksh-20090630-1.fc12.x86_64.rpm
wget $BASEURL/perl-Test-Simple-0.92-82.fc12.x86_64.rpm
wget $BASEURL/perl-Net-Telnet-3.03-9.fc12.noarch.rpm
wget $BASEURL/ipmitool-1.8.11-4.fc12.x86_64.rpm
wget $BASEURL/perl-Crypt-PasswdMD5-1.3-5.fc12.noarch.rpm
wget $BASEURL/perl-DBD-SQLite-1.25-4.fc12.x86_64.rpm
```

[Fedora13]

```
wget $BASEURL/dhcp-4.1.1-15.fc13.x86_64.rpm
wget $BASEURL/ipmitool-1.8.11-4.fc13.x86_64.rpm
wget $BASEURL/syslinux-3.84-1.fc13.x86_64.rpm
wget $BASEURL/ksh-20100309-3.fc13.x86_64.rpm
wget $BASEURL/net-snmp-perl-5.5-12.fc13.x86_64.rpm
wget $BASEURL/expect-5.43.0-19.fc12.i686.rpm
wget $BASEURL/perl-Test-Simple-0.92-112.fc13.x86_64.rpm
```

```
wget $BASEURL/perl-Crypt-PasswdMD5-1.3-6.fc13.noarch.rpm
wget $BASEURL/perl-DBD-SQLite-1.27-3.fc13.x86_64.rpm

createrepo .
```

[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`

[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
```

2.6 Install xCAT and Dependencies on the MN (yum/zypper)

[RH]:

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

[SLES]:

```
zypper install xCAT
```

2.7 Install/Update xCAT and Deps on the MN (without yum/zypper)

If you chose not to use yum or zypper, then you will be managing all the dependencies that are needed for xCAT and it's dependency package. You may want to make sure the entire OS is installed on the Management Node to avoid having to search for missing dependencies. We recommend that you use yum/zypper for the install to avoid having to worry about missing dependencies. You will need to download the xCAT rpms and dependencies as was described in Download and Install xCAT 2 For an MN That Does Not Have Internet Access.

A list of all the dependencies and xCAT rpms must be input to rpm. There are cross dependencies, xCAT requires some dependency rpms and some of the dependency rpms require xCAT rpms.

```
cd /root/xcat2
rpm -Uvh xcat-dep/<release>/<arch>/*.rpm xcat-core/perl-xCAT*.rpm xcat-
core/xCAT-2*.rpm xcat-core/xCAT-client*.rpm xcat-core/xCAT-nbroot-core*.rpm xcat-
core/xCAT-server*.rpm --replacepkgs
```

for example:

```
rpm -Uvh xcat-dep/fedora9/x86_64/*.rpm xcat-core/perl-xCAT*.rpm xcat-
core/xCAT-2*.rpm xcat-core/xCAT-client*.rpm xcat-core/xCAT-nbroot-core*.rpm xcat-
core/xCAT-server*.rpm --replacepkgs
```

You can add these optional rpms to the list:

```
xcat-core/xCAT-rmc*.rpm xcat-core/xCAT-IBMhpc*.rpm xcat-core/xCAT-UI*.rpm
for rmc monitoring, hpc stack support and the web interface.
```

Do not install xCATsn*- this is only for a service node.

Note: you may end up with two atftp rpms during this install, remove atftp and leave atftp-xcat, unless on Power5, then do the reverse leave atftp and remove atftp-xcat.

```
rpm -qa | grep atftp
atfpt*
atftp-xcat*
```

The rpms installed on the MN, without the optional rpms, should look similar to this:

```
xCAT-nbkernel-x86_64-2.6.18_92-4.noarch
perl-xCAT-2.4-snap201005031505.noarch
xCAT-nbkernel-ppc64-2.6.18_92-4.noarch
xCAT-nbroot-oss-x86_64-2.0-snap200801291344.noarch
xCAT-nbroot-core-ppc64-2.4-snap201005031505.noarch
xCAT-nbkernel-x86-2.6.18_92-4.noarch
xCAT-client-2.4-snap201005031505.noarch
xCAT-nbroot-core-ppc64-2.4-snap201004300946.noarch
```



```
xCAT-nbroot-oss-x86-2.0-snap200804021050.noarch
xCAT-nbroot-oss-ppc64-2.0-snap200801291320.noarch
xCAT-nbroot-core-x86-2.4-snap201005031505.noarch
xCAT-2.4-snap201005031505.x86_64
xCAT-nbroot-core-x86_64-2.2-snap200904010841.noarch
xCAT-server-2.4-snap201005031505.noarch
```

Install/Update Service Node (without yum/zypper)

You use the same process to as in the Install/Update xCAT and Deps on the MN (without yum/zypper) for downloading and installing except one rpms changes.

Instead of the installing the xcat-core/xCAT-2*.rpm which is for the MN, you will install the xcat-core/xCATsn-2*.rpm for the service node.

So the command would look like the following:

```
cd /root/xcat2
rpm -Uvh xcat-dep/<release>/<arch>/*.rpm xcat-core/perl-xCAT*.rpm xcat-
core/xCATsn-2*.rpm xcat-core/xCAT-client*.rpm xcat-core/xCAT-nbroot-core*.rpm
xcat-core/xCAT-server*.rpm --replacepks
```

2.8 Install Yaboot on the MN for pLinux(pre2.3)

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",,,
"tftpdirdir","/tftpboot",,,
"installdir","/install",,
```

2.9 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.

3.0 Deploying and Maintaining Cluster Nodes

Note: In xCAT 2.3 and beyond, 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.

3.2 pLinux Cookbook

The [xCAT2 pLinux Cookbook](#) provides information on setting up diskfull and diskless pLinux clusters.

3.3 Linux BladeCenter How-To

Instructions for installation of a BladeCenter configuration are contained in the [BladeCenter How-to](#).

3.4 IdataPlex How-To

An example of an iDataPlex configuration, and instructions for installation are contained in the [iDataPlex How-to](#).

3.5 SLES 10.1 notes

Some helpful notes on installing SLES 10 SP1 are contained in the “[SLES 10 SP 1 notes](#)”.

4.0 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).

4.1 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>

4.2 Booting AIX diskless nodes

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

4.3 Cloning AIX nodes (using an AIX mksysb image)

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

4.4 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.

5.0 Updating the xCAT Cluster

5.1 Updating Linux

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.

5.2 Updating AIX cluster nodes

Use [Updating AIX Cluster Nodes](#) for information on updating your nodes in the cluster.

6.0 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.

7.0 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.

7.1 Setting up a hierarchical cluster

For information about adding hierarchy to your xCAT cluster, read [xCAT2SetupHierarchy](#).

8.0 Setting up alternate databases

XCAT supports a pluggable 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.

8.1 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.

8.2 PostgreSQL

Instructions for setting up a PostgreSQL database on Linux, are found in the [xCAT2 Setup Posgresql documentation](#).

8.3 MySQL

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

8.4 DB2

Instructions for setting up a DB2 database for xCAT on AIX or Linux Power systems are found in the [xCAT2 DB2 setup](#) documentation.

9.0 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.

10.0 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](#) .

11.0 xCAT imaging

Steps for imaging with xCAT are documented in [xCAT imaging with Partimage](#). Note: this code and documentation is user supplied and not tested by the xCAT development team.

12.0 xCAT Stalite

Information on setting up a Linux Cluster with NFS root diskless nodes is contained in the [xCAT Stalite Cookbook](#).

13.0 xCAT IB Support

The [xCAT IB support document](#) will explain how to configure xCAT for infiniband.

14.0 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](#).

15.0 xCAT IBM HPC Integration Support

The wiki page [Setting up the IBM HPC Stack in an xCAT Cluster](#) will explain how to install and configure the following IBM HPC software products in your xCAT cluster:

- GPFS
- LoadLeveler
- Parallel Environment (PE), including:
 - Parallel Operating Environment (POE)
 - Low-Level Application Programming Interface (LAPI)
 - Protocol Network Services Daemo(PNSD)
 - Parallel Debugger (PDB)
- ESSL and PESSL
- IBM vacpp and xlf compilers
- RSCT

16.0 Customize your installation methods

How to setup customized methods of installation are documented in [Boot Targets and Customized Installation methods](#).

17.0 Uninstalling xCAT on Linux

Steps for removing xCAT from your Management Node are documented in [Uninstalling xCAT2](#).

18.0 Migrating from xCAT 1.3 to 2.x

This information is a quick overview, and has not been updated with each new release of xCAT, so some parts become out of date over time.

If you are planning to migrate from xCAT 1.3 to 2.x, read the [xCAT Quick Install Tutorial](#).

19.0 xCAT for CSM Admin

These redbooks contain great information, but they are all one-time efforts and are not updated with each new release of xCAT, so some parts become more out of date as time goes on.

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.

20.0 xCAT How-To(s)

The [wiki](#) has a great ["HowTo" section](#).

Note about How-Tos : Much of this documentation is user contributed and hasn't been verified by the core xCAT development team, so use this documentation with that in mind.

20.1 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](#). See Note about How-Tos.

20.2 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](#). See Note about How-Tos.

20.3 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. See Note about How-Tos.

20.4 Maui

Installing and setting up MAUI for xCAT is documented in the [xCAT How-to for MAUI](#). See Note about How-Tos.

20.5 Moab

Installing and setting up Moab for xCAT is documented in the [Moab Adaptive Computing Installation How-to](#). See Note about How-Tos

20.6 Torque

Installing and setting up Torque for xCAT is documented in the following [Paper](#). See Note about How-Tos.

20.7 Ganglia

Installing and setting up Ganglia for xCAT is documented in the [xCAT How-to for Ganglia](#). See Note about How-Tos.

20.8 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](#).

20.9 XCAT Developer Guide

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

21.0 Known Bugs

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

22.0 Feature requests

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

23.0 References

- xCAT web site: <http://xcat.sourceforge.net/>
- xCAT wiki: [WIKI main page](#)
- xCAT mailing list: [Joining the mailing list](#)

24.0 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.
- [lsflexnode](#) – Displays the information of a flexible node. (2.5 or later)
- [mkdef](#) – used to create xCAT data object definitions.
- [mkflexnode](#) – Create a flexible node. (2.5 or later)
- [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.
- [rmflexnode](#) – Remove a flexible node. (2.5 or later)
- [tabdump](#) - display 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.
- [tabprune](#) – delete records from the eventlog and auditlog tables (2.4 or later).
- [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.
- [rsreset](#) - 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.
- [makeroutes](#) – connect nodes to the Management Node using Service node as gateway. (2.5+)
- [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.
- [snmove](#) – moves nodes from one Service Node to another. (2.5 or later)
- [updateSNimage](#) – (No longer used) Adds the needed Service Node configuration files to the install image.
- [updatenode](#) - Reruns postscripts 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.
- [xcatchroot](#)- Use this command to modify an xCAT AIX diskless operating system image. (2.5)
- [chkosimage](#)- Use this command to check an osimage. (2.5 or later)

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.
- [pgsqlsetup](#) - (2.5 or later) automatically setup of the PostgreSQL database and xCAT to use PostgreSQL.
- [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)
- [xcatchroot](#) – AIX command (2.4+) to modify an xCAT diskless operating system image.
- [xCATWorld](#) – Sample client program for xCAT.
- [xpbsnodes](#) - PBS pbsnodes front-end for a noderange.
- [Summary of xCAT Commands](#)

25.0 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 -d -w os=~sles[0-9]+ -w arch=~ppc
```

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

```
mkdef -t group -o nonpladenodes -d -w mgt!=blade -w 'arch!~ppc'
```

26.0 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.

26.1 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.

27.0 Appendix D: Customizing Your Nodes by Creating Your Own Scripts

xCAT automatically runs a few postscripts and postbootscripts that are delivered with xCAT to set up the nodes. You can also add your own scripts to further customize the nodes. This Appendix explains the xCAT support to do this.

27.1 Types of Post*scripts

There are two types of *scripts in the postscripts table. The types are based on when in the install process they will be executed. `tabdump -d postscripts` also gives information.

- **postscripts attribute** – List of scripts that should be run on this node after diskfull installation or diskless boot. xCAT automatically adds the postscripts from the `xcatdefaults.postscripts` attribute of the table to run first on the nodes after install or diskless boot.
For installation of RedHat, CentOS, Fedora, the scripts will be run before the reboot.
For installation of SLES, the scripts will be run after the reboot but before the `init.d` process.
For Linux diskless deployment, the scripts will be run at the `init.d` time, and xCAT will automatically add the list of scripts from the `postbootscripts` attribute to run after postscripts list.
For AIX, the scripts will run before the reboot.
- **postbootscripts attribute** -list of scripts that should be run on this node at the `init.d` time after diskfull installation reboot or diskless boot. xCAT automatically adds the scripts in the `xcatdefaults.postbootscripts` attribute to run first in the list.

To add your own *script, place it in `/install/postscripts` on the management node. Make sure it is executable and world readable. 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 in the appropriate attribute, according to when you want it to run.

To check what scripts will be run on your node during installation:

```
lsdef node1 | grep scripts
  postbootscripts=otherpkgs
  postscripts=syslog,remoteshell,syncfiles
```

27.2 Post*Script execution

When your script is executed on the node, all the attributes in the site table are exported as variables for your scripts to use. You can add extra attributes for yourself. See the sample mypostscript file below.

To run the postscripts, a script is built, so the above exported variables can be input. You can usually find that script in /tmp on the node and for example in the Linux case it is call mypostscript. A good way to debug problems it to go to the node and just run mypostscript and see errors. You can also check the syslog on the Management Node for errors.

When writing you postscripts, it is good to follow the example of the current postscripts and write errors to syslog and in shell. See Suggestions for writing scripts.

Example of mypostscript:

```
# subroutine used to run postscripts
run_ps () {
  logdir="/var/log/xcat"
  mkdir -p $logdir
  logfile="/var/log/xcat/xcat.log"
  if [[ -f $1 ]]; then
    echo "Running postscript: $1" | tee -a $logfile
    ./$1 2>&1 | tee -a $logfile
  else
    echo "Postscript $1 does NOT exist." | tee -a $logfile
  fi
}
# subroutine end
AUDITSKIPCMDS='tabdump,nodels'
export AUDITSKIPCMDS
TEST='test'
export TEST
NAMESERVERS='7.114.8.1'
```

```
export NAMESERVERS
NTPSERVERS='7.113.47.250'
export NTPSERVERS
INSTALLLOC='/install'
export INSTALLLOC
DEFSERIALPORT='0'
export DEFSERIALPORT
DEFSERIALSPEED='19200'
export DEFSERIALSPEED
DHCPIINTERFACES=""xcat20RRmn|eth0;rra000-m|eth1""
export DHCPIINTERFACES
FORWARDERS='7.113.8.1,7.114.8.2'
export FORWARDERS
NAMESERVER='7.113.8.1,7.114.47.250'
export NAMESERVER
DB='postg'
export DB
BLADEMAXP='64'
export BLADEMAXP
FSPTIMEOUT='0'
export FSPTIMEOUT
INSTALLDIR='/install'
export INSTALLDIR
IPMIMAXP='64'
export IPMIMAXP
IPMIRETRIES='3'
export IPMIRETRIES
IPMITIMEOUT='2'
export IPMITIMEOUT
CONSOLEONDEMAND='no'
export CONSOLEONDEMAND
SITEMASTER=7.113.47.250
export SITEMASTER
MASTER=7.113.47.250
export MASTER
MAXSSH='8'
export MAXSSH
PPCMAXP='64'
export PPCMAXP
PPCRETRY='3'
export PPCRETRY
PPCTIMEOUT='0'
export PPCTIMEOUT
```

```
SHAREDFTFTP='1'
export SHAREDFTFTP
SNSYNCFILEDIR='/var/xcat/syncfiles'
export SNSYNCFILEDIR
TFTPDIR='/tftpboot'
export TFTPDIR
XCATDPORT='3001'
export XCATDPORT
XCATIPORT='3002'
export XCATIPORT
XCATCONFDIR='/etc/xcat'
export XCATCONFDIR
TIMEZONE='America/New_York'
export TIMEZONE
USENMAPFROMMN='no'
export USENMAPFROMMN
DOMAIN='cluster.net'
export DOMAIN
USESSHONAIX='no'
export USESSHONAIX
NODE=rra000-m
export NODE
NFSSERVER=7.113.47.250
export NFSSERVER
INSTALLNIC=eth0
export INSTALLNIC
PRIMARYNIC=eth1
OSVER=fedora9
export OSVER
ARCH=x86_64
export ARCH
PROFILE=service
export PROFILE
PATH='dirname $0':$PATH
export PATH
NODESETSTATE='netboot'
export NODESETSTATE
UPDATENODE=1
export UPDATENODE
NTYPE=service
export NTYPE
MACADDRESS='00:14:5E:5B:51:FA'
export MACADDRESS
```

```

MONSERVER=7.113.47.250
export MONSERVER
MONMASTER=7.113.47.250
export MONMASTER
OSPKGS=bash,openssl,dhclient,kernel,openssh-server,openssh-clients,busybox-anaconda,vim-minimal,rpm,bind,bind-utils,ksh,nfs-
utils,dhcp,bzip2,rootfiles,vixie-cron,wget,vsftpd,ntp,rsync
OTHERPKGS1=xCATsn,xCAT-rmc,rsct/rsct.core,rsct/rsct.core.utils,rsct/src,yaboot-xcat
export OTHERPKGS1
OTHERPKGS_INDEX=1
export OTHERPKGS_INDEX
NOSYNCFILES=1
export NOSYNCFILES
# postscripts-start-here\n
run_ps ospkgs
# postscripts-end-here\n

```

27.3 Prescripts

The prescript table will allow you to run scripts before the install process.

The scripts will be run on the MASTER for the node. If there is a service node for the node, then the scripts will be run on the service node.

```

# tabdump prescripts
#node,begin,end,comments,disable

```

- **begin:** This attribute lists the scripts to be run at the beginning of the nodeset(Linux), nimnodeset(AIX) or mkdsklsnode(AIX) command.
- **end:** This attribute lists the scripts to be run at the end of the nodeset(Linux),nimnodeset(AIX) or mkdsklsnode(AIX) command.

Format for naming prescripts:

The general format is:

```
[action1:]s1,s2...[action2:s3,s4,s5...]
```

where:

- action1 and action2 are the nodeset actions (install, netboot,etc) specified in the command for Linux.
- action1 and action2 can be 'diskless' for mkdsklsnode command and 'standalone' for nimnodeset cmd for AIX.
- s1 and s2 are the scripts to run for action1 in order.
- s3, s4, and s5 are the scripts to run for actions2.

If actions are omitted, the scripts apply to all actions.

Examples:

- myscript1,myscript2 - run scripts for all supported commands (command depends on whether AIX or Linux)
- diskless:myscript1,myscript2 (AIX) - run scripts 1,2 for the mkdsklnode command
- install:myscript1,myscript2|netboot:myscript3
(Linux) – run script 1,2 for nodeset(install), runs script3 for nodeset(netboot).

All the scripts should be copied to /install/prescripts directory and made executable and world readable for mounting.

The following two environment variables will be passed to each script:

1. NODES a coma separated list of node names that need to run the script for
2. ACTION current nodeset action.

If '#xCAT setting:MAX_INSTANCE=number' is specified in the script, the script will get invoked for each node in parallel, but no more than number of instances specified in **number** will be invoked at a time. If it is not specified, the script will be invoked once for all the nodes.

27.4 Suggestions for writing scripts

- 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.